DOCTORAL THESIS

Harnessing Multimodality: Diffusion based Generative Modeling and Information Estimation

Mustapha Bounoua

A thesis submitted in fulfillment of the requirements
for the degree of Doctor of Philosophy in the

Doctoral School N. 130 : Computer Science, Telecommunications and Electronics
Sorbonne University - EDITE de Paris

Committee in charge :

Pietro Michiardi	EURECOM	Advisor
Giulio Franzese	EURECOM	Co-Advisor
Christophe Beaugeant	Ampere	Co-Advisor
Serena Villata	CNRS, INRIA	Reviewer
Tania Cerquitelli	Politecnico di Torino	Reviewer
Giovanni Neglia	Université Côte d'Azur, INRIA	Examiner
Nicholas Evans	EURECOM	Examiner (Jury President)

SORBONNE UNIVERSITY THESIS

Abstract

Computer Science, Telecommunications and Electronics Doctoral School
Sorbonne University (ED130) - EDITE de Paris

Doctor of Philosophy

Harnessing Multimodality: Diffusion based Generative Modeling and Information Estimation

by Mustapha Bounoua

Machine learning has profoundly reshaped both business and daily life over the past decade, primarily due to the availability of vast datasets that power foundation models such as large language models. However, data is inherently multimodal, mirroring how humans perceive the world through their five senses. This pervasive nature of multimodality has driven breakthroughs in applications ranging from text-to-image synthesis and video generation to vision-language models and has influenced fields such as automotive technology, neuroscience, and biology. Nevertheless, multimodality poses several challenges, including the heterogeneity of data types, complex inter-modal interactions, and issues of observability that may lead to unpaired data, thereby constraining their effective utilization. In addressing these challenges, this thesis first proposes a novel generative modeling approach designed to approximate complex multimodal data distribution. By leveraging score-based diffusion models, we propose a method capable of capturing the rich structures of multimodal data. Once the multimodal data is accurately modeled, the focus shifts to interpreting the inter-modal interactions. Accordingly, this thesis develops estimators for information-theoretic measures, including mutual information in dual-modal settings, as well as total correlation, dual total correlation, and O-information for arbitrary numbers of modalities. Finally, recognizing that paired multimodal data is not always available, the thesis introduces an original method for constructing paired couplings in dual-modal cases without pre-existing paired samples. This method leverages diffusion models alongside reinforcement learning to achieve minimum entropy coupling, thereby maximizing mutual information between modalities. The proposed contributions achieved state-of-the-art performance on several benchmarks and unlocked new capabilities in multimodal learning, with tangible impact in automotive applications where enhanced sensor integration improved system robustness. These findings underscore the practical importance of the proposed methods in real-world contexts, paving the way for further advancements in multimodal machine learning.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisor, Prof. Pietro Michiardi, for his unwavering support and invaluable guidance throughout my PhD. I feel incredibly fortunate to have had the opportunity to work under his mentorship. His expertise, openness, and trust in my ideas have been instrumental in shaping both my research and my growth as a researcher.

I am equally grateful to my co-supervisor, Prof. Giulio Franzese, for his thoughtful advice, collaborative spirit, and insightful feedback at every stage of this journey. I would also like to sincerely thank my industrial supervisor, Christophe Beaugeant, for his mentorship and for offering a unique industry perspective that enriched my work, especially through the collaboration between EURECOM's Data Science Department and Ampere Software Technology (Renault Group).

I am deeply thankful to the members of my jury, Prof. Serena Villata, Prof. Tania Cerquitelli, Prof. Giovanni Neglia, and Prof. Nicholas Evans for their time, valuable feedback, and thoughtful discussion during my defense.

I also want to thank my colleagues and friends at EURECOM and Ampere, with whom I shared stimulating discussions, technical challenges, and countless moments of camaraderie. A special thanks goes to the data science department at EURECOM for providing a dynamic and supportive research environment.

Finally, and most importantly, I am deeply grateful to my wife, whose unwavering support, strength, and constant encouragement have been my anchor throughout every high and low of this journey. I am also profoundly grateful to my family for their endless encouragement, sacrifices, and for always being there, even from afar. This accomplishment is as much theirs as it is mine.

Contents

List of Figures

List of Tables

List of Abbreviations

ADAS Advanced driver-assistance systems. 44

CFG Classifier-Free Guidance. 24

Clip-s CLIP-Score. 42, 127, 140

CLUB CLUB. 79

DDIM Denoising Diffusion Implicit Models. 18, 23, 131

DDPM Denoising Diffusion Probabilistic Models. 18, 22, 23, 94, 95, 189

DiT Diffusion Transformer. 47

DM Diffusion Model. 13, 107, 111

DTC Dual Total Correlation. viii, 72, 73, 77–79, 81, 168–170, 175, 177, 181

EBM Energy based Model. 13, 15

ELBO Evidence Lower Bound. vi, 18, 28–30, 32

EMA Exponential moving average. 120

FAD Fréchet Audio Distance. 38, 40, 123, 128

FID Fréchet Inception Distance. 38, 39, 41–43, 101, 103, 124, 125, 128, 132, 137, 138, 140

FMD Fréchet Modality Distance. 39, 40, 123, 125, 128, 132

GAN Generative Adversarial Network. 12, 14, 15, 100, 101

GPS Global Positioning System. 45

InfoNCE InfoNCE. 79

KL Kullback–Leibler divergence. vi, 7, 12, 30, 51–57, 60, 61, 64, 67, 71, 72, 74–77, 82, 92, 93, 108, 109, 111

LiDar Light Detection And Ranging. 7, 44–48, 87, 107, 131

LPIPS Learned Perceptual Image Patch Similarity. 47, 48

MCMC Markov chain Monte Carlo. 16, 21

MEC Minimum Entropy Coupling. 8, 87, 89–94, 100, 102, 109, 188

MI Mutual Information. vi, xi, 7, 51–53, 58, 60–64, 67–74, 78, 79, 81, 82, 85, 91, 108–110, 150, 152, 172, 175–177, 182

MINDE Mutual Information Neural Diffusion Estimation. vi, viii, xiii, xv, 53, 60–68, 108, 110, 150–156, 158, 180, 181

MINE MINE. 79, 175

MLD Multi-modal Latent Diffusion. vi, vii, xi, xii, xv, 7, 26, 27, 31, 37–49, 107, 109, 110, 115, 117, 119–125, 128–143

MLD in-paint Multi-modal Latent Diffusion with In-painting. 117, 118, 121–124, 137, 140

MLD uni Multi-modal Latent Diffusion UniDiffuser. 119, 121–124, 137, 140

MLE Maximum Likelihood Estimation. 12

MLP Multilayer perceptron. xii, 38, 79, 120, 175, 177

MMVAE Mixture of Expert. 29, 37–41, 128, 130, 132–137, 139–141

MMVAE+ MMVAE+. 37, 39, 40, 43, 133–135, 137, 138, 140

MoPoE Mixture of Product of Experts. 29, 37–43, 128, 130, 132–137, 139–141

MSE Mean square error. 130

MVAE Product of Experts. 29, 37–42, 128, 130, 132–137, 139–141

MVTCAE Multi-view Total Correlation Autoencoder. 37–43, 128, 130, 132–137, 139–141

NEXUS NEXUS. 37, 39–41, 128, 130, 132–137, 139–141

NF Normalizing Flow. 13, 17

NLP Natural Language Processing. 2

NWJ NWJ. 79

ODE Ordinary Differential Equation. 21, 23

O-information O-information. vii, ix, xi, xiii, 8, 69–71, 73–76, 78–85, 108, 169–174, 176–185

OT optimal transport. 89–91, 98

PCA Principal Component Analysis. 99

PDE Partial Differential Equation. 74

PED Partial Entropy Decomposition. 70

PID Partial Information Decomposition. 70, 73, 84, 108, 109

PSNR Peak Signal-to-Noise Ratio. 47, 48

RaDar Radio Detection And Ranging. 7, 44–48, 107, 131

SDE Stochastic Differential Equation. 19–21, 32–36, 45, 53, 54, 59, 62, 116, 118, 119, 121, 145, 169

SΩI Score-based O-Information estimation. ix, xiii, xvi, 8, 69, 71, 78–85, 108, 109, 169, 171, 172, 174–181

S-information S-information. xiii, 72, 73, 77, 184, 185

SSIM Structural Similarity Index Measure. 47, 48, 101, 103

TC Total Correlation. viii, 72, 78, 79, 168–170, 175, 177, 181

VAE Variational Autoencoder. 7, 13, 17, 18, 25–31, 38, 40–42, 48, 107, 115, 128–130, 132

VE Variance Exploding. 54

VESDE Variance exploding SDE. 20, 21

VP Variance Preserving. 54

VPSDE Variance preserving SDE. vii, 20, 22, 115, 116, 170–172

S

Notation

Pairwise Case

- **Random Variables:** We consider two random variables, $\mathbf{X}$ and $\mathbf{Y}$, defined over the data spaces $\mathcal{X}$ and $\mathcal{Y}$, respectively.
- **Realizations:** Their realizations are denoted as $\mathbf{x}$ and $\mathbf{y}$.

Multiple Variables

- **Definition:** When dealing with more than two variables, we define:
  - $\mathbf{X} = [\mathbf{X}^1, \ldots, \mathbf{X}^m]$, representing m random variables.
  - Their realizations are denoted as $[\mathbf{x}^1, \ldots, \mathbf{x}^m]$.
  - Each variable $\mathbf{X}^i$ has a corresponding data space $\mathcal{X}^i$ for $i = 1, \ldots, m$.
- **Modality-Specific Data Space:** The data space associated with modality m ($m = 1, \ldots, M$) is denoted by $\mathcal{X}^m$.

Probability Densities

- p and q denote probability density functions.
- $p(\mathbf{x})$ represents the probability density of $\mathbf{x}$ under p.
- **Conditional and Joint Densities:**
  - We refer to the joint density as $p(\mathbf{x}^1, \mathbf{x}^2)$ and the conditional density as $p(\mathbf{x}^1 \mid \mathbf{x}^2)$.
  - When necessary (in some parts of the thesis, such as Chapter 4 and Chapter 6), we use superscripts on p to specify the type of density. For example, $p^{\mathbf{X}|\mathbf{Y}}$ denotes the conditional probability density of $\mathbf{X}$ given $\mathbf{Y} = \mathbf{y}$.

Diffusion Process

- $\mathbf{X}_t$ represents the random variable at time t, with realization $\mathbf{x}_t$.
- $p_t(\mathbf{x}_t)$ denotes the probability density induced by the diffusion process at time t.
- **Learned Score Network:**
  - The learned score network associated with a probability density is denoted as s_θ, where θ are the neural network parameters.
  - When necessary, we use a superscript to indicate the associated density. For example, the learned score network associated with the conditional probability density $p^{\mathbf{X}|\mathbf{Y}}$ is denoted as $s^{p^{\mathbf{X}|\mathbf{Y}}}(\mathbf{x}, \mathbf{y}, t)$. In some cases, we simplify this notation to $s^{\mathbf{X}|\mathbf{Y}}(\mathbf{x}, \mathbf{y}, t)$.
- **Neural Network Parameters:**
  - θ and ϕ represent parameters of the neural networks optimized during training.

Chapter 1

Introduction

1.1 What is a modality ?

> *A modality is the particular way in which something exists, is experienced, or is done.* – **Oxford Advanced Learner's Dictionary**

Although the term "*modality*" is often used in everyday language, its implications in computational systems are both deep and nuanced. The concept traces its origins from human perception and our capacity to gather information through various senses such as sight, hearing, smell and touch (Baltrušaitis, Ahuja, and Morency, 2018; Liang, Zadeh, and Morency, 2022). In the machine learning domain, a modality is defined more rigorously as *a specific mechanism to encoding and conveying information* (Guo, Wang, and Wang, 2019), with some scholars extending this definition in a task-specific manner (Parcalabescu, Trost, and Frank, 2021). Throughout this thesis, we will consider as modality *any channel or source through which information is conveyed.* This conceptualization is grounded in the classical notion of a communication channel, as formulated by Shannon (1948) and elaborated by Cover, Thomas, et al. (1991). Analogous to how a communication channel transmits information from a sender to a receiver, each modality (e.g., text, image, audio) operates as a distinct conduit with its own structure and content. Multimodality designates the case where several modalities are considered together to address a specific task. Several properties make multimodal learning a compelling area of study. As noted by (Liang, Zadeh, and Morency, 2022), modalities in a multimodal context possess several key properties:

- **Heterogeneity:** Different modalities exhibit distinct structures and representations such as pixel intensities in images versus the linguistic patterns of text—reflecting their unique information characteristics.

- **Redundancy:** Modalities often contain overlapping or redundant information, which can be harnessed for tasks such as cross-modal translation.
- **Synergy:** The joint processing of modalities can yield synergistic insights that are inaccessible when considering any single modality in isolation.

1.2 Multimodal Machine Learning

Machine learning originated from the ambition of replicating human brain behavior, with early artificial neural networks explicitly designed to emulate the functioning of biological neurons (McCulloch and Pitts, 1943). Turing (1950) posed the foundational question of whether machines can think, laying the groundwork for the development of algorithms capable of simulating learning and problem-solving.

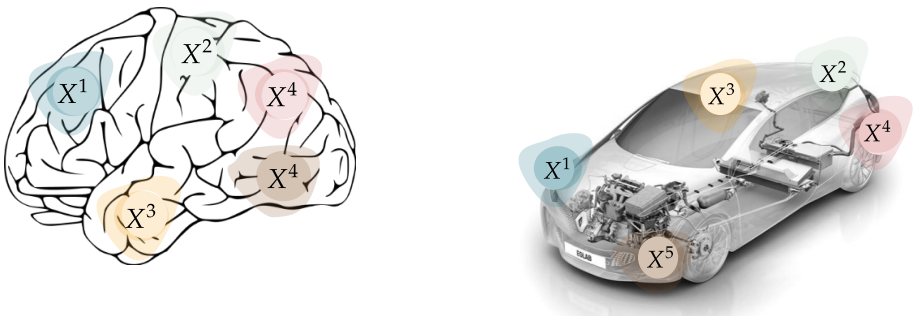
The advancement of computational resources, the development of increasingly sophisticated neural network architectures, improvements in training methodologies, and the growing availability of large-scale datasets have collectively driven significant progress in the field of deep learning (LeCun, Bengio, and Hinton, 2015). Pioneering models such as AlexNet (Krizhevsky, Sutskever, and Hinton, 2012) and, subsequently, ResNet (He et al., 2016) have fundamentally transformed image classification tasks. Moreover, the introduction of Transformer architectures (Vaswani et al., 2017) has led to substantial breakthroughs in Natural Language Processing (NLP). Further advances include the emergence of diffusion models for high-fidelity image generation (Ho, Jain, and Abbeel, 2020), the development of large language models such as GPT (Brown et al., 2020) for conversational agents, and the deployment of AlphaFold (Jumper et al., 2021) for accurate protein structure prediction.

However, in striving toward the ultimate goal of emulating the human brain's reasoning capabilities, a critical need emerges: the integration of multiple data modalities. The human brain is inherently multimodal (Stein and Meredith, 1993), simultaneously processing visual, auditory, tactile, and other sensory inputs to achieve complex reasoning. Humans naturally synthesize information from diverse sensory channels in everyday activities, for instance, using visual supports to facilitate language learning or relying on both perceptual and acoustic signals when riding a bicycle or driving a car.

Inspired by these biological insights, Deep learning has evolved significantly over the past few decades. The early algorithms were predominantly focused on a single modality such as text or images. However, as researchers aimed to emulate the brain's integrative and holistic processing, novel architectures were developed to jointly process and align information from diverse data sources. Modern deep learning models are explicitly designed

to align and fuse features from different modalities. These advances have enabled a wide range of applications, including text-to-image generation (Rombach et al., 2022), visual question answering (Anderson et al., 2018), audio-visual scene understanding (Jaegle et al., 2021), and multimodal robotics (Akkaya et al., 2019). More recently, the development of conversational agents capable of processing and reasoning across diverse modalities has been exemplified by the emergence of multimodal large language models (LLMs) such as ChatGPT-4 (Achiam et al., 2023) and LLaMA-2 (Touvron et al., 2023).

Multimodal learning arguably represents the closest computational parallel to the brain's integrative processing of information. By leveraging the inherent redundancy across different modalities, these models assimilate information more efficiently. Furthermore, by capturing the interactions and dependencies among modalities, multimodal systems can extract synergistic signals that further enhance performance. With increasing data volumes, more complex tasks, and growing computational power, multimodal learning is gaining traction and holds great promise for developing better machine learning models. However, even the most powerful multimodal applications may fall short of their potential if the unique characteristics of each modality and their underlying interactions are not sufficiently captured. A clear understanding of the data distribution within each modality and the nature of their inter-modal interactions is essential to fully exploit multimodality.



Brain regions

Automotive: Multimodal Sensors

Figure 1.1: Multimodal systems

1.3 Modeling and Interpreting Multimodality

Modeling multimodal data presents significant challenges, primarily due to the inherent heterogeneity and the presence of complex, latent patterns across modalities. Generative modeling offers a powerful framework for capturing the underlying distributions of multimodal data and for synthesizing new, coherent samples that effectively integrate information from multiple modalities.

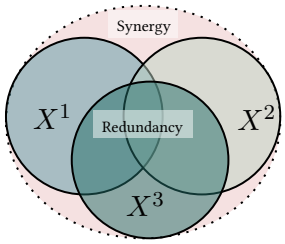


Figure 1.2: Venn diagram modeling how information is organized across several modalities.

The rationale behind employing generative modeling lies in its ability to learn and represent the underlying structure of the data. Once this structure is captured, it becomes possible to generate new samples that reflect the joint multimodal distribution. Moreover, generative models enable the exploration of diverse generation pathways, such as translating from images to speech or from speech to text. These cross-modal translations pave the way for a range of compelling applications, including text-to-image synthesis, image captioning, and video-to-audio generation. Crucially, the success of such translations depends on accurately capturing the inter-modal interactions embedded within multimodal data.

By accurately modeling the joint distribution, generative models not only produce realistic samples but also reveal the interactions between modalities (see § 1.3). Understanding and quantifying these interactions is of critical importance. Certain generative modeling techniques function as explicit density estimators, thereby enabling the estimation of information-theoretic measures that provide deeper insights into multimodality. This capability is particularly valuable in the analysis of complex systems, where each modality may correspond to the output of a distinct component. Quantifying these interactions contributes to a deeper understanding of the overall structure of a system. For example, the human brain (see § 1.2) can be conceptualized as a multimodal system, wherein distinct neural regions function as specialized modalities. It is therefore crucial to understand how these entities interact and coordinate to capture the underlying dynamics of complex cognitive processes. Similarly, in the automotive domain, data from multiple vehicle sensors, provides essential insights for system design and optimization. In this context, generative models serve not only as tools for data synthesis but also as analytical instruments for probing the underlying dynamics of complex multimodal systems.

Moreover, while effective multimodal modeling frequently assumes the availability of paired and aligned data, many real-world applications are characterized by unpaired data scenarios. For instance, in RNA sequencing, the destructive nature of the measurement process inherently leads to unpaired observations (Kester and Oudenaarden, 2018). Developing methods for constructing reliable pairings from unpaired data thus represents a critical research challenge. A practical strategy involves seeking pairings that fulfill specific

Figure 1.3: Ink gracefully diffuses throughout a glass of water. Source DALL-E 3

interaction criteria such as constructing a multimodal distribution that maximizes the shared information between modalities.

1.4 Multimodality and Diffusion models

In physics, *diffusion* refers to the stochastic spread of particles from high concentration regions to low concentration due to random motion. From a thermodynamic perspective, this process is governed by *entropy*, a measure of disorder in a system. As diffusion occurs, the system evolves towards a state of maximal entropy, as seen in phenomena such as heat dissipation or the spreading of ink in water (Callen, 1991) (see Figure 1.3). This natural tendency of isolated systems to transition toward greater disorder is formalized by the *second law of thermodynamics*, which states that the total entropy of an isolated system can never decrease over time (Crooks, 1999).

Diffusion models are a powerful class of generative models inspired by thermodynamic diffusion processes. These models define a forward process in which structured data is progressively corrupted through the sequential addition of noise. This process incrementally increases the entropy of the data until all original information is effectively lost, mirroring how physical systems naturally evolve toward thermodynamic equilibrium. A key distinguishing feature of diffusion models is their ability to model the reverse process, allowing reconstruction of data from noise (Sohl-Dickstein et al., 2015; Ho, Jain, and Abbeel, 2020; Song et al., 2021). Although the second law of thermodynamics stipulates that entropy in an isolated system cannot decrease, the reverse process in diffusion models does not occur in isolation; thus, the second law does not directly apply. Instead, the reverse process is guided by external information, encapsulated in a learned score function that represents the gradient of the log density of the data (*score function*) (see Figure 1.4). Diffusion models offer several advantages over alternative generative models. The gradual reverse process helps capture a wide range of variations and ensures broad mode coverage, effectively exploring

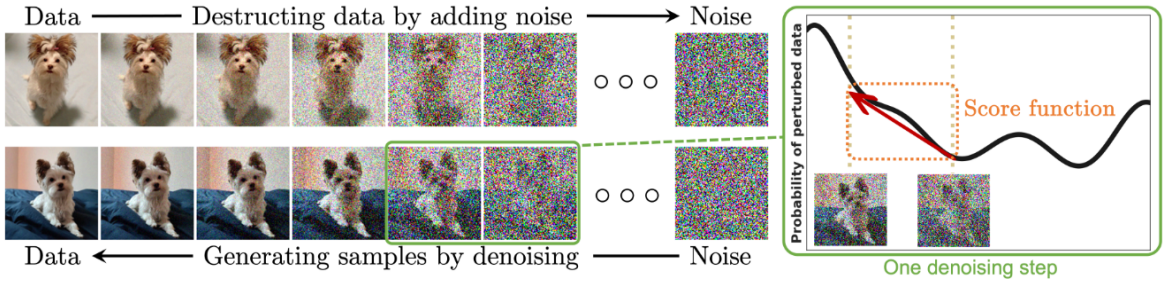


Figure 1.4: Diffusion models smoothly perturb data by adding noise, then reverse this process to generate new data from noise using the score function. Source: ((Yang et al., 2023), Figure 2).

the entire distribution. In addition, diffusion models are explicit density estimators. These models are trained using a variational bound, which provides greater stability by optimizing a well-defined probability distribution. Due to their continuous and probabilistic nature, diffusion models are particularly well suited for handling multimodal distributions. They have demonstrated efficiency across various data modalities and have also been successfully applied in conditional settings, highlighting their ability to capture intra-modal relationships. In multimodal settings, diffusion models are a strong candidate for modeling complex multimodal distributions.

1.5 Outline and Contributions of the Thesis

In this thesis, we tackle various challenges associated with multimodality and explore strategies for leveraging it in a principled and effective manner. First, we propose a multimodal generative model specifically designed to handle multimodal data and address the inherent challenges in such settings. We demonstrate that diffusion models are particularly well suited for modeling these complex scenarios. Next, we show that, beyond their application in generative modeling, diffusion-based multimodal generative models can also be used to quantify the interactions between modalities. Initially, we focus on pairwise interactions, estimating mutual information, and then extend this approach to larger-scale multimodal systems, where we quantify multivariate information. This extension enables a deeper analysis of complex multimodal systems. Finally, we question the assumption that paired modalities are always available, as observability issues may lead to unpaired data. To address this, we introduce a method for constructing multimodal joint distributions starting from unpaired modalities, based on the principle of information maximization. The thesis is organized as follows:

- Chapter 2 describes the fundamental concepts to provide a background for the rest of the thesis. It presents a brief introduction to generative modeling methods in machine

learning, with a focus on Diffusion models.

- Chapter 3 presents the first contribution of the thesis, where we address the challenge of multimodal generative modeling, a domain traditionally dominated by multimodal VAEs. We begin by analyzing this family of models and examining their limitations, which often lead to suboptimal performance and trade-offs. Motivated by these limitations and inspired by the success of diffusion models, we propose a new method, MLD, a novel approach that utilizes a set of independently trained, unimodal deterministic autoencoders. The latent variables from each autoencoder are concatenated into a common latent space, which is then fed into a masked diffusion model for generative modeling. Additionally, we propose a new multi-time training method to learn the conditional score network for multimodal diffusion. Our approach is designed to overcome the limitations of prior work, achieving state-of-the-art performance in extensive experimental comparisons. We then demonstrate how MLD can be applied in the automotive domain, presenting a paradigm to enhance sensor robustness in automotive systems. As a use case, we apply MLD to improve automotive night vision by integrating information from automotive sensors, such as LiDar and RaDar, to enhance the camera modality.

- Chapter 4 introduces a new method for estimating MI between random variables. We show that the KL divergence can be expressed as the difference of score functions, utilizing the Fokker–Planck equation. This proof leads to the same expression as in our original paper (Franzese, Bounoua, and Michiardi, 2024), where Girsanov Theorem (Øksendal, 2003) was applied within a different mathematical framework. The KL estimator is then used to construct an entropy estimator as a byproduct. We propose a general framework for measuring MI, offering two approaches: one using conditional diffusion processes and the other employing joint diffusion processes to simultaneously model two random variables. Our experimental results show that our method outperforms existing alternatives, especially for challenging distributions. Furthermore, our method passes MI self-consistency tests, such as data processing and additivity under independence, which are problematic for current methods. Finally, we demonstrate how pre-trained text-to-image models can be used to compute MI between input modalities, aiding the analysis of the generative properties of these models.

- In Chapter 5, we focus on studying multimodal systems described by multivariate information. Since such systems may involve more than two random variables, mutual information alone is a limited tool. A key concept here is information synergy and redundancy, which are crucial for understanding higher-order dependencies be-

tween variables. One of the most prominent and versatile measures for capturing multivariate interactions is O-information, which quantifies the synergy-redundancy balance in these systems. In this chapter, we introduce SΩI, a method for computing O-information without restrictive assumptions about the number of modalities, leveraging a unified model. Our experiments validate this approach on synthetic data and demonstrate its effectiveness in a real-world use case.

- Chapter 6 investigates the case where multimodal data are unpaired, presenting a significant challenge in learning a joint distribution. A prominent approach to address the modality coupling problem is MEC, which aims to minimize the joint entropy while satisfying constraints on the marginals. While MEC has mainly been studied for discrete distributions, we extend it to continuous distributions and present a relaxed version. We then introduce a novel method, DDMEC, to solve the MEC problem using diffusion models that approximate and minimize the joint entropy. Our approach employs a cooperative scheme within a reinforcement learning framework, while ensuring that the relaxed marginal constraints are satisfied. We empirically demonstrate that DDMEC is a general method, easily applied to challenging tasks such as unsupervised single-cell multi-omics data alignment and unpaired image translation, outperforming alternatives.

  Chapter 7 summarizes the contributions presented in this thesis, discusses their impact, and concludes with an outlook on potential future research directions.

Publications The works in this thesis were done in collaboration with colleagues, and have been mainly peer-reviewed by program committees in top-tier conferences and journals.

- Chapter 3 is based on the following publications:
  - Mustapha Bounoua, Giulio Franzese, and Pietro Michiardi (2023). "Masked Multi-time Diffusion for Multi-modal Generative Modeling". In: *Neural Information Processing Systems (NeurIPS) 2023 Workshop on Diffusion Models*. New Orleans, US
  - Mustapha Bounoua, Giulio Franzese, and Pietro Michiardi (2024a). "Multi-modal latent diffusion". In: *Entropy* 26.4, p. 320
  - Mustapha Bounoua et al. (2024). "Enhancing Sensor Robustness in Automotive Systems: A Multimodal Generative Approach". In: *SIA-Vision 2024*. Paris, France
- Chapter 4 is based on the following publications while revisiting the mathematical framework and providing an alternative proof:

  - Giulio Franzese, Mustapha Bounoua, and Pietro Michiardi (2024). "MINDE: Mutual Information Neural Diffusion Estimation". In: *ICLR 2024, The Twelfth International Conference on Learning Representations*. Vienna, Austria

- Chapter 5 is based on the following publications:

  - Mustapha Bounoua, Giulio Franzese, and Pietro Michiardi (2024b). "SΩI: Score-based O-INFORMATION Estimation". In: *ICML 2024, 41st International Conference on Machine Learning*. IEEE. Vienna, Austria

- Chapter 6 is based on the following publications:

  - Mustapha Bounoua, Giulio Franzese, and Pietro Michiardi (2025). "Learning to Match Unpaired Data with Minimum Entropy Coupling". In: *ICML 2025, 42nd International Conference on Machine Learning*. Vancouver, Canada

Chapter 2

Background

In this chapter, we establish the foundational groundwork for the thesis by providing succinct introductions to the key concepts of generative modeling and multimodality, which form the basis for the discussions presented in the subsequent chapters.

2.1 Generative Modeling

2.1.1 What is a generative model ?

To define generative modeling, we build upon well-established definitions provided in seminal works such as (Goodfellow, Bengio, and Courville, 2016) (Deep Learning), (Murphy, 2012) (Machine Learning: A Probabilistic Perspective), and (Bishop, 2006) (Pattern Recognition and Machine Learning). These sources agree on a central feature of generative models which is : The ability to capture the underlying probability distribution of the data which enables the generation of new, plausible samples.

Definition 1. *The objective of generative modeling is to infer the underlying data distribution* $p_{data}(\mathbf{x})$ *from a given dataset. To accomplish this, we construct a parametric model* $p_\theta(\mathbf{x})$*, with parameters* θ*, such that*

$$p_\theta(\mathbf{x}) \approx p_{data}(\mathbf{x}). \tag{2.1}$$

This model not only enables the evaluation of the probability associated with any data point, but also facilitates the generation of new samples by drawing from $p_\theta(\mathbf{x})$*.*

Given a distance metric $D(p, q)$ that quantifies the discrepancy between $p_\theta(\mathbf{x})$ and $p_{\text{data}}(\mathbf{x})$, the optimal parameter θ^* can be obtained by minimizing the following optimization

problem:

$$\theta^* = \arg\min_{\theta} D\big(p_{\text{data}}(\mathbf{x}),\, p_\theta(\mathbf{x})\big). \tag{2.2}$$

Popular instances of such divergence measures include the family of f-divergences (Csiszár, 1967), among which the Kullback–Leibler divergence (KL) divergence is widely used. The KL between two probability distributions $p_{\text{data}}(\mathbf{x})$ and $q(\mathbf{x})$ is defined as:

$$\mathbb{KL}\left(p_{\text{data}} \parallel q\right) = \int p_{\text{data}}(\mathbf{x}) \log \frac{p_{\text{data}}(\mathbf{x})}{q(\mathbf{x})}\, d\mathbf{x} = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}\left[\log \frac{p_{\text{data}}(\mathbf{x})}{q(\mathbf{x})}\right]. \tag{2.3}$$

Using KL divergence in the optimization problem Eq. (2.2), we obtain:

$$\arg\min_{\theta} \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}\left[\log \frac{p_{\text{data}}(\mathbf{x})}{p_\theta(\mathbf{x})}\right] = \arg\min_{\theta} \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}\left[\log p_{\text{data}}(\mathbf{x}) - \log p_\theta(\mathbf{x})\right] \tag{2.4}$$

$$= \arg\min_{\theta} \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}\left[-\log p_\theta(\mathbf{x})\right] + \text{const.} \tag{2.5}$$

The term $\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})} \log p_{\text{data}}(\mathbf{x})$ is considered constant with respect to the model parameters θ. This recovers the Maximum Likelihood Estimation (MLE) training widely adopted in machine learning.

2.1.2 Families of generative models

A fundamental requirement for any valid probability distribution $p_\theta(\mathbf{x})$ is that it satisfies two properties for all values of θ:

1. **Non-negativity:** $p_\theta(\mathbf{x}) \geq 0$ for all $\mathbf{x}$.
2. **Normalization:** $\int p_\theta(\mathbf{x})\, d\mathbf{x} = 1$.

Ensuring the non-negativity is relatively easy to satisfy. The normalization constraint is more challenging and is often computationally intractable, especially for complex models over high-dimensional data such as images, audio, or text. This difficulty in normalization has driven the development of several distinct families of generative models, each addressing the issue in different ways. Generative models can be broadly categorized into two classes (Wu, Gao, and Zha, 2021):

- **Implicit Density Models:** These models do not yield an explicit likelihood function; instead, they model only the generative process. For instance, in Generative Adversarial Networks (GANs) (Goodfellow et al., 2014), the generator is trained to produce samples evaluated using a discriminator to ensure faithfulness to data.

- **Explicit Density Models:** These models provide an explicit likelihood function that can be evaluated and optimized—typically via maximum likelihood estimation. It is possible to enforce normalization exactly by designing model architectures in special ways or by approximation. Examples include Energy based Models (EBMs) (Ackley, Hinton, and Sejnowski, 1985), autoregressive models (Bengio and Bengio, 1999), Variational Autoencoders (VAEs) (Kingma and Welling, 2013), Normalizing Flows (NFs) (Rezende and Mohamed, 2015) and Score based Diffusion Models (DMs) (Song et al., 2021; Ho, Jain, and Abbeel, 2020).

In the next section of this chapter, we provide a short overview on these methods while putting the focus on DMs and VAEs.

2.1.3 Multimodal generative models

Definition 2. *A* multimodal generative model *extends the concept of generative modeling to the case where the data come from multiple modalities, denoted as* $\mathbf{X}^1, \mathbf{X}^2, \ldots, \mathbf{X}^M$. *Such a model defines a joint probability distribution over these modalities:*

$$p_\theta(\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^M) \approx p_{data}(\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^M). \tag{2.6}$$

In the case of latent-variable models, this joint distribution is often factorized via a shared latent variable $\mathbf{z}$, which captures the common structure among modalities:

$$p(\mathbf{x}^1, \ldots, \mathbf{x}^M) = \int p(\mathbf{z}) \prod_{i=1}^{M} p(\mathbf{x}^i \mid \mathbf{z})\, d\mathbf{z}. \tag{2.7}$$

Alternatively, one may model the joint distribution directly without introducing an explicit latent variable.

Sampling from multimodal generative models is inherently more flexible than in the unimodal setting due to the multiple ways in which the joint distribution can be factorized into marginals and conditionals. Specifically, for any partitioning of the modalities into two disjoint subsets $A_1, A_2 \subseteq \{1, \ldots, M\}$ such that $A_1 \cup A_2 = \{1, \ldots, M\}$ and $A_1 \cap A_2 = \emptyset$, the joint distribution can be expressed as:

$$p(\mathbf{x}^1, \ldots, \mathbf{x}^M) = p(\mathbf{x}^{A_1})\, p(\mathbf{x}^{A_2} \mid \mathbf{x}^{A_1}), \tag{2.8}$$

where $p(\mathbf{x}^{A_1})$ represents the marginal distribution over a subset of modalities A_1, and

$p(\mathbf{x}^{A_2} \mid \mathbf{x}^{A_1})$ models the conditional dependence between the remaining modalities.

When focusing on latent-variable-based generative models, (Shi et al., 2019) introduces the following desirable properties of multimodal generative models:

- **Coherent Joint Generation**: The model should be able to generate all modalities simultaneously $\mathbf{x}^1, \ldots, \mathbf{x}^M \sim p_\theta(\mathbf{x}^1, \ldots, \mathbf{x}^M)$, while ensuring that the generated modalities are semantically consistent and faithful to the training distribution, where they are naturally paired. In line with the properties of multimodal data, this ensures that the redundant information between the modalities is preserved.

- **Coherent Cross-Generation**: Cross-modal generation can be viewed as sampling from the conditional distribution $\mathbf{x}^{A_2} \sim p(\mathbf{x}^{A_2} \mid \mathbf{x}^{A_1} = \hat{\mathbf{x}}^{A_1})$, where access is granted to an observed set of modalities $\hat{\mathbf{x}}^{A_1} \sim p_{\text{data}}(\mathbf{x}^{A_1})$. This approach ensures that the inherent *connectivity* between modalities is preserved.

- **Synergy**: the generative capabilities are enhanced when additional modalities are observed, leveraging additional complementary information due to the *interaction* between modalities.

- **Latent Factorization**: the model should implicitly leverage the disentanglement of information into shared features common to all modalities from private features that are specific to individual modalities.

2.2 Taxonomy of generative models

In this section, we present a structured overview of generative models, which are a class of machine learning models designed to generate new data samples that resemble a given dataset.

2.2.1 Generative Adversarial Networks

GANs (Goodfellow et al., 2014) are a class of implicit generative models that learn to generate samples from a target data distribution $p_{\text{data}}(\mathbf{x})$ by optimizing an adversarial game between two neural networks: a generator and a discriminator.

- The generator G_θ is a deterministic function that maps a latent variable $\mathbf{z} \sim p(\mathbf{z})$,

typically drawn from a simple prior (e.g., standard normal), to the data space:

$$\mathbf{x} = G_\theta(\mathbf{z}) \tag{2.9}$$

- The discriminator D_ϕ is trained to distinguish between real data samples $\mathbf{x} \sim p_{\text{data}}(\mathbf{x})$ and generated samples $G_\theta(\mathbf{z}) \sim p_\theta(\mathbf{x})$.

The original GAN framework formulates a minimax objective:

$$\min_\theta \max_\phi \underbrace{\mathbb{E}_{\mathbf{x}\sim p_(\mathbf{x})}\left[\log D_\phi(\mathbf{x})\right]}_{(1)} + \underbrace{\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}\left[\log(1 - D_\phi(G_\theta(\mathbf{z})))\right]}_{(2)} \tag{2.10}$$

Here: (1) encourages the discriminator to assign high probability to real data, and (2) penalizes the discriminator when it fails to identify generated samples as fake.

GANs are trained by alternately updating the discriminator ϕ to improve its classification accuracy and the generator θ to fool the discriminator by generating more realistic samples. At convergence, the generator ideally produces data indistinguishable from real samples. Despite their empirical success especially in image domains, GANs suffer from training instabilities and issues like mode collapse. Several variants like WGANs (Arjovsky, Chintala, and Bottou, 2017), LSGANs(Mao et al., 2017), and StyleGAN (Karras, Laine, and Aila, 2019) have been proposed to improve training dynamics and sample diversity.

2.2.2 Energy based Models

Since exact normalization is generally intractable, we rely on approximation methods. Energy-based models (EBMs) are one family of generative models that perform approximate normalization using techniques such as Monte Carlo sampling. EBMs define a probability density over the data using an energy function. The model is expressed as:

$$p_\theta(\mathbf{x}) = \frac{\exp\Big(-E_\theta(\mathbf{x})\Big)}{Z_\theta}, \tag{2.11}$$

where $E_\theta(\mathbf{x})$ is the energy function, which assigns lower energy to more plausible data, and Z_θ is the partition function that ensures proper normalization:

$$Z_\theta = \int \exp\Big(-E_\theta(\mathbf{x})\Big)\mathrm{d}\mathbf{x}. \tag{2.12}$$

The model is trained by maximizing the expected log-likelihood of the data:

$$\theta^* = \arg\max_\theta \ \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}\Big[\log p_\theta(\mathbf{x})\Big] = \arg\max_\theta \ \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}\Big[-E_\theta(\mathbf{x})\Big] - \log Z_\theta. \quad (2.13)$$

Because $\log Z_\theta$ is intractable to compute exactly, its gradient are typically approximated using methods like MCMC or contrastive divergence. For data generation, samples from $p_\theta(\mathbf{x})$ are drawn using Markov chain Monte Carlo (MCMC) techniques. This approach leverages the differences in energy between samples, thereby bypassing the need to compute the partition function Z_θ directly when generating new data.

2.2.3 Auto-regressive models

Autoregressive models are based on the chain rule of probability, which states that any high-dimensional probability distribution can be factorized into a product of one-dimensional conditional distributions. Exploiting this idea, autoregressive models define the data distribution as:

$$p_\theta(\mathbf{x}) = \prod_{i=1}^{d} p_\theta\big(\mathbf{x}_i \mid \mathbf{x}_{<i}\big), \quad (2.14)$$

where d is the dimensionality of $\mathbf{x}$, $\mathbf{x}_i$ denotes the i-th element of $\mathbf{x}$, and $\mathbf{x}_{<i}$ represents the set $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{i-1}\}$.

Each conditional distribution $p_\theta\big(\mathbf{x}_i \mid \mathbf{x}_{<i}\big)$ is chosen to be a normalized parametric distribution, often implemented using a deep neural network. This autoregressive factorization guarantees that $p_\theta(\mathbf{x})$ is exactly normalized:

$$\int p_\theta(\mathbf{x})\,\mathrm{d}\mathbf{x} = \int \prod_{i=1}^{d} p_\theta\big(\mathbf{x}_i \mid \mathbf{x}_{<i}\big)\,\mathrm{d}\mathbf{x} = 1. \quad (2.15)$$

These models are typically trained using maximum likelihood estimation (MLE):

$$\theta^* = \arg\max_\theta \ \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}\Big[\log p_\theta(\mathbf{x})\Big]. \quad (2.16)$$

For data generation, autoregressive models employ ancestral sampling. Starting with $\mathbf{x}_1$ drawn from $p_\theta(\mathbf{x}_1)$, each subsequent element $\mathbf{x}_i$ is sampled from $p_\theta\big(\mathbf{x}_i \mid \mathbf{x}_1, \ldots, \mathbf{x}_{i-1}\big)$, ensuring that the entire sample $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_d)$ is generated sequentially.

Note that this factorization requires a predetermined ordering of the data dimensions. While this is natural for sequential data (e.g., text), it can be less straightforward in other domains.

2.2.4 Normalizing Flows

NFs are generative models that enable exact likelihood computation through invertible transformations. Let $\mathbf{z} \in \mathbb{R}^d$ be a continuous latent variable following a tractable prior distribution $\pi(\mathbf{z})$, which allows for efficient density evaluation and fast sampling (Gaussian distribution). NFs parameterize a smooth, invertible function $\mathbf{f}_\theta : \mathbb{R}^d \to \mathbb{R}^d$. Then, the observed data $\mathbf{x}$ is obtained by transforming the latent variable:

$$\mathbf{x} = \mathbf{f}_\theta(\mathbf{z}) \tag{2.17}$$

The transformation induces a new distribution $p_\theta(\mathbf{x})$, derived from the prior $\pi(\mathbf{z})$ via the change of variables formula:

$$p_\theta(\mathbf{x}) = \pi(\mathbf{f}_\theta^{-1}(\mathbf{x})) \left| \det \left(\frac{\partial \mathbf{f}_\theta^{-1}}{\partial \mathbf{x}} \right) \right| \tag{2.18}$$

When the determinant of the Jacobian $\left| \det \left(\frac{\partial \mathbf{f}_\theta^{-1}}{\partial \mathbf{x}} \right) \right|$ is tractable, normalizing flows allow exact evaluation of the data likelihood.

The objective is to maximize the log-likelihood of the observed data:

$$\log p_\theta(\mathbf{x}) = \log \pi(\mathbf{f}_\theta^{-1}(\mathbf{x})) + \log \left| \det \left(\frac{\partial \mathbf{f}_\theta^{-1}}{\partial \mathbf{x}} \right) \right| \tag{2.19}$$

To generate new samples data $\mathbf{x} \sim p_\theta(\mathbf{x})$, one first sample from the prior distribution samples $\mathbf{z} \sim \pi(\mathbf{z})$ then use the parameterized function $\mathbf{x} = \mathbf{f}_\theta(\mathbf{z})$.

However, parameterizing an invertible function with a tractable Jacobian determinant is non-trivial in deep neural networks. Therefore, normalizing flows rely on specific architectures such as coupling layers and autoregressive flows to ensure both invertibility and efficient computation. Notable flow-based models include NICE (Dinh, Krueger, and Bengio, 2014), RealNVP (Dinh, Sohl-Dickstein, and Bengio, 2016), Glow (Kingma and Dhariwal, 2018), and Masked Autoregressive Flows (Papamakarios, Pavlakou, and Murray, 2017).

2.2.5 Variational Autoencoders

VAEs are probabilistic generative models that introduce a latent variable $\mathbf{z}$ to facilitate the representation of the modeling of $p_{\text{data}}(\mathbf{x})$. It consists of several components:

- A prior $p(\mathbf{z})$, which allows fast and efficient sampling.

- The encoder $q_\phi(\mathbf{z} \mid \mathbf{x})$, which maps input data $\mathbf{x}$ to a latent space $\mathbf{z}$.
- The decoder $p_\theta(\mathbf{x} \mid \mathbf{z})$, which reconstructs data from the latent representation.

Then the learned generative model can be modeled as:

$$p_\theta(\mathbf{x}) = \int p(\mathbf{z})\, p_\theta(\mathbf{x} \mid \mathbf{z})\, \mathrm{d}\mathbf{z} \tag{2.20}$$

The VAE framework is grounded in variational inference, aiming to approximate the true posterior distribution $p(\mathbf{z}|\mathbf{x})$ with a simpler, parameterized distribution $q_\phi(\mathbf{z}|\mathbf{x})$. The objective is to maximize the Evidence Lower Bound (ELBO), which can be obtained using the Jensen inequality and is used to train a VAE:

$$\log p_\theta(\mathbf{x}) \geq \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right]}_{(1)} - \underbrace{\mathbb{KL}\left(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z})\right)}_{(2)} \tag{2.21}$$

Here: (1) is the reconstruction term, ensuring the decoder reconstructs $\mathbf{x}$ accurately from $\mathbf{z}$, and (2) is the regularization term, penalizing the divergence between the approximate posterior and the prior distribution $p(\mathbf{z})$.

The *reparameterization trick* is employed to enable backpropagation through the stochastic sampling process:

$$\mathbf{z} = \mu_\phi(\mathbf{x}) + \sigma_\phi(\mathbf{x}) \cdot \epsilon, \quad \epsilon \sim \mathcal{N}(0, \mathbb{I}) \tag{2.22}$$

This formulation allows for efficient training of VAEs using stochastic gradient descent. For a comprehensive discussion on VAEs, refer to (Kingma and Welling, 2013). To generate new samples data $\mathbf{x} \sim p_\theta(\mathbf{x})$, one first samples from the prior distribution $\hat{\mathbf{z}} \sim p(\mathbf{z})$ then use the decoder to generate samples $\mathbf{x} \sim p_\theta(\mathbf{x}|\mathbf{z} = \hat{\mathbf{z}})$.

2.2.6 Score based Diffusion Models

Score-based diffusion models (Sohl-Dickstein et al., 2015; Ho, Jain, and Abbeel, 2020; Song et al., 2021) represent a family of explicit generative models that model data generation as a gradual denoising process. The core idea is to define a forward process that progressively perturbs data until it reaches a simple prior (e.g., a standard Gaussian), and a reverse process that recovers the original data from the noise. We first present the continuous-time formulation, which provides a unifying framework, then show how discrete-time models such as Denoising Diffusion Probabilistic Models (DDPM) and Denoising Diffusion Implicit

Models (DDIM) arise as special cases.

Continuous-Time Formulation

Let $\mathbf{x}_0 \sim p_{\text{data}}(\mathbf{x})$ denote a realization of the random variable $\mathbf{X}_0$ from the data distribution. The forward process is defined by an Itô Stochastic Differential Equation (SDE) (Song et al., 2021):

$$\mathrm{d}\mathbf{X}_t = f(\mathbf{X}_t, t)\,\mathrm{d}t + g(t)\,\mathrm{d}\mathbf{W}_t, \quad t \in [0, T] \tag{2.23}$$

where $f(\mathbf{X}_t, t)$ is the drift coefficient, $g(t)$ is the diffusion coefficient, and $\mathbf{W}_t$ is a standard Wiener process (Song et al., 2021). This SDE progressively transforms the data distribution $p_0(\mathbf{x}) = p_{\text{data}}(\mathbf{x})$ into a known prior distribution $p_T(\mathbf{x})$ (typically a Gaussian $\mathcal{N}(0, \mathbf{I})$) by time T.

The evolution of the probability density $p_t(\mathbf{x})$ over time is governed by the Fokker–Planck equation (Risken, 1996):

$$\frac{\partial p_t(\mathbf{x})}{\partial t} = -\nabla_{\mathbf{x}} \cdot [f(\mathbf{x}, t)p_t(\mathbf{x})] + \frac{1}{2}g(t)^2 \Delta_{\mathbf{x}} p_t(\mathbf{x}), \tag{2.24}$$

where $\nabla_{\mathbf{x}}$ denotes the gradient and $\Delta_{\mathbf{x}}$ denotes the Laplacian.

The forward process defines transition kernels $p_{0t}(\mathbf{x}_t|\mathbf{x}_0)$ that describe the conditional distribution of the realization $\mathbf{x}_t$ of the random variable $\mathbf{X}_t$ given the initial sample $\mathbf{x}_0$. For many diffusion processes of interest, these transition kernels have closed-form Gaussian expressions:

$$p_{0t}(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \mu_t(\mathbf{x}_0), \Sigma_t), \tag{2.25}$$

where $\mu_t(\mathbf{x}_0)$ and Σ_t depend on the specific forms of $f(\mathbf{x}_t, t)$ and $g(t)$.

The Reverse Process and Score-Based Modeling

To generate data from $p_0(\mathbf{x})$, we simulate a reverse-time SDE, which has the following form (Anderson, 1982):

$$\mathrm{d}\mathbf{X}_t = \left[f(\mathbf{X}_t, t) - g(t)^2 \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)\right] \mathrm{d}t + g(t)\,\mathrm{d}\bar{\mathbf{W}}_t, \quad t \in [T, 0], \tag{2.26}$$

where $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$ is the **score function** evaluated at a realization $\mathbf{x}_t$, and $\mathrm{d}\bar{\mathbf{W}}_t$ is a Wiener process running backwards in time.

In practice, the true score function is unknown and must be approximated with a neural

network $s_\theta(\mathbf{x}_t, t)$, trained via score matching (Song et al., 2021):

$$\mathcal{L}_{\text{SM}}(\theta) = \frac{1}{2}\int_0^T \mathbb{E}_{\mathbf{x}_t \sim p_t(\mathbf{x}_t)} \left[\lambda(t) \left\|\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) - s_\theta(\mathbf{x}_t, t)\right\|^2\right] \mathrm{d}t, \tag{2.27}$$

where $\lambda(t)$ is a positive weighting function.

However, since $\nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t)$ is intractable, denoising score matching can be used. It can be shown that Eq. (2.27) is equivalent, up to a constant, to the denoising score matching objective (Song et al., 2021):

$$\mathcal{L}_{\text{DSM}}(\theta) = \frac{1}{2}\int_0^T \mathbb{E}_{\substack{\mathbf{x}_0 \sim p_0(\mathbf{x}_0) \\ \mathbf{x}_t \sim p_{0t}(\mathbf{x}_t|\mathbf{x}_0)}} \left[\lambda(t) \left\|\nabla_{\mathbf{x}_t} \log p_{0t}(\mathbf{x}_t|\mathbf{x}_0) - s_\theta(\mathbf{x}_t, t)\right\|^2\right] \mathrm{d}t. \tag{2.28}$$

This approach is particularly effective because it allows training without requiring explicit density evaluation, leveraging only the score function.

Indeed, the generated data distribution $p_\theta(\mathbf{x})$ is close (in the KL sense) to the true density, as described by (Song et al., 2021; Franzese et al., 2023):

$$\mathbb{KL}\left(p_0(\mathbf{x}) \parallel p_\theta(\mathbf{x})\right) \leq \int_0^T \mathbb{E}_{\mathbf{x}_t \sim p_t(\mathbf{x}_t)} \left[g^2(t) \left\|\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) - s_\theta(\mathbf{x}_t, t)\right\|^2\right] \mathrm{d}t + \mathbb{KL}\left(p_T(\mathbf{x}) \parallel \rho(\mathbf{x})\right), \tag{2.29}$$

where the first term corresponds to the score matching objective Eq. (2.27) with $\lambda(t) = g^2(t)$.

We now introduce two specializations of SDEs used in score-based generative modeling.

Variance preserving SDE (VPSDE) (Song et al., 2021):

$$\mathrm{d}\mathbf{X}_t = -\frac{1}{2}\beta(t)\mathbf{X}_t\, dt + \sqrt{\beta(t)}\,\mathrm{d}\mathbf{W}_t \tag{2.30}$$

where $\beta(t)$ is a noise schedule that typically increases from 0 to a large value. For this SDE, the transition kernel is:

$$p_{0t}(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_t; \mathbf{x}_0 e^{-\frac{1}{2}\int_0^t \beta(s)\, ds}, \mathbf{I}\left(1 - e^{-\int_0^t \beta(s)\, ds}\right)\right).$$

The VPSDE maintains an approximately constant signal-to-noise ratio throughout the diffusion process.

Variance exploding SDE (VESDE) (Song et al., 2021):

$$\mathrm{d}\mathbf{X}_t = 0 \cdot \mathrm{d}t + \sigma(t)\,\mathrm{d}\mathbf{W}_t \tag{2.31}$$

where $\sigma(t)$ is an increasing function of time. The transition kernel for this SDE is:

$$p_{0t}(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \mathbf{x}_0, \sigma^2(t)\mathbf{I}).$$

The VESDE allows the variance to grow unbounded, which is beneficial for modeling data with a high dynamic range.

Both SDEs can be reversed using the same reverse-time formulation and trained via the same score-matching loss. The primary differences lie in the choice of the marginal distribution $p_t(\mathbf{x}_t)$, the noise scaling behavior, and their sampling characteristics. Recent studies have investigated optimal noise schedule designs (Nichol and Dhariwal, 2021; Kingma et al., 2021), showing that well-tuned schedules can significantly enhance both training stability and sample quality.

Sampling Techniques

The reverse SDE can be simulated using numerical solvers such as the Euler–Maruyama method (Song et al., 2021):

$$\mathbf{x}_{t-\Delta t} = \mathbf{x}_t - \left[f(\mathbf{x}_t, t) - g^2(t)s_\theta(\mathbf{x}_t, t)\right]\Delta t + g(t)\sqrt{\Delta t}\,\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}). \tag{2.32}$$

Alternatively, generation can be performed using the corresponding probability flow Ordinary Differential Equation (ODE):

$$\frac{\mathrm{d}\mathbf{X}_t}{\mathrm{d}t} = f(\mathbf{X}_t, t) - \frac{1}{2}g^2(t)s_\theta(\mathbf{x}_t, t), \tag{2.33}$$

which enables deterministic sampling without injecting noise during the generation process.

Predictor–Corrector Sampling (Song et al., 2021) alternates between two steps: a deterministic predictor step (via ODE solver) and a stochastic corrector step (via Langevin dynamics). This hybrid approach often achieves better sample quality than pure ODE or SDE-based sampling, at the cost of additional computation.

The Langevin dynamics step performs MCMC sampling to better approximate the true distribution $p_t(\mathbf{x}_t)$, helping to compensate for errors introduced by the learned score function. Recent works have proposed enhanced sampling techniques, such as momentum-based samplers (Dockhorn, Vahdat, and Kreis, 2022), dynamic programming methods (Watson et al., 2021), and adaptive computational schemes (Tang et al., 2024).

Discrete-Time Formulations

While the continuous-time framework provides a unifying theoretical foundation, most practical implementations use discrete-time formulations. Here we show how popular discrete-time models connect to the continuous framework.

DDPM

DDPM (Ho, Jain, and Abbeel, 2020; Sohl-Dickstein et al., 2015) can be interpreted as a specific discretization of the VPSDE (Song et al., 2021). The forward process is a fixed-length Markov chain with T steps:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\,\mathbf{x}_{t-1}, \beta_t\mathbf{I}), \tag{2.34}$$

where $\{\beta_t\}_{t=1}^T$ is a pre-defined variance schedule. The marginal distribution at time t given $\mathbf{x}_0$ is:

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}), \quad \text{where} \quad \bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s). \tag{2.35}$$

The reverse process is modeled using a neural network that predicts the added noise ϵ (Ho, Jain, and Abbeel, 2020). The forward sampling process can be written as:

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\epsilon, \quad \epsilon \sim \mathcal{N}(0, \mathbf{I}). \tag{2.36}$$

During training, the model learns to predict the noise ϵ that was added to the data during the forward process. Given a clean sample $\mathbf{x}_0$, a noise level t, and sampled noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$, we construct a noisy sample $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\epsilon$. The neural network $\epsilon_\theta(\mathbf{x}_t, t)$ is then trained to recover the original noise via the following loss:

$$\mathcal{L}_{\text{DDPM}}(\theta) = \mathbb{E}_{\mathbf{x}_0,\epsilon,t}\left[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\|^2\right]. \tag{2.37}$$

This objective is known as the noise-prediction loss and corresponds to a simplified form of the variational lower bound (VLB) (Ho, Jain, and Abbeel, 2020). At inference time, the learned model is used to estimate the reverse transition $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$, allowing the denoising process to be run backward from pure noise to a data sample. The reverse distribution is defined as:

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t\mathbf{I}), \tag{2.38}$$

where the mean $\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0)$ is derived analytically from the forward process, and $\tilde{\beta}_t$ controls the variance. Following (Nichol and Dhariwal, 2021), $\tilde{\beta}_t$ can either be parameterized and learned, or set heuristically as in the original formulation by (Ho, Jain, and Abbeel, 2020).

DDIM

DDIM (Song, Meng, and Ermon, 2020) generalizes DDPM by introducing non-Markovian, deterministic sampling that preserves the same marginal distributions $q(\mathbf{x}_t|\mathbf{x}_0)$ as DDPM. It can be interpreted as solving a discretized version of the probability flow ODE associated with the continuous-time SDE.

Given a noise schedule $\{\alpha_t\}_{t=1}^T$, the DDIM update rule is:

$$\mathbf{x}_{t-1} = \sqrt{\alpha_{t-1}} \left(\frac{\mathbf{x}_t - \sqrt{1-\alpha_t}\, \epsilon_\theta(\mathbf{x}_t, t)}{\sqrt{\alpha_t}} \right) + \sqrt{1-\alpha_{t-1}}\, \epsilon_\theta(\mathbf{x}_t, t). \tag{2.39}$$

DDIM also introduces a family of inference distributions q_σ parameterized by σ_t, all of which preserve the marginal $q(\mathbf{x}_t|\mathbf{x}_0)$:

$$q_\sigma(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_{t-1}; \sqrt{\alpha_{t-1}}\mathbf{x}_0 + \sqrt{1-\alpha_{t-1}-\sigma_t^2} \cdot \frac{\mathbf{x}_t - \sqrt{\alpha_t}\mathbf{x}_0}{\sqrt{1-\alpha_t}}, \sigma_t^2 \mathbf{I} \right). \tag{2.40}$$

Setting $\sigma_t = 0$ results in a deterministic sampling procedure, while setting $\sigma_t^2 = \tilde{\beta}_t$ recovers the original DDPM formulation. This flexibility enables a trade-off between sample quality and inference speed, with deterministic DDIM sampling offering significantly faster generation while maintaining high sample fidelity.

Conditional Diffusion Models

Diffusion models can be conditioned on additional variable $\mathbf{Y}$ to control the generation process. Conditional diffusion models have been successful in text-to-image generation (Rombach et al., 2022), image-to-image translation (Saharia et al., 2022), and many other conditional generation tasks.

Classifier Guidance (Dhariwal and Nichol, 2021): An external classifier $p_\phi(\mathbf{y}|\mathbf{x}_t)$ can guide the diffusion process. Using Bayes' rule, we can derive:

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{y}) = \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log p_\phi(\mathbf{y}|\mathbf{x}_t) \tag{2.41}$$

This modifies the reverse SDE to:

$$\mathrm{d}\mathbf{x}_t = \left[f(\mathbf{X}_t, t) - g(t)^2 \left[\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) - \omega \nabla_{\mathbf{x}_t} \log p_\phi(\mathbf{y}|\mathbf{x}_t) \right] \right] \mathrm{d}t + g(t)\, \mathrm{d}\bar{\mathbf{W}}_t \tag{2.42}$$

where ω controls the guidance strength. Higher values of ω produce samples that are more strongly aligned with the conditioning class or attribute, often at the cost of diversity.

The classifier $p_\phi(\mathbf{y}|\mathbf{x}_t)$ must be trained to classify noisy samples at different diffusion timesteps. Though effective, classifier guidance has several limitations: it requires training a separate classifier, can amplify model biases, and may struggle with complex conditioning information like text.

Classifier-Free Guidance (CFG) (Ho and Salimans, 2022): Instead of training a separate classifier, two score functions are trained jointly: an unconditional model $s_\theta(\mathbf{x}_t, t)$ and a conditional model $s_\theta(\mathbf{x}_t, t, \mathbf{y})$. During training, the conditioning information is randomly dropped with some probability to enable the model to handle both conditional and unconditional generation. During sampling, a weighted combination is used:

$$\tilde{s}_\theta(\mathbf{x}_t, t, \mathbf{y}) = s_\theta(\mathbf{x}_t, t) + \gamma \cdot (s_\theta(\mathbf{x}_t, t, \mathbf{y}) - s_\theta(\mathbf{x}_t, t)) \tag{2.43}$$

where γ is the guidance scale. Intuitively, the term $s_\theta(\mathbf{x}_t, t, \mathbf{y}) - s_\theta(\mathbf{x}_t, t)$ represents the "direction" in which the noisy data should be modified to better align with the conditioning. Multiplying by $\gamma > 1$ amplifies this direction, leading to stronger conditioning fidelity.

CFG has become the standard for text-to-image diffusion models like Stable Diffusion (Rombach et al., 2022) and DALL-E (Ramesh et al., 2022), as it tends to produce higher-quality and more accurately conditioned samples than both standard conditional diffusion and classifier guidance approaches.

Latent Diffusion Models

Latent Diffusion Models (Rombach et al., 2022) operate in a compressed latent space rather than in input space, significantly reducing computational requirements while preserving generation quality. First, an autoencoder is trained to compress data into a latent space. Then, a diffusion model is trained in the latent space. The diffusion process in latent space follows the same mathematical formulation as in input space, but operates on the compressed representations. Samples are generated by first sampling from the latent diffusion model and then decoding. Operating in a lower-dimensional space reduces memory requirements and speeds up both training and sampling. The autoencoder can learn to discard perceptually irrelevant information, allowing the diffusion model to focus on modeling important structures. This approach has enabled high-resolution generation with reasonable computational requirements, democratizing access to powerful generative AI.

Chapter 3

Multi-modal Generative Modeling

Effective harnessing of multimodal data begins with the development of accurate and expressive generative models capable of capturing the intricate relationships across diverse modalities. Multimodal datasets are increasingly prevalent across a wide range of domains and offer significant potential for developing high-performing and efficient generative models. A dominant approach in this area involves VAEs, which aim to learn unified latent representations that capture the joint structure of heterogeneous data modalities. However, existing approaches suffer from a coherence–quality tradeoff in which models with good generation quality lack generative coherence across modalities and vice versa. In this chapter, we discuss the limitations underlying the unsatisfactory performance of existing methods in order to motivate the need for a different approach. We propose a novel method that uses a set of independently trained and unimodal deterministic autoencoders. Individual latent variables are concatenated into a common latent space, which is then fed to a masked diffusion model to enable generative modeling. We introduce a new multi-time training method to learn the conditional score network for multimodal diffusion. Our methodology substantially outperforms competitors in both generation quality and coherence, as shown through an extensive experimental campaign.

3.1 Introduction

Multi-modal generative modeling is a crucial area of research in machine learning that aims to develop models capable of generating data according to multiple modalities, such as images, text, audio, and more. This is important because real-world observations are often captured in various forms; thus, combining multiple modalities describing the same information can be an invaluable asset. For instance, images and text can provide complementary

information in describing an object, while audio and video can capture different aspects of a scene. Multimodal generative models can help in tasks such as data augmentation (He et al., 2023; Azizi et al., 2023; Sariyildiz et al., 2023), missing modality imputation (Antelmi et al., 2019; Da Silva–Filarder et al., 2021; Zhang et al., 2023; Tran et al., 2017), and conditional generation (Huang et al., 2022; Lee, Ha, and Kim, 2019).

Multimodal models have flourished over the past years and seen tremendous interest from academia and industry, especially in the content creation sector. Whereas most recent approaches focus on specialization, by considering text as a primary input to be associated mainly with images (Rombach et al., 2022; Saharia et al., 2022; Ramesh et al., 2022; Tao et al., 2022; Wu et al., 2022; Nichol et al., 2022; Chang et al., 2023) and videos (Blattmann et al., 2023; Hong et al., 2023; Singer et al., 2022), in this work we target an established literature with more general scope and in which all modalities are considered equally important.

Multi modal generative models aim at *high-quality* data generation, as well as at generative *coherence* across all modalities. These objectives apply to both joint generation of new data and to conditional generation of missing modalities given a disjoint set of available modalities. The predominant literature in this field is based on extensions of the VAE (Kingma and Welling, 2013) to the multimodal domain; initially interested in learning joint latent representation of multimodal data, such works have mostly focused on generative modeling.

In § 3.3, we investigate the limitations of multimodal VAEs and prepare the ground to substantiate a new approach which overcomes the shortcomings in the state of the art. We further investigate the tradeoff (Daunhawer et al., 2022) between generative coherence and quality, and argue that it is intrinsic to all variants of multimodal VAEs. We indicate two root causes of the problem: latent variable collapse (Alemi et al., 2018; Dieng et al., 2019) and information loss due to mixture subsampling. To tackle these issues, in § 3.4 of this work we propose a new approach that uses a set of independent and unimodal *deterministic* autoencoders with the latent variables simply concatenated in a joint latent variable. Joint and conditional generative capabilities are provided by an additional model that learns a probability density associated with the joint latent variable. We propose an extension of score-based diffusion models (Song et al., 2021) to operate on the multimodal latent space. Thus, we derive both forward and backward dynamics that are compatible with the multimodal nature of the latent data. In § 3.4.2, we propose a novel multi-time diffusion process that can both be used for joint and conditional generation. We label our approach Multi-modal Latent Diffusion (MLD).

Our experimental evaluation of MLD in § 3.5 provides compelling evidence of the

superiority of our approach for multimodal generative modeling. We compare MLD to a large variety of VAE-based alternatives on several real-life multimodal datasets in terms of generative quality and both joint and conditional coherence. Our model outperforms alternatives in all possible scenarios, even those that are notoriously difficult because the modalities might be only loosely correlated. We note that recent works have explored the joint generation of multiple modalities (Ruan et al., 2023; Hu et al., 2023); however, such approaches are application-specific, e.g., text-to-image, and essentially only target two modalities. When relevant, we compare our method to additional recent alternatives to multimodal diffusion (Bao et al., 2023; Wesego and Rooshenas, 2023) and show the superior performance of MLD.

3.2 Related works

Multimodal VAEs In short, multimodal VAEs rely on combinations of unimodal VAEs, and the design space mainly consists of the way in which the unimodal latent variables are combined to construct the joint posterior distribution. Early works such as (Wu and Goodman, 2018) adopted a product-of-experts approach, whereas others (Shi et al., 2019) considered a mixture-of-experts approach. While product-based models achieve high generative quality, they suffer in terms of both joint and conditional coherence. This has been found to be due to mis-calibration issues on the part of the experts (Shi et al., 2019; Sutter, Daunhawer, and Vogt, 2021). On the other hand, mixture-based models produce coherent but qualitatively poor samples. A first attempt to address the so-called **coherence–quality tradeoff**(Daunhawer et al., 2022) was represented by the mixture of products of experts approach (Sutter, Daunhawer, and Vogt, 2021). However, recent comparative studies (Daunhawer et al., 2022) have shown that none of the existing approaches fulfill the criteria of both generative quality and coherence. A variety of techniques are aimed at finding a better operating point, such as contrastive learning techniques (Shi et al., 2021), hierarchical schemes (Vasco et al., 2022), total correlation-based calibration of single-modality encoders (Hwang et al., 2021), and different training objectives (Sutter, Daunhawer, and Vogt, 2020). More recently, in (Palumbo, Daunhawer, and Vogt, 2023), explicitly separated shared and private latent spaces were considered as a way to overcome the aforementioned limitations.

Any-to-any multimodality Any-to-any multimodality has been recently studied through the composition of modality-specific diffusion models (Tang et al., 2023) by designing cross-attention and training procedures that allow for arbitrary conditional generation. The work by Tang et al. (2023b) relies on latent interpolation of input modalities, which is akin to

mixture models, and uses it as conditioning signal for individual diffusion models. This is substantially different from the joint nature of the multimodal latent diffusion we present in our work; instead of forcing entanglement through cross-attention between score networks, our model relies on a joint diffusion process whereby modalities naturally co-evolve according to the diffusion process. Another recent work, (Wu et al., 2024), targeted multimodal conversational agents, wherein the strong underlying assumption is to consider one modality, i.e., text, as a guide for the alignment and generation of other modalities. Even if conversational objectives are orthogonal to our work, techniques akin to instruction-following for cross-generation are an interesting illustration of the powerful capabilities of in-context learning on the part of LLMs (Xie et al., 2022; Min et al., 2022).

3.3 Motivation

In this work, we consider multimodal VAEs (Wu and Goodman, 2018; Shi et al., 2019; Sutter, Daunhawer, and Vogt, 2021; Palumbo, Daunhawer, and Vogt, 2023) as the standard modeling approach to tackle both joint and conditional generation of multiple modalities. Our goal here is the need to go beyond such a standard approach in order to overcome limitations that affect multimodal VAEs, which result in a tradeoff between generation quality and generative coherence (Daunhawer et al., 2022; Palumbo, Daunhawer, and Vogt, 2023).

3.3.1 Multimodal ELBO

Consider the random variable $\mathbf{X} = \{\mathbf{X}^1, \dots, \mathbf{X}^M\} \sim p_{data}(\mathbf{x}^1, \dots, \mathbf{x}^M)$, consisting of the set of M modalities sampled from the (unknown) multimodal data distribution p_{data}.

We indicate the marginal distribution of a single modality by $\mathbf{X}^i \sim p_D^i(\mathbf{x}^i)$, and the collection of a generic subset of modalities by $\mathbf{X}^A \sim p_D^A(\mathbf{x}^A)$, with $\mathbf{X}^A \coloneqq \{\mathbf{X}^i\}_{i \in A}$, where $A \subset \{1, \dots, M\}$ is a set of indices; for example, given $A = \{1, 3, 5\}$, we have $\mathbf{X}^A = \{\mathbf{X}^1, \mathbf{X}^3, \mathbf{X}^5\}$.

We begin by considering unimodal VAEs as particular instances of the Markov chain $\mathbf{X} \to \mathbf{Z} \to \hat{\mathbf{X}}$, where $\mathbf{Z}$ is a latent random variable and $\hat{\mathbf{X}}$ is the generated random variable.

Models are specified by two conditional distributions. The encoder maps an input $\mathbf{x}$ to a latent representation $\mathbf{z}$, and is denoted by $q_\phi(\mathbf{z} \mid \mathbf{x})$, while the decoder generates (or reconstructs) data from the latent variable, and is denoted by $p_\theta(\hat{\mathbf{x}} \mid \mathbf{z})$.

For a given prior distribution $\rho(\mathbf{z})$, the objective is to define a generative model whose

samples are distributed as similarly as possible to the original data. In the case of multimodal VAEs, we consider the general family of Mixture of Product of Experts (MoPoE) (Sutter, Daunhawer, and Vogt, 2021), which includes as particular cases many existing variants such as Product of Experts (MVAE) (Wu and Goodman, 2018) and Mixture of Expert (MMVAE) (Shi et al., 2019).

Formally, let $S = \{A_1, \ldots, A_K\}$ be a collection of K arbitrary subsets of modalities. The posterior distribution is then defined as

$$q_\phi(\mathbf{z} \mid \mathbf{x}) = \sum_{i=1}^{K} \omega_i \, q_{\phi^{A_i}}(\mathbf{z} \mid \mathbf{x}^{A_i}), \tag{3.1}$$

where $\omega_i \geq 0$ and $\sum_{i=1}^{K} \omega_i = 1$ are weighting coefficients, and $\phi = \{\phi^1, \ldots, \phi^K\}$ represents the set of parameters. For notational simplicity, we use $q_{\phi^{A_i}}$ instead of $q^i_{\phi^{A_i}}$, while recognizing that each $q^i_{\phi^{A_i}}$ may have distinct parameters ϕ^{A_i} and functional forms.

For example, in the MoPoE (Sutter, Daunhawer, and Vogt, 2021) parametrization, we have:

$$q_{\phi^{A_i}}(\mathbf{z} \mid \mathbf{x}^{A_i}) = \prod_{j \in A_i} q_{\phi^j}(\mathbf{z} \mid \mathbf{x}^j). \tag{3.2}$$

Our exposition is more general and is not limited to this assumption. The selection of the posterior can be understood as the result induced by the two-step procedure where (i) each subset of modalities A_i is encoded into specific latent variables $\hat{\mathbf{Z}}_i \sim q_{\phi^{A_i}}(\cdot \mid \mathbf{x}^{A_i})$ and (ii) the latent variable $\mathbf{Z}$ is obtained as $\mathbf{Z} = \hat{\mathbf{Z}}_i$ with probability ω_i.

The optimization is performed with respect to the following ELBO (Daunhawer et al., 2022; Sutter, Daunhawer, and Vogt, 2021):

$$\mathcal{L} = \sum_i \omega_i \, \mathbb{E}_{p_D(\mathbf{x}) \, q_{\phi^{A_i}}(\mathbf{z} \mid \mathbf{x}^{A_i})} \left[\log p_\theta(\mathbf{x} \mid \mathbf{z}) \right] - \mathbb{E}_{p_D(\mathbf{x})} \mathbb{KL} \left(q_{\phi^{A_i}}(\mathbf{z} \mid \mathbf{x}^{A_i}) \parallel \rho(\mathbf{z}) \right). \tag{3.3}$$

3.3.2 Limitation and Trade-offs

A well-known limitation called the latent collapse problem (Alemi et al., 2018; Dieng et al., 2019) affects the quality of the latent variables $\mathbf{Z}$. Consider the hypothetical case of arbitrarily flexible encoders and decoders. Posteriors with zero mutual information with respect to the model inputs are valid maximizers of Eq. (3.3). To prove this, it is sufficient to substitute the posteriors $q_{\phi^{A_i}}(\mathbf{z} \mid \mathbf{x}^{A_i})$ by $\rho(\mathbf{z})$ and $p_\theta(\mathbf{x} \mid \mathbf{z})$ by $p_D(\mathbf{x})$ into Eq. (3.3) to observe that the optimal value of $\mathcal{L} = \mathbb{E}_{p_D(\mathbf{x})} \log p_D(\mathbf{x})$ is achieved (Alemi et al., 2018; Dieng et al., 2019).

The problem of *information loss* is exacerbated in the case of multimodal VAEs (Daun-

hawer et al., 2022). Intuitively, even if the encoders $q_{\phi^{A_i}}(\mathbf{z} \mid \mathbf{x}^{A_i})$ carry relevant information about their inputs $\mathbf{X}^{A_i}$, step (ii) of the multimodal encoding procedure described above induces a further information bottleneck. Some fraction ω_i of the time, the latent variable $\mathbf{Z}$ will be a copy of $\hat{\mathbf{Z}}_i$, which only provides information about the subset $\mathbf{X}^{A_i}$. No matter how good the encoding step is, the information about $\mathbf{X}^{\{1,\dots,M\}\setminus A}$ that is not contained in $\mathbf{X}^{A_i}$ cannot be retrieved.

The variable collapse problem can be analyzed through the lens of self-reconstruction, whereby a multimodal VAE is evaluated by simply reconstructing the same modality it receives as input. We have observed that these models tend to encode input samples into a latent space with potential information loss, leading to inconsistent reconstruction. This is particularly shown by the quantitative results in Table A.8.

Furthermore, if the latent variable carries zero mutual information with respect to the multimodal input, a coherent *conditional* generation of a set of modalities given others is impossible, as $\hat{\mathbf{X}}^{A_1} \perp \mathbf{X}^{A_2}$ for any generic sets A_1, A_2.

Let us consider the following factorization:

$$p_\theta(\mathbf{x} \mid \mathbf{z}) = \prod_{i=1}^{M} p_{\theta^i}(\mathbf{x}^i \mid \mathbf{z}), \tag{3.4}$$

with $\theta = \{\theta_1, \dots, \theta_M\}$ representing the set of parameters of the decoders. We use p_{θ^i} here instead of $p^i_{\theta^i}$ to unclutter the notation.

This factorization could enforce preservation of information and guarantee better quality of the *jointly* generated data; in practice, the latent collapse phenomenon induces multimodal VAEs to converge toward a suboptimal operating regime. When the posterior $q_\phi(\mathbf{z} \mid \mathbf{x})$ collapses onto the uninformative prior $\rho(\mathbf{z})$, the ELBO in Eq. (3.3) reduces to the sum of modality-independent reconstruction terms:

$$\sum_i \omega_i \sum_{j \in A_i} \mathbb{E}_{p_D^j(\mathbf{x}^j)\rho(\mathbf{z})} \left[\log p_{\theta^j}(\mathbf{x}^j \mid \mathbf{z}) \right], \tag{3.5}$$

where, paradoxically, the quality of the approximation of the various marginal distributions is extremely high, while there is a complete lack of joint coherence.

General principles to avoid latent collapse involve explicitly forcing the learning of informative encoders $q_\phi(\mathbf{z} \mid \mathbf{x})$ via β-annealing of the KL term in the ELBO and reducing the representational power of the encoders and decoders. While β-annealing (Asperti and Trentin, 2020) has been explored in the multimodal VAE literature (Wu and Goodman, 2018) with limited improvements reported, reducing the flexibility of the encoders/decoders clearly

impacts the generation quality. Hence, there is a tradeoff: in order to improve coherence, the flexibility of the encoders/decoders should be constrained, which in turn impacts generative quality. This tradeoff has recently been addressed in the literature on multimodal VAEs (Daunhawer et al., 2022; Palumbo, Daunhawer, and Vogt, 2023); however, our experimental results in § 3.5 indicate that there is ample room for improvement and that a new approach is truly needed.

3.4 Our Approach: Multimodal Latent Diffusion

We propose a new method for multimodal generative modeling that by design does not suffer from the limitations discussed in § 3.3. Our objective is to enable both high quality and coherent joint/conditional data generation using a simple design (see Figure 3.1 for a schematic representation). As an overview, we use deterministic unimodal autoencoders whereby each modality is encoded into a latent space. To enable such a simple design to become a generative model, we follow the two-stage procedure described in (Loaiza-Ganem et al., 2022; Tran et al., 2021; Dai and Wipf, 2019), where samples from the latent space are obtained through a score-based generative model. These models have shown tremendous performance in fitting complex distributions (Rombach et al., 2022; Vahdat, Kreis, and Kautz, 2021), an ability which aligns with our objective of learning the distribution within a multimodal latent space. Furthermore, the conditioning mechanism inherent in score-based models facilitates highly coherent generation. MLD is further enhanced by a multi-time diffusion process, a novel mechanism that allows for the generation of any subset of modalities, and which we explain in § 3.4.3.

It may be helpful at this point to clarify that the two-stage training of MLD is carried out separately. Unimodal deterministic autoencoders are pretrained first, followed by the training of the score-based diffusion model, which is explained in more detail later.

To conclude this overview of our method, for joint data generation it is possible to sample from noise, perform backward diffusion, and then decode the generated multimodal latent variable to obtain the corresponding data samples. For conditional data generation, given one modality, the reverse diffusion is guided by this modality, while the other modalities are generated by sampling from noise. The generated latent variable is then decoded to obtain data samples of the missing modality.

3.4.1 Modalities Encoding

We use deterministic unimodal autoencoders whereby each modality $\mathbf{X}^i$ is encoded through its encoder $\mathcal{E}_{\phi^i}$ (which is a short form for $\mathcal{E}^i_{\phi^i}$) into the modality-specific latent variable $\mathbf{Z}^i$ and decoded into the corresponding $\hat{\mathbf{X}}^i = \mathcal{D}_{\theta^i}(\mathbf{Z}^i)$. Our approach can be interpreted as a latent variable model in which the different latent variables $\mathbf{Z}^i$ are concatenated as $\mathbf{Z} = [\mathbf{Z}^1, \ldots, \mathbf{Z}^M]$.

Then, in place of an ELBO, we optimize the parameters of our autoencoders by minimizing the following sum of modality-specific losses:

$$\mathcal{L} = \sum_{i=1}^{M} \mathcal{L}_i, \quad \mathcal{L}_i = \mathbb{E}_{p^i_D(\mathbf{x}^i)} l^i(x^i, \mathcal{D}_{\theta^i}\left[\mathcal{E}_{\phi^i}(\mathbf{x}^i)\right]), \tag{3.6}$$

where l^i can be any valid distance function, e.g, the square norm $\|\cdot\|^2$. The parameters ϕ^i, θ^i are modality-specific; thus, minimization of Eq. (3.6) corresponds to individual training of the different autoencoders. The deterministic encoding ensures that there is no stochastic noise during the mapping from input $\mathbf{X}$ to latent variable $\mathbf{Z}$. Consequently, the mutual information $\mathcal{I}(\mathbf{X}, \mathbf{Z})$ becomes infinite in the continuous case, thereby preserving all information from $\mathbf{X}$ in $\mathbf{Z}$. Moreover, this choice avoids any form of interference in the back-propagated gradients corresponding to the unimodal reconstruction losses. Consequently, gradient conflict issues (Javaloy, Meghdadi, and Valera, 2022), in which stronger modalities pollute weaker ones, are avoided.

3.4.2 Multimodal Latent Diffusion Processes

In the first stage of our method, the deterministic encoders project the input modalities $\mathbf{X}^i$ into the corresponding latent spaces $\mathbf{Z}^i$. This transformation induces a distribution $p(\mathbf{z})$ for the latent variable $\mathbf{Z} = [\mathbf{Z}^1, \ldots, \mathbf{Z}^M]$, resulting from the concatenation of unimodal latent variables.

To generate a new sample for all modalities, we use a simple score-based diffusion model in latent space (Sohl-Dickstein et al., 2015; Song et al., 2021; Vahdat, Kreis, and Kautz, 2021; Loaiza-Ganem et al., 2022; Tran et al., 2021). This requires reversing a stochastic noising process, starting from a simple Gaussian distribution. Formally, these processes are defined using SDEs. In our multimodal setting, this translates as follows:

Forward SDE:

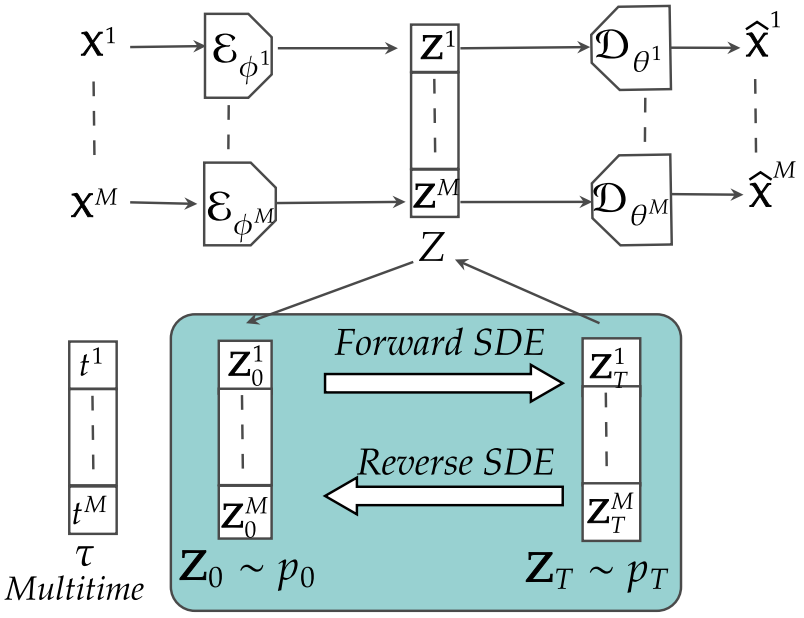


Figure 3.1: Multimodal Latent Diffusion: two-stage model involving (**Top**): deterministic modality-specific encoder/decoders and (**Bottom**): the score-based diffusion model on the latent spaces of the modalities, which evolve differently through the diffusion process according to a multi-time vector.

The noising process is defined by a SDE of the form.

$$\mathrm{d}\mathbf{Z}_t = f(\mathbf{Z}_t, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{W}_t, \tag{3.7}$$

where $f(\mathbf{Z}_t, t)$ and $g(t)$ are the drift and diffusion terms, respectively, and $\mathbf{W}_t$ is a Wiener process. The time-varying probability density $p_t(\mathbf{z})$ of the stochastic process at time $t \in [0, T]$, where T is finite, satisfies the Fokker–Planck equation (Oksendal, 2013). We consider $\mathbf{z}_0 \sim p(\mathbf{z})$ to be the initial condition for the diffusion process.

Reverse SDE: Under loose conditions (Anderson, 1982), a time-reversed stochastic process exists, with a new SDE of the form :

$$\mathrm{d}\mathbf{Z}_t = \left[f(\mathbf{Z}_t, t) - g^2(t)\nabla_{\mathbf{z}_t} \log p_t(\mathbf{z}_t)\right] \mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{W}}_t \tag{3.8}$$

where $\bar{\mathbf{W}}$ is a standard Wiener process when time flows backwards from T to 0. The simulation of Eq. (3.8) allows samples to be generated from the desired distribution $p_0(\mathbf{z})$. In practice, we use a **parametric score network** $s_\chi(\mathbf{z}_t, t)$ to approximate the true score function, and we approximate $p_T(\mathbf{z})$ with a stationary distribution $\rho(\mathbf{z})$ which is in general considered to be Gaussian. Learning the score function is possible via denoising score matching objective : (Song et al., 2021) (See § 2.2.6):

$$\mathcal{L}_{\text{DSM}}(\chi) = \frac{1}{2}\int_0^T \mathbb{E}_{\substack{\mathbf{z}_0\sim p(\mathbf{z}_0)\\ \mathbf{z}_t\sim p_{0t}(\mathbf{z}_t|\mathbf{z}_0)}} \left[\lambda(t)\left\|\nabla_{\mathbf{z}_t}\log p_{0t}(\mathbf{z}_t|\mathbf{z}_0) - s_\chi(\mathbf{z}_t,t)\right\|^2\right] \mathrm{d}t. \tag{3.9}$$

where $\lambda(t)$ is a positive weighting function and $p_{0t}(\mathbf{z}_t \mid \mathbf{z}_0)$ denotes the transition density of the forward diffusion process, i.e., the probability density of reaching state $\mathbf{z}_t$ at time t, given that the process started at state $\mathbf{z}_0$ at time 0. The learned score function s_χ allows to simulate the reverse process and approximate $p_0(\mathbf{z}) = p(\mathbf{z})$. Indeed, the generated data distribution $p_\chi(\mathbf{z})$ is close (in the KL sense) to the true density $p(\mathbf{z})$ as described by (Song et al., 2021; Franzese et al., 2023):

$$\mathbb{KL}\left(p(\mathbf{z}) \parallel p_\chi(\mathbf{z})\right) \leq \frac{1}{2}\int_0^T \mathbb{E}_{p_t(\mathbf{z})}\left[g^2(t)\|\nabla_{\mathbf{z}}\log p_t(\mathbf{z}) - s_\chi(\mathbf{z},t)\|^2\right] \mathrm{d}t \tag{3.10}$$

$$+ \mathbb{KL}\left(p_T(\mathbf{z}) \parallel \rho(\mathbf{z})\right). \tag{3.11}$$

The first term is referred to as the score-matching objective, while the second is a vanishing term for $T \to \infty$.

The joint generation of all modalities is achieved through simulation of the reverse-time SDE in Eq. (3.8) to generate a latent space $\mathbf{Z}_0$ followed by a simple decoding procedure using the modality specific decoders $\{\mathcal{D}_{\theta^i}(\mathbf{z}_0^i)\}_{i=0}^M$.

3.4.3 Masked Multi-time Diffusion

Given a partition of modalities into two non-overlapping sets A_1 and A_2, where $A_2 = \{1,\ldots,M\} \setminus A_1$, conditional generation involves sampling from the conditional distribution $p(\mathbf{z}^{A_1} \mid \mathbf{z}^{A_2})$, using masked forward and backward diffusion processes.

Given the conditioning latents $\mathbf{z}^{A_2}$, we define a modified forward diffusion process with initial conditions $\mathbf{Z}_0 = \mathcal{C}(\mathbf{Z}_0^{A_1}, \mathbf{z}^{A_2})$, where $\mathbf{Z}_0^{A_1} \sim p(\mathbf{z}^{A_1} \mid \mathbf{z}^{A_2})$. The composition operation $\mathcal{C}(\cdot)$ concatenates $\mathbf{Z}_0^{A_1}$ and $\mathbf{z}^{A_2}$.

Example: Consider $A_1 = \{1,3,5\}$ and $A_2 = \{2,4,6\}$. Then, $\mathbf{Z}_0$ is:

$$\mathbf{Z}_0 = \mathcal{C}(\mathbf{Z}_0^{A_1}, \mathbf{z}^{A_2}) = [\mathbf{Z}_0^1, \mathbf{z}^2, \mathbf{Z}_0^3, \mathbf{z}^4, \mathbf{Z}_0^5, \mathbf{z}^6].$$

Masked SDEs

More formally, we define the following masked forward-diffusion SDE:

$$\mathrm{d}\mathbf{Z}_t = m(A_1) \odot [f(\mathbf{Z}_t, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{W}_t] \tag{3.12}$$

The mask $m(A_1)$ contains M vectors u^i, one per modality, with the corresponding cardinality. If modality $j \in A_1$, then $u^j = 1$; otherwise, $u^j = 0$. Then, the effect of masking is to "freeze" the part of the random variable $\mathbf{Z}_t$ corresponding to the conditioning latent modalities $\mathbf{z}^{A_2}$ throughout the diffusion process. We naturally associate the conditional time-varying density $p_t(\mathbf{z}^{A_1} \mid \mathbf{z}^{A_2})$ with this modified forward process.

To sample from the desired conditional distribution $p_0(\mathbf{z}^{A_1} \mid \mathbf{z}^{A_2})$, we derive the reverse-time dynamics of Eq. (3.12) as follows:

$$\mathrm{d}\mathbf{Z}_t = m(A_1) \odot \left(\left[f(t)\mathbf{Z}_t - g^2(t)\nabla \log p_t(\mathbf{z}_t^{A_1} \mid \mathbf{z}^{A_2}) \right] \mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{W}}_t \right) \tag{3.13}$$

when time flows backwards from T to 0 with initial conditions $\mathbf{Z}_T = \mathcal{C}(\mathbf{Z}_T^{A_1}, \mathbf{z}^{A_2})$ and $\mathbf{Z}_T^{A_1} \sim p_T(\mathbf{z}_T^{A_1})$. which is approximated by its corresponding steady-state distribution $\rho(\mathbf{z}^{A_1})$ and the true (conditional) score function $\nabla \log p_t(\mathbf{z}_t^{A_1} \mid \mathbf{z}^{A_2})$ by a conditional score network $s_\chi(\mathbf{z}_t^{A_1}, t, \mathbf{z}^{A_2})$.

A correctly optimized score network $s_\chi(\mathbf{z}_t, t)$ allows sampling from the distribution $p_0(\mathbf{z})$ to be obtained through simulation of Eq. (3.8). Similarly, through the simulation of Eq. (3.13), a *conditional* score network $s_\chi(\mathbf{z}_t^{A_1}, t, \mathbf{z}^{A_2})$ allows for sampling from $p_0(\mathbf{z}^{A_1} \mid \mathbf{z}^{A_2})$. A naïve alternative is to rely on the unconditional score network $s_\chi(\mathbf{z}_t, t)$ for the conditional generation task by casting it as an *in-painting* objective.

Intuitively, any missing modality could be recovered in the same way that a unimodal diffusion model can recover masked information. The implicit assumptions underlying in-painting from an information-theoretic perspective are difficult to satisfy in the context of multimodal data. This intuition is corroborated by ample empirical evidence, where our method consistently outperforms alternatives Appendix A.1.3.

Next, we present a mechanism to allow conditional generation in an any-to-any scheme by learning all the conditional score networks wherby the conditioning can be applied.

Multi-Time Diffusion

We propose a modification to the classifier-free guidance technique (Ho and Salimans, 2022) (see § 2.2.6) to learn a score network that can generate conditional and unconditional samples from any subset of modalities. Instead of training a separate score network for each possible combination of conditional modalities, which is computationally infeasible, we use a single architecture that accepts all modalities as inputs and a *multi-time vector* $\tau = [t_1, \ldots, t_M]$. The multi-time vector serves two purposes: it is both a conditioning signal and the time at which we observe the diffusion process.

Training: Learning the conditional score network relies on randomization. As discussed in § 3.4.2, we consider an arbitrary partitioning of all modalities in two disjoint sets, A_1 and A_2; set A_2 contains randomly selected conditioning modalities, while the remaining modalities belong to set A_1. During training, the parametric score network estimates $\nabla \log p_t(\mathbf{z}_t^{A_1} \mid \mathbf{z}^{A_2})$, whereby the sets A_1, A_2 are randomly chosen at every step. This is achieved by the *masked diffusion process* from Eq. (3.12), which only diffuses modalities in A_1. More formally, the score network input is $\mathcal{C}(\mathbf{z}_t^{A_1}, \mathbf{z}^{A_2})$, along with a multi-time vector $\tau(A_1, t) = t \left[\mathbb{1}(1 \in A_1), \ldots, \mathbb{1}(M \in A_1)\right]$.

Example: As a follow-up of the example in § 3.4.2, given $A_1 = \{1, 3, 5\}$ such that $\mathbf{Z}^{A_1} = \{\mathbf{Z}^1, \mathbf{Z}^3, \mathbf{Z}^5\}$ and $A_2 = \{2, 4, 6\}$ such that $\mathbf{Z}^{A_2} = \{\mathbf{Z}^2, \mathbf{Z}^4, \mathbf{Z}^6\}$, we have $\tau(A_1, t) = [t, 0, t, 0, t, 0]$.

More precisely, the algorithm for multi-time diffusion training (see Appendix A.1 for the pseudo-code) proceeds as follows. At each step, a set of conditioning modalities A_2 is sampled from a predefined distribution ν, where $\nu(\emptyset) \stackrel{\text{def}}{=} \Pr(A_2 = \emptyset) = d$ and $\nu(U) \stackrel{\text{def}}{=} \Pr(A_2 = U) = {}^{(1-d)}/_{(2^M-1)}$ with $U \in \mathcal{P}(\{1, \ldots, M\}) \setminus \emptyset$, where $\mathcal{P}(\{1, \ldots, M\})$ is the powerset of all modalities. The corresponding set A_1 and mask $m(A_1)$ are constructed, and a sample $\mathbf{X}$ is drawn from the training dataset $P_{data}(\mathbf{x})$.

The corresponding latent variables $\mathbf{Z} = \{\mathcal{E}_\phi^i(\mathbf{X}^i)\}_{i=0}^M$ are obtained using the pretrained encoders and a diffusion process starting from $\mathbf{Z}_0 = \mathbf{Z}$ is simulated for a randomly chosen diffusion time t using the conditional forward SDE with the mask $m(A_1)$. The score network is then fed the current state $\mathbf{z}_t$ and multi-time vector $\tau(A_1, t)$ and the difference between the score network's prediction and the true score is computed while applying mask $m(A_1)$. The score network parameters are updated using stochastic gradient descent, and this process is repeated for until convergence. Clearly, when $A_2 = \emptyset$, training proceeds the same as for an unmasked diffusion process, as mask $m(A_1)$ allows all of the latent variables to be diffused.

Conditional generation: Inference time conditional generation is not randomized; the conditioning modalities are the ones that are available, whereas those remaining are the ones we wish to generate. Any valid numerical integration scheme for Eq. (3.13) can be used for conditional sampling (see Appendix A.1 for an implementation using the Euler–Maruyama integrator). First, conditioning modalities in set A_2 are encoded into the corresponding latent variables $\mathbf{z}^{A_2} = \{\mathcal{E}^j(\mathbf{x}^j)\}_{j\in A_2}$. Then, numerical integration is performed with a step size of $\Delta t = {}^{T}/_{N}$, starting from initial conditions $\mathbf{Z}_T = \mathcal{C}(\mathbf{Z}_T^{A_1}, \mathbf{z}^{A_2})$ with $\mathbf{Z}_T^{A_1} \sim \rho(\mathbf{z}^{A_1})$. At each integration step, the score network s_χ is fed the current state of the process and the multi-time vector $\tau(A_1, \cdot)$. Before updating the state, the masking is applied. Finally, the generated modalities are obtained thanks to the decoders as $\hat{\mathbf{X}}^{A_1} = \{\mathcal{D}_\theta^j(\mathbf{Z}_0^j)\}_{j\in A_1}$.

3.4.4 Understanding Modality Interactions in MLD

MLD treats the latent spaces of each modality as variables that evolve differently through the diffusion process according to a multi-time vector. The masked multi-time training enables the model to learn the score of all the combinations of conditionally diffused modalities, using the frozen modalities as the conditioning signal through a randomized scheme. By learning the score function of the diffused modalities at different time steps, the score model captures the correlation between the modalities.

At test time, the diffusion time of each modality is chosen so as to modulate its influence on the generation. For joint generation, the model uses the unconditional score, which corresponds to using the same diffusion time for all modalities. Thus, all the modalities influence each other equally. This ensures that the modality interaction information is faithful to the information characterizing the observed data distribution. The model can also generate modalities conditionally using the conditional score by freezing the conditioning modalities during the reverse process. The frozen state is similar to the final state of the revere process, where information is not perturbed; thus, the influence of the conditioning modalities is maximal. Subsequently, the generated modalities reflect the necessary information from the conditioning modalities and achieve the desired correlation.

3.5 Experimental Validation

We compared our MLD method to MVAE (Wu and Goodman, 2018), MMVAE (Shi et al., 2019), MoPoE (Sutter, Daunhawer, and Vogt, 2021), NEXUS (Vasco et al., 2022), MVTCAE (Hwang et al., 2021), and MMVAE+ (Palumbo, Daunhawer, and Vogt, 2023), re-implementing all competitors in the same code base as our method and selecting their best hyperparameters

as indicated by the authors (see Appendix A.4 for more details). For a fair comparison, we used the same encoder/decoder architecture for all models. For MLD, the score network was implemented using a simple stacked Multilayer perceptron (MLP) with skip connections (see Appendix A.1 for more details). MLD was also contrasted with multimodal diffusion-based approaches: (Bao et al., 2023) in Appendix A.2 and (Wesego and Rooshenas, 2023) in § 3.5.5.

Evaluation metrics:

- *Coherence* was measured as in (Shi et al., 2019; Sutter, Daunhawer, and Vogt, 2021; Palumbo, Daunhawer, and Vogt, 2023), using pretrained classifiers on the generated data and checking the consistency of their outputs.
- *Generative quality* was computed using the Fréchet Inception Distance (FID) (Heusel et al., 2017) and Fréchet Audio Distance (FAD) (Kilgour et al., 2019) scores for images and audio, respectively.

Full details on the metrics are included in Appendix A.3. All results were averaged over five seeds. We report the standard deviations in Appendix A.5.

Overall Results: Overall, MLD largely outperformed the alternatives from the literature in terms of both coherence and generative quality. The VAE-based models suffered from the coherence–quality tradeoff as well as from modality collapse for highly heterogeneous datasets. We proceed to show this on several standard benchmarks from the multimodal VAE-based literature; see Appendix A.3 for details on the datasets.

3.5.1 MNIST-SVHN

The first dataset we consider is **MNIST-SVHN** (Shi et al., 2019), where the two modalities differ in complexity. High variability, noise, and ambiguity make attaining good coherence for the SVHN modality a challenging task.

Overall, MLD outperforms all VAE-based alternatives in terms of coherency, especially in terms of joint generation and conditional generation of MNIST given SVHN (see Table 3.1). The mixture models, MMVAE and MoPoE, suffer from modality collapse (poor SVHN generation), whereas the product-of-experts models MVAE and MVTCAE generate better-quality samples at the expense of SVHN to MNIST conditional coherence. Joint generation is poor for all VAE models. Interestingly, these models also fail at SVHN self-reconstruction, which we discuss in Appendix A.5. MLD also achieves the best performance in terms of generation quality, as confirmed by qualitative results (Figure 3.2) showing, for example, how MLD

conditionally generates multiple SVHN digits within one sample given the input MNIST image, whereas the other methods fail to do so.

Table 3.1: Generation coherence and quality for **MNIST-SVHN** (M: MNIST, S: SVHN). The generation quality is measured in terms of the Fréchet Modality Distance (FMD) for MNIST and FID for SVHN. We report both joint and conditional generation performance results. Bold and underlined numbers indicate the best and second best scores respectively.

Models	Coherence (%↑)			Quality (↓)			
	Joint	M → S	S → M	Joint (M)	Joint(S)	M → S	S → M
MVAE	38.19	48.21	28.57	13.34	68.9	<u>68.0</u>	13.66
MMVAE	37.82	11.72	67.55	25.89	146.82	393.33	53.37
MoPoE	39.93	12.27	68.82	20.11	129.2	373.73	43.34
NEXUS	40.0	16.68	<u>70.67</u>	13.84	98.13	281.28	53.41
MVTCAE	<u>48.78</u>	<u>81.97</u>	49.78	<u>12.98</u>	**52.92**	69.48	<u>13.55</u>
MMVAE+	17.64	13.23	29.69	26.60	121.77	240.90	35.11
MMVAE+ (K = 10)	41.59	55.3	56.41	19.05	67.13	75.9	18.16
MLD (ours)	**85.22**	**83.79**	**79.13**	**3.93**	<u>56.36</u>	**57.2**	**3.67**

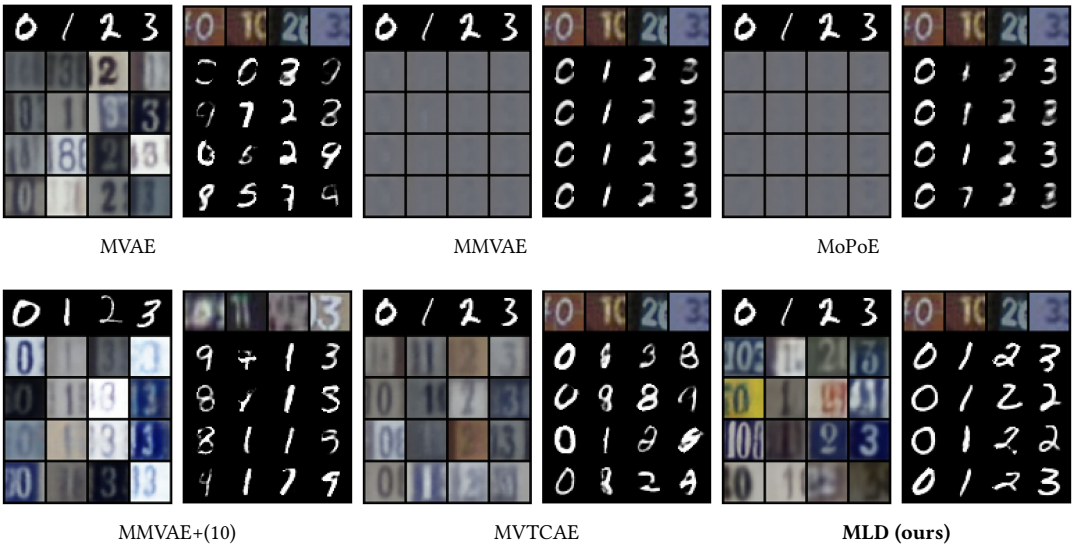


Figure 3.2: Qualitative results for **MNIST-SVHN**. For each model, we report MNIST to SVHN conditional generation on the left and SVHN to MNIST conditional generation on the right. The conditioning modality is illustrated by the first row, with the generated samples below.

3.5.2 MHD

The Multimodal Handwritten Digits dataset (**MHD**) (Vasco et al., 2022) contains gray-scale images of digits, the motion trajectory of the handwriting, and the sounds of the spoken digits. In our experiments, we did not use the label as a fourth modality. While the images and trajectories share a good amount of information, the sound modality contains a great deal more modality-specific variation. Consequently, both conditional generation involving

the sound modality and joint generation represent challenging tasks. Coherency-wise, (Table 3.2) MLD outperforms all the competitors, with the biggest difference seen in joint generation and generation from sound to other modalities. On the latter task, MVTCAE performs better than other competitors, but is still worse than MLD. MLD dominates the alternatives in terms of generation quality (Table 3.3). This is true both for image and sound modalities, for which some VAE-based models struggle to produce high-quality results, demonstrating the limitation of these methods in handling highly heterogeneous modalities. MLD, on the other hand, achieves high generation quality for all modalities, possibly due to the independent training of the autoencoders avoiding interference.

Table 3.2: Generation coherence (%) for **MHD** (higher is better). Line above refers to the generated modality, while the subset of observed modalities is presented below. Bold and underlined numbers indicate the best and second best scores respectively.

Models	Joint	I (Image)			T (Trajectory)			S (Sound)		
		T	S	T,S	I	S	I,S	I	T	I,T
MVAE	37.77	11.68	26.46	28.4	95.55	26.66	96.58	58.87	10.76	58.16
MMVAE	34.78	**99.7**	69.69	84.74	<u>99.3</u>	85.46	92.39	49.95	50.14	50.17
MoPoE	48.84	<u>99.64</u>	68.67	<u>99.69</u>	99.28	<u>87.42</u>	99.35	50.73	51.5	56.97
NEXUS	26.56	94.58	<u>83.1</u>	95.27	88.51	76.82	93.27	70.06	75.84	89.48
MVTCAE	42.28	99.54	72.05	99.63	99.22	72.03	<u>99.39</u>	<u>92.58</u>	<u>93.07</u>	<u>94.78</u>
MMVAE+	41.67	98.05	84.16	91.88	97.47	81.16	89.31	64.34	65.42	64.88
MMVAE+ (K = 10)	42.60	99.44	**89.75**	94.7	99.44	**89.58**	95.01	87.15	87.99	87.57
MLD (ours)	**98.34**	99.45	<u>88.91</u>	**99.88**	**99.58**	<u>88.92</u>	**99.91**	**97.63**	**97.7**	**98.01**

Table 3.3: Generation quality for **MHD** in terms of FMD for image and trajectory modalities and FAD for the sound modality (lower is better). Bold and underlined numbers indicate the best and second best scores respectively.

Models	I (Image)				T (Trajectory)				S (Sound)			
	Joint	T	S	T,S	Joint	I	S	I,S	Joint	I	T	I,T
MVAE	<u>94.9</u>	93.73	92.55	91.08	39.51	20.42	38.77	19.25	14.14	<u>14.13</u>	14.08	14.17
MMVAE	224.01	22.6	789.12	170.41	16.52	**0.5**	30.39	6.07	22.8	22.61	23.72	23.01
MoPoE	147.81	16.29	838.38	15.89	<u>13.92</u>	<u>0.52</u>	33.38	**0.53**	18.53	24.11	24.1	23.93
NEXUS	281.76	116.65	282.34	117.24	18.59	6.67	33.01	7.54	<u>13.99</u>	19.52	18.71	16.3
MVTCAE	121.85	<u>5.34</u>	<u>54.57</u>	<u>3.16</u>	19.49	0.62	<u>13.65</u>	0.75	15.88	14.22	<u>14.02</u>	<u>13.96</u>
MMVAE+	97.19	2.80	128.56	114.3	22.37	1.21	21.74	15.2	16.12	17.31	17.92	17.56
MMVAE+ (K = 10)	85.98	1.83	70.72	62.43	21.10	1.38	8.52	7.22	14.58	14.33	14.34	14.32
MLD	**7.98**	**1.7**	**4.54**	**1.84**	**3.18**	0.83	**2.07**	<u>0.6</u>	**2.39**	**2.31**	**2.33**	**2.29**

3.5.3 POLYMNIST

The **POLYMNIST** dataset (Sutter, Daunhawer, and Vogt, 2021) consists of five modalities synthetically generated using MNIST digits and varying the background images. The homo-

geneous nature of the modalities is expected to mitigate gradient conflict issues in VAE-based models and consequently reduce modality collapse. However, MLD still outperforms all alternatives, as shown in Figure 3.3 and Figure 3.4 . Concerning generation coherence, MLD achieves the best performance in all cases, with the one exception of a single observed modality. On the qualitative performance side, not only is MLD superior to all alternatives, its results are stable when more modalities are considered, a capability that not all competitors share.

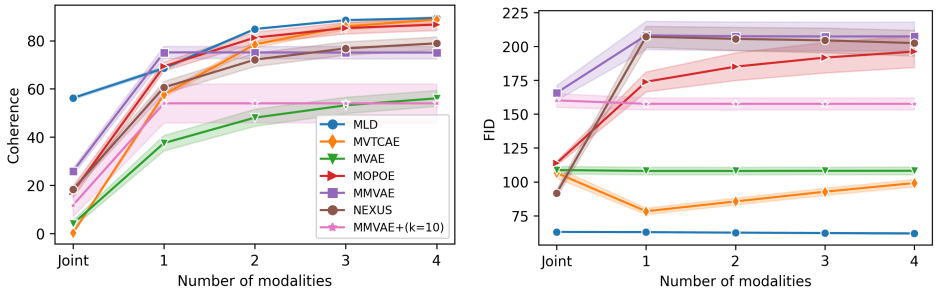


Figure 3.3: Performance results for **POLYMNIST** as a function of the number of inputs. (**Right**): Generative coherence (% ↑). (**Left**): Generative quality in terms of FID (↓). We report the average performance following the leave-one-out strategy (see Appendix A.3).

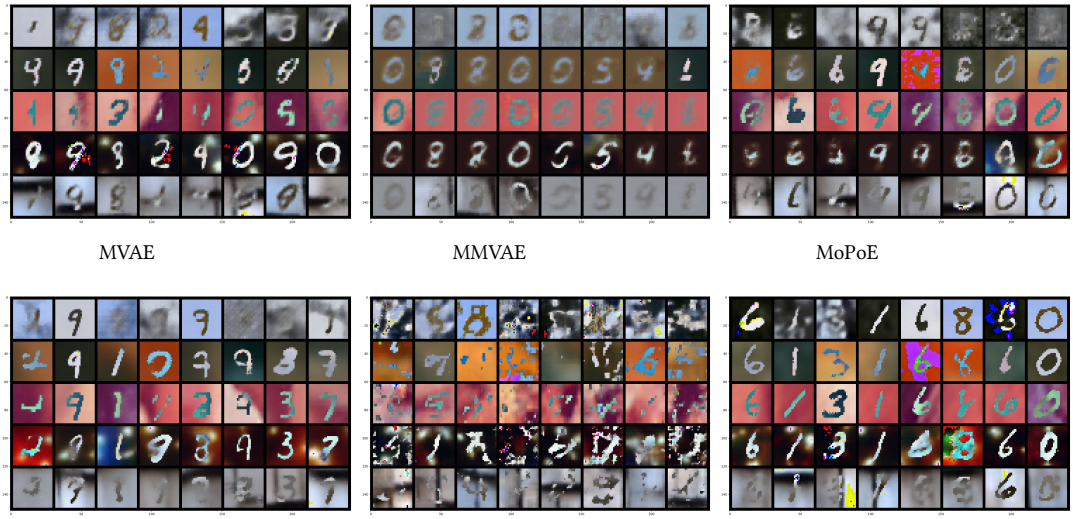


Figure 3.4: Joint generation qualitative results for **POLYMNIST** across the five modalities.

3.5.4 CUB

Next, we explored the Caltech Birds **CUB** (Shi et al., 2019) dataset, following the experimental protocol in (Daunhawer et al., 2022) using real bird images instead of ResNet-features as in (Shi et al., 2019). Figure 3.5 presents qualitative results for caption-to-image conditional

generation. MLD is the only model capable of generating bird images with convincing coherence. Clearly, none of the VAE-based methods is able to achieve sufficient caption-to-image conditional generation quality using the same simple autoencoder architecture. Note that an image autoencoder with larger capacity considerably improves the generative performance of MLD, suggesting that careful engineering applied to modality-specific autoencoders is a promising avenue for future work. We report quantitative results in Appendix A.5, where we show the generation quality FID metric. Due to the unavailability of the labels in this dataset, the coherence evaluation performed with the previous datasets was not possible. Thus, we resorted to CLIP-Score (Clip-s) (Hessel et al., 2021), an image-captioning metric. Despite its limitations for the considered dataset (Kim et al., 2022), Clip-s shows that MLD outperforms all competitors.

Figure 3.5: Qualitative results on the **CUB** dataset, with the caption used as the condition to generate the bird images. **MLD*** denotes the version of our method using a powerful image autoencoder.

3.5.5 CelebAMask-HQ

We also consider the CelebAMask-HQ dataset (Lee et al., 2019), which consists of three modalities: face images, each having a segmentation mask and text attributes. We followed the same experimental protocol as in (Wesego and Rooshenas, 2023), including the autoencoder base architecture.

The image generation quality was evaluated in terms of FID score. The attributes and the mask, both having binary values, were evaluated against the ground truth in terms of the $F1$ score. The competitors performance results are reported from (Wesego and Rooshenas, 2023).

The quantitative results in Table 3.4 show that MLD outperforms the competitors in terms of generation quality. Our method achieves the best $F1$ score in generation of the attribute modalities given the image and mask modalities. In mask generation, MoPoE and MVTCAE achieve the best performance, with MLD achieving the second-best performance in mask generation conditioning on both the image and attribute modalities.

Overall, MLD stands out with the best image quality generation, while being on par with the competition in terms of mask and attribute generation coherence. Figure 3.6 shows the qualitative results for MLD on the joint generation task. It can be observed that our method succeeds at generating all three modalities with high coherence and quality. The same observation is valid for the conditional generation tasks (see Figures 3.7,A.22,A.23, A.24).

Table 3.4: Quantitative results on the CelebAMask-HQ dataset. Performance is measured in terms of the FID (↓) and F1 score (↑). The first row shows the generated modality, while the second row shows the modalities used as conditions. Supervised classifier designates a classifier performance to predict the attributes or the mask from an image. Bold numbers indicate the best scores.

Models	Attributes		Image				Mask	
	Img + Mask F1	Img F1	Att + Mask FID	Mask FID	Att FID	Joint FID	Img + Att F1	Img F1
SBM-RAE	0.62	0.6	84.9	86.4	85.6	84.2	0.83	0.82
SBM-RAE-C	0.66	0.64	83.6	82.8	83.1	84.2	0.83	0.82
SBM-VAE	0.62	0.58	81.6	81.9	78.7	79.1	0.83	0.83
SBM-VAE-C	0.69	0.66	82.4	81.7	76.3	79.1	0.84	0.84
MoPoE	0.68	**0.71**	114.9	101.1	186.8	164.8	0.85	**0.92**
MVTCAE	0.71	0.69	94	84.2	87.2	162.2	**0.89**	0.89
MMVAE+	0.64	0.61	133	97.3	153	103.7	0.82	0.89
Supervised classifier		0.79					0.94	
MLD (ours)	**0.72**	0.69	**52.75**	**51.73**	**53.09**	**54.27**	0.87	0.87

(a) Image **(b)** Mask **(c)** Attributes

Figure 3.6: Joint (unconditional) generation: qualitative results of **MLD** on CelebAMask-HQ.

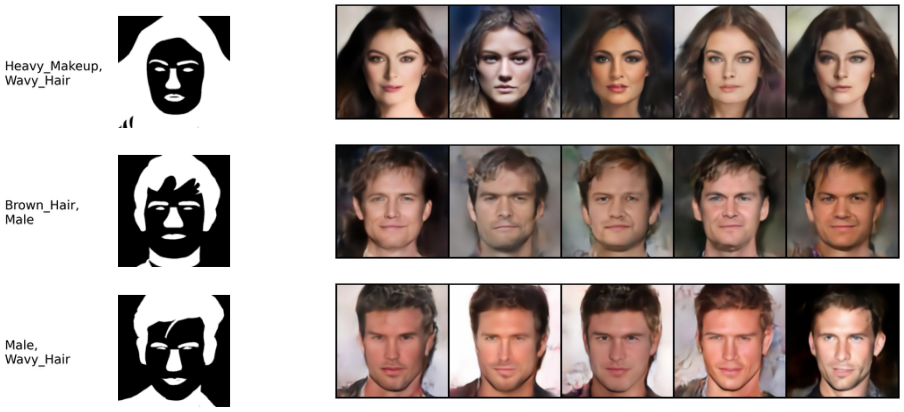


Figure 3.7: (Attributes, Mask → Image). Conditional generation of **MLD** on CelebAMask-HQ. The two columns on the left present the conditioning modalities, while several conditionally generated samples are displayed on the right.

3.6 Use-Case : Enhancing Sensor Robustness in Automotive Systems

Modern automotive systems increasingly rely on a wide array of sensors to deliver safe and efficient operation. However, sensor data quality can be compromised due to environmental conditions, sensor malfunctions, or occlusions, which can jeopardize vehicle performance. In this section we proposes MLD to enhance sensor robustness. MLD can be used for modality cross-generation and enhancement, enabling the reconstruction of impaired sensor data through information from other modalities. We demonstrate the performance of MLD in the automotive context, focusing on improving night vision capabilities using data from Light Detection And Ranging (LiDar) and Radio Detection And Ranging (RaDar) sensors to generate enhanced camera images. This approach ensures robust sensor functionality, offering a general solution for sensor data integrity in automotive systems.

3.6.1 Multimodality in Automotive

Advanced driver-assistance systems (ADAS) rely on sophisticated sensor integration. Cameras, RaDar and LiDar are essential components, providing comprehensive data for navigation, obstacle detection, and decision-making processes. However, the reliability of these systems is often compromised by issues such as sensor degradation, environmental factors, and data loss, which necessitate robust solutions for ensuring data integrity.

Multimodal generative models can be used to address these challenges. By leveraging data from multiple sensor modalities, we aim to reconstruct missing or impaired sensor data,

thereby enhancing the overall robustness of automotive systems. Our focus is on modality cross-generation and enhancement, where information from one modality can be used to generate or improve another. We explore a specific application in enhancing night vision capabilities, using RaDar and LiDar data to augment camera imagery in low-light conditions.

The integration of multimodal sensors in the automotive industry is pivotal to the advancement of autonomous driving systems. By combining data from multiple sensor types, such as cameras, LiDar, RaDar, and vehicles can achieve a comprehensive understanding of their surroundings, which is critical for safe and reliable navigation in diverse driving conditions. This multimodal approach leverages the strengths of each sensor type: cameras provide high-resolution visual data, LiDar offers precise depth information, RaDar excels in detecting objects in adverse weather conditions, and Global Positioning System (GPS) ensures accurate localization.

The use of generative models with multimodal sensor data has demonstrated great potential in improving autonomous vehicle performance. For example, (Abu Tami et al., 2024) employed multimodal large models for hazard detection, (Ivanovic et al., 2020) utilized them for trajectory prediction and (Li et al., 2024) to enhance perception. Works like (Da Silva–Filarder et al., 2021) (Roy et al., 2023) (Huang et al., 2020) focused on sensor fusion to enhance vehicle perception. (Bogdoll, Yang, and Zöllner, 2023) and (Hu et al., 2023) introduced multimodal world models to improve prediction quality in autonomous driving. (Zheng et al., 2024) proposed an end-to-end autonomous driving paradigm based on generative models.

3.6.2 Modality enhancement with MLD

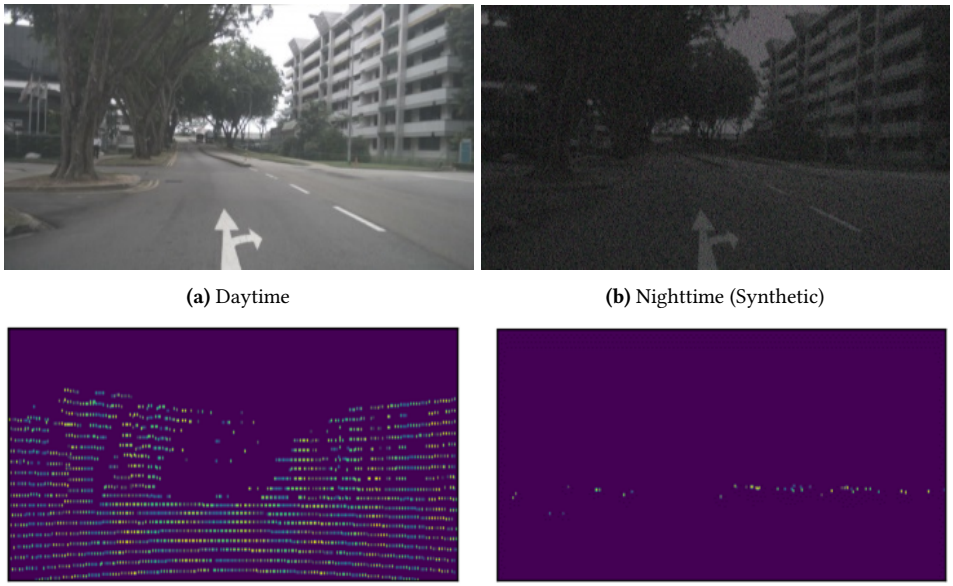
Modality enhancement extends the concept of cross generation by starting the reverse diffusion process from an intermediate latent state $\mathbf{z}_{\tilde{t}}$ rather than the terminal noise state $\mathbf{z}_T$ with $\tilde{t} \in [0, T]$. This approach is particularly useful when aiming to refine or augment the quality of an existing modality while retaining the influence of the initial data sample. In this setting, the masked forward diffusion SDE Eq. (3.13) remains the same. Consequently, the same model trained for cross generation purposes can be reused in zero shot manner.

The choice of $\tilde{t}$ determines the extent of enhancement: starting from a smaller $\tilde{t}$ (closer to 0) retains more of the original data characteristics, while a higher value $\tilde{t}$ (closer to T) allows for more significant generative modifications. Thus, modality enhancement provides a flexible mechanism for balancing between preservation of original modality details and the introduction of new features via the generative process.

3.6.3 Improved Night Vision

Perception systems in automotive applications rely heavily on camera data to provide visibility. However, in low-light conditions, cameras often struggle to capture high-quality images impacting object detection and navigation. Our approach relies on MLD to integrate the information from LiDar and RaDar sensors to reconstruct or enhance night vision capabilities. By generating high-quality camera images using radar and LiDar information, we improve visibility and object detection accuracy in low-light environments.

A crucial challenge in night to day application is the absence of paired night -day dataset. To solve this problem, during training we synthetically generate an additional modality which consists of a low illumination version of the daytime sample (See Figure 3.8 for an overview).



(a) Daytime **(b)** Nighttime (Synthetic)

(c) LiDar **(d)** RaDar

Figure 3.8: The different modalities used for training in the improved night vision application

As our goal is to focus on camera generation, we adjust the randomized set during training to target only the camera modality. Suppose that $\mathbf{X}^1$, $\mathbf{X}^2$, $\mathbf{X}^3$, and $\mathbf{X}^4$ represent the daytime camera, nighttime camera, RaDar, and LiDar, respectively. Then, during training, the target (generated) modality is defined as $A_1 = \{1\}$, and the conditioning modalities are given by the power set of the remaining modalities: $A_2 = \mathcal{P}(\{2, 3, 4\})$.

Dataset

We use nuScenes(Caesar et al., 2020) dataset which is a comprehensive autonomous driving dataset that includes sensor data from cameras, LiDar, RaDar, and more, collected from urban environments in various weather and lighting conditions.

In our approach, we train models using the front camera frames sampled with $12Hz$ of 512×288 resolution. RaDar, and LiDar data which are projected onto the same intrinsic parameters as the front camera, ensuring a unified perspective across modalities. We exclusively use the daytime data from the nuScenesdataset for training. In Table A.7, we provide the dataset size information after preprocessing. To simulate nighttime conditions, we apply a low-illumination algorithm, like (Li et al., 2024).

Implementation

Our approach employs a Diffusion Transformer (DiT) architecture (Peebles and Xie, 2023) , leveraging the power of transformer networks to handle multiple modalities and capture long-range dependencies. The DiT architecture is designed to process multiple modalities simultaneously, utilizing token-based representations. The diffusion model is initialized with weights pre-trained on the ImageNet dataset (Deng et al., 2009) and the autoencoder part, leverage (Rombach et al., 2022) to encode the different modalities. For more details please refer to Appendix A.4.4.

Results

During testing, we reverse this process by reconstructing daytime modalities from the simulated nighttime data. We evaluate the quality of this reconstruction using metrics like Peak Signal-to-Noise Ratio (PSNR) (Gonzalez, 2009), Structural Similarity Index Measure (SSIM) (Wang et al., 2004), and Learned Perceptual Image Patch Similarity (LPIPS) (Zhang et al., 2018).

Table 3.5 presents the performance results of MLD improvement across different modes. In our experiments, we reconstructed the daytime camera modality using nighttime camera data, testing various scenarios that incorporate additional information from radar and LiDar, LiDar only, RaDar only, and no additional sensors. We observe that LiDar contributes the most to improving camera reconstruction, though radar also yields comparable results. The performance is lowest when no additional sensors are used, indicating the importance of supplementary sensor data in enhancing reconstruction quality.

Figure 3.9 illustrates MLD performance on real nighttime images, showcasing clearer and more detailed nighttime images in challenging illumination conditions.

MLD	PSNR (↑)	SSIM (↑)	LPIPS (↓)
All sensors	**26.965**	**0.705**	**0.201**
LiDar	26.751	0.698	0.207
RaDar	26.589	0.694	0.211
Only camera	26.565	0.680	0.221

Table 3.5: Performance results on Daytime camera reconstruction using nighttime camera image with the help of additional sensors, (LiDar, RaDar).

(a) Nighttime Camera **(b)** Improved Nighttime camera

Figure 3.9: Quantitative results of MLD conditional generation using real nighttime modality and the different sensors to improve night perception. In this experiment, we consider an intermediate diffusion time $\tilde{t} = 0.5$

3.7 Conclusion

We have presented a new multimodal generative model, Multimodal Latent Diffusion (MLD), to address the well known coherence–quality tradeoff that is inherent in existing multimodal VAE-based models. MLD uses a set of independently trained unimodal deterministic autoencoders. The generative properties of our model stem from a masked diffusion process that operates on latent variables. In addition, we have developed a new multi-time training method to learn the conditional score network for multimodal diffusion. An extensive experimental campaign on various real-life datasets provides compelling evidence of the effectiveness of MLD for multimodal generative modeling. In all scenarios, including cases

with loosely correlated modalities and high-resolution datasets, MLD consistently outperforms state-of-the-art alternatives. Lastly, we demonstrate the potential of our method in addressing real-world challenges in automotive applications. MLD leverages the strengths of different sensor modalities which allows the generation and enhancement of sensor data, improving the overall reliability and performance of automotive systems.

Chapter 4

Mutual Information Estimation

In this work we present a new method for the estimation of Mutual Information (MI) between random variables. Our approach is based on an original interpretation of the the Fokker–Planck equations, which allows us to use score-based diffusion models to estimate the KL divergence between two densities as a difference between their score functions. As a by-product, our method also enables the estimation of the entropy of random variables. Armed with such building blocks, we present a general recipe to measure MI, which unfolds in two directions: one uses conditional diffusion process, whereas the other uses joint diffusion processes that allow simultaneous modeling of two random variables. Our results, which derive from a thorough experimental protocol over all the variants of our approach, indicate that our method is more accurate than the main alternatives from the literature, especially for challenging distributions. Furthermore, our methods pass MI self-consistency tests, including data processing and additivity under independence, which instead are a pain-point of existing methods. Finally, we show how to exploit pre-trained, text-to-image models to compute MI between input modalities, which is instrumental for the analysis of the generative properties of such models.

4.1 Introduction

Mutual Information (MI) is a central measure to study the non-linear dependence between random variables (Shannon, 1948; MacKay, 2003), and has been extensively used in machine learning for representation learning (Bell and Sejnowski, 1995; Stratos, 2019; Belghazi et al., 2018; Oord, Li, and Vinyals, 2018; Hjelm et al., 2019), and for both training (Alemi et al., 2016; Chen et al., 2016; Zhao, Song, and Ermon, 2018) and evaluating generative models (Alemi and Fischer, 2018; Huang et al., 2020).

For many problems of interest, precise computation of MI is not an easy task (McAllester and Stratos, 2020; Paninski, 2003), and a wide range of techniques for MI estimation have flourished. As the application of existing parametric and non-parametric methods (Pizer et al., 1987; Moon, Rajagopalan, and Lall, 1995; Kraskov, Stögbauer, and Grassberger, 2004; Gao, Ver Steeg, and Galstyan, 2015) to realistic, high-dimensional data is extremely challenging, if not unfeasible, recent research has focused on variational approaches (Barber and Agakov, 2004; Nguyen, Wainwright, and Jordan, 2007; Nowozin, Cseke, and Tomioka, 2016; Poole et al., 2019; Wunder et al., 2021; Letizia, Novello, and Tonello, 2023; Federici, Ruhe, and Forré, 2023) and neural estimators (Papamakarios, Pavlakou, and Murray, 2017; Belghazi et al., 2018; Oord, Li, and Vinyals, 2018; Song and Ermon, 2019; Rhodes, Xu, and Gutmann, 2020; Letizia and Tonello, 2022; Brekelmans et al., 2022) for MI estimation. In particular, the work by Song and Ermon (2019a) and Federici, Ruhe, and Forré (2023) classify recent MI estimation methods into discriminative and generative approaches. The first class directly learns to estimate the ratio between joint and marginal densities, whereas the second estimates and approximates them separately.

In this work, we address the problem of estimating MI through generative methods, introducing a novel perspective. In § 4.2, we describe how score functions can be utilized to estimate the KL divergence between two probability distributions. While this approach was previously demonstrated via Girsanov's theorem (Øksendal, 2003) in (Franzese, Bounoua, and Michiardi, 2024), we provide an alternative and simplified derivation grounded in the Fokker–Planck formalism (Risken, 1996).In § 4.3, we investigate the theoretical guarantees and estimation error associated with our proposed method. As further shown in § 4.4, the resulting KL divergence estimator also facilitates the estimation of the entropy of a continuous random variable. In § 4.5 we present our general recipe for computing the MI between two arbitrary distributions, which we develop according to two modeling approaches, i.e., conditional and joint diffusion processes. The conditional approach is simple and capitalizes on standard diffusion models, but it is inherently more rigid, as it requires one distribution to be selected as the conditioning signal. Joint diffusion processes, on the other hand, are more flexible, but require an extension of traditional diffusion models, which deal with dynamics that allow data distributions to evolve according to multiple arrows of time.

Recent work by Czyż et al. (2023) argue that MI estimators are mostly evaluated assuming simple, multivariate normal distributions for which MI is analytically tractable, and propose a novel benchmark that introduces several challenges for estimators, such as sparsity of interactions, long-tailed distributions, invariance, and high mutual information. Furthermore, Song and Ermon (2019a) introduce measures of self-consistency (additivity under

independence and the data processing inequality) for MI estimators, to discern the properties of various approaches. In § 4.6 we evaluate several variants of our method, which we call Mutual Information Neural Diffusion Estimation (MINDE), according to such challenging benchmarks: our results show that MINDE outperforms the competitors on a majority of tasks, especially those involving challenging data distributions. Moreover, MINDE passes all self-consistency tests, a property that has remained elusive so far, for existing neural MI estimators. Finally, we demonstrate that leveraging MINDE in conjunction with pre-trained text-to-image models to compute MI between input modalities is crucial for analyzing the generative properties of these models.

4.2 Score-based KL estimation

Consider a generic random variable $\mathbf{X}$ with associated distribution $p(\mathbf{x})$. SDEs (Song and Ermon, 2019; Song et al., 2021) have emerged as an essential tool for introducing controlled noise into data distributions. This controlled noising process is critical for techniques such as score-based generative modeling, where understanding the evolution of the data distribution under stochastic dynamics is key to generating new samples. The noisy process can be modeled in terms of the following SDE:

$$\mathrm{d}\mathbf{X}_t = f(t)\mathbf{X}_t\mathrm{d}t + g(t)\mathrm{d}\mathbf{W}_t, \tag{4.1}$$

where $f(t)\mathbf{X}_t$ and $g(t)$ are the drift and diffusion terms, respectively, and $\mathbf{W}_t$ is a Wiener process. We consider $\mathbf{X}_0 \sim p(\mathbf{x})$ to be the initial condition for the diffusion process. The time-varying probability density $p_t(\mathbf{x})$ of the stochastic process at time $t \in [0, T]$, where T is finite, satisfies the Fokker–Planck equation (Oksendal, 2013), which we express in the following compact form (see Appendix B.1.1 for the detailed derivation).

$$\frac{\partial p_t}{\partial t} = -\nabla \cdot (f(t)\mathbf{x}p_t) + \frac{1}{2}g^2(t)\Delta p_t. \tag{4.2}$$

Where:

- $\nabla p_t = \frac{\partial p_t}{\partial \mathbf{x}}$ is the gradient of $p_t(\mathbf{x})$.
- $\Delta p_t = \frac{\partial^2 p_t}{\partial \mathbf{x}}$ is the Laplacian.

Next, we consider the KL divergence between two generic distributions and define how it can be computed using score functions, a result which we will use later to infer information

measures of interests.

Proposition 1. *Let p_t and q_t be time-varying probability densities associated with similar SDE of the form defined in Eq. (4.1) with different initial conditions as $\mathbf{X}_0 \sim p$ and $\mathbf{X}_0 \sim q$.*

The KL divergence between two generic distributions $p(\mathbf{x})$ and $q(\mathbf{x})$, defined as

$$\mathbb{KL}\left(p(\mathbf{x}) \parallel q(\mathbf{x})\right) = \int p(\mathbf{x}) \log \frac{p(\mathbf{x})}{q(\mathbf{x})} \mathrm{d}\mathbf{x},$$

can be computed considering the time-varying score functions $\nabla \log p_t$ and $\nabla \log q_t$ in $[0, T]$ and satisfying the Fokker–Planck equations, according to the following expression:

$$\boxed{\mathbb{KL}\left(p(\mathbf{x}) \parallel q(\mathbf{x})\right) = \int p_t(\mathbf{x}) \frac{g^2(t)}{2} \|\nabla \log p_t(\mathbf{x}) - \nabla \log q_t(\mathbf{x})\|^2 \mathrm{d}\mathbf{x}\mathrm{d}t + \mathbb{KL}\left(p_T(\mathbf{x}) \parallel q_T(\mathbf{x})\right)}$$

Sketch Proof. Hereafter, we present an outline of the proof and differ details to Appendix B.1.2. We start by defining :

$$\mathbf{r}_t = \int p_t(\mathbf{x}) \log \frac{p_t(\mathbf{x})}{q_t(\mathbf{x})} \, \mathrm{d}\mathbf{x}. \tag{4.3}$$

To avoid clutter, we use a the simplified notation p_t and q_t instead of $p_t(\mathbf{x})$ and $q_t(\mathbf{x})$. Then, by definition, we have

$$\mathbf{r}_0 = \mathbb{KL}\left(p \parallel q\right) \quad \text{and} \quad \mathbf{r}_T = \mathbb{KL}\left(p_T \parallel q_T\right). \tag{4.4}$$

Note that $\mathbf{r}_T = \int p_T \log \frac{p_T}{q_T} \mathrm{d}x = \mathbb{KL}\left(p_T \parallel q_T\right)$ is a vanishing term, i.e. $\lim_{T\to\infty} \mathbb{KL}\left(p_T \parallel q_T\right) = 0$. To ground this claim, we borrow the results by Collet and Malrieu (2008), which hold for several forward diffusion SDEs of interest, such as the Variance Preserving (VP) or Variance Exploding (VE) SDEs (Song et al., 2021).

$$\int_0^T \frac{\mathrm{d}\mathbf{r}_t}{\mathrm{d}t} \, \mathrm{d}t = \mathbf{r}_T - \mathbf{r}_0 \quad \Longrightarrow \quad \mathbb{KL}\left(p \parallel q\right) = -\int_0^T \frac{\mathrm{d}\mathbf{r}_t}{\mathrm{d}t} \, \mathrm{d}t + \mathbb{KL}\left(p_T \parallel q_T\right). \tag{4.5}$$

Focusing on the term $C = \int_0^T \frac{\mathrm{d}\mathbf{r}_t}{\mathrm{d}t} \, \mathrm{d}t$ and using the Leibniz rule and the product rule, we obtain:

$$\frac{\mathrm{d}\mathbf{r}_t}{\mathrm{d}t} = \frac{\mathrm{d}}{\mathrm{d}t} \int p_t \log \frac{p_t}{q_t} \, \mathrm{d}\mathbf{x} = \int \frac{\partial p_t}{\partial t} \log \frac{p_t}{q_t} \, \mathrm{d}\mathbf{x} + \int p_t \frac{\partial}{\partial t} \log \frac{p_t}{q_t} \, \mathrm{d}\mathbf{x}. \tag{4.6}$$

Noting that

$$\int \frac{\partial p_t}{\partial t} \, \mathrm{d}\mathbf{x} = \frac{\mathrm{d}}{\mathrm{d}t} \int p_t \, \mathrm{d}\mathbf{x} = 0, \tag{4.7}$$

we can combine terms to write

$$C = \int \frac{\mathrm{d}\mathbf{r}_t}{\mathrm{d}t} = \int \frac{\partial p_t}{\partial t} \log \frac{p_t}{q_t} - \int \frac{p_t}{q_t} \frac{\partial q_t}{\partial t} \, \mathrm{d}\mathbf{x}\mathrm{d}t. \tag{4.8}$$

Using the Fokker–Planck compact form Eq. (4.2), we can write $C = C_1 + C_2$ with :

$$C_1 = -\int \log \frac{p_t}{q_t} \nabla \cdot \big(f(t)\,\mathbf{x}\,p_t\big)\,\mathrm{d}\mathbf{x} + \int \frac{p_t}{q_t} \nabla \cdot \big(f(t)\,\mathbf{x}\,q_t\big)\,\mathrm{d}\mathbf{x}\mathrm{d}t, \tag{4.9}$$

$$C_2 = \frac{1}{2}\int g^2(t)\,\Delta p_t \,\log \frac{p_t}{q_t}\,\mathrm{d}\mathbf{x} - \frac{1}{2}\int \frac{p_t}{q_t}\, g^2(t)\,\Delta q_t \,\mathrm{d}\mathbf{x}\mathrm{d}t. \tag{4.10}$$

Under appropriate boundary conditions (i.e., p_t and q_t vanish at $|\mathbf{x}| \to \infty$), an application of integration by parts shows that $C_1 = 0$. We apply the integration by parts on the term C_2 using the equality : $\int \Delta p_t \,\log \frac{p_t}{q_t}\,\mathrm{d}\mathbf{x} = -\int \nabla p_t \cdot \nabla \left(\log \frac{p_t}{q_t}\right) \mathrm{d}\mathbf{x}$. After substituting these expressions and rearranging we can write:

$$C = 0 + C_2 = -\frac{1}{2}\int g^2(t)\,p_t \left\| \nabla \log \frac{p_t}{q_t} \right\|^2 \mathrm{d}\mathbf{x}\,\mathrm{d}t. \tag{4.11}$$

□

The result in Proposition 1 allows, in principle, the exact computation of KL divergences, provided knowledge of the score functions $\nabla \log p_t, \nabla \log q_t$. Such knowledge is however out of reach in practical cases, which is why in this work we consider a *parametric* approximation leading to a KL divergence *estimator*. In particular, we leverage the methodology considered in (Song and Ermon, 2019; Song et al., 2021) where the parametric score s_θ is obtained by minimizing the *denoising score-matching* loss :

$$\mathcal{L}_{\mathrm{DSM}}(\theta) = \frac{1}{2}\int_0^T \mathbb{E}_{\substack{\mathbf{x}_0 \sim p(\mathbf{x}) \\ \mathbf{x} \sim p_{0t}(\mathbf{x}|\mathbf{x}_0)}} \left[g^2(t) \| s_\theta(\mathbf{x}, t) - \nabla_{\mathbf{x}} \log p_{0t}(\mathbf{x}|\mathbf{x}_0) \|^2 \right] \mathrm{d}t \tag{4.12}$$

where $p_{0t}(\mathbf{x} \,|\, \mathbf{x}_0)$ is the conditional distribution of the noised random variable *given* initial conditions $\mathbf{X}_0 \sim p(\mathbf{x})$, i.e. $p_t(\mathbf{x}) = \int p_{0t}(\mathbf{x} \,|\, \mathbf{x}_0) p(\mathbf{x}_0) \mathrm{d}\mathbf{x}$. Note that p_{0t} has known Gaussian distribution with known mean $\mu(t)$ and variance $\sigma(t)$. This allows, with the knowledge of the score functions, the implementation of an estimator for the KL divergence.

Remark: In (Franzese, Bounoua, and Michiardi, 2024), it is shown that, under certain conditions, divergences between probability distributions can be computed equivalently in a latent space. By defining an encoder–decoder pair that enables near-perfect reconstruction, it becomes possible to compute the divergences directly in the latent space, a property we

leverage in § 4.6.2.

4.3 Theoretical Guarantees

Given the parametric approximations of the score networks through minimization of Eq. (4.12), and the result in Proposition 1, we are ready to discuss our proposed **estimator** of the KL divergence. We note $s^p_*(\mathbf{x}, t) = \nabla \log p_t(\mathbf{x})$ and $s^q_*(\mathbf{x}, t) = \nabla \log q_t(\mathbf{x})$ the *true* score function the *parametrized* learned score functions and $s^p(\mathbf{x}, t)$ and $s^q(\mathbf{x}, t)$ are learned score functions (Typically by neural networks). Using the parametric approximations of the score networks we have the following estimator:

$$\widetilde{\mathbb{KL}}\left(p \parallel q\right) = \int_0^T \mathbb{E}_{\mathbf{x}\sim p_t(\mathbf{x})}\left[\frac{g^2(t)}{2}\|s^p(\mathbf{x}, t) - s^q(\mathbf{x}, t)\|^2 \mathrm{d}t\right] + \mathbb{KL}\left(p_T(\mathbf{x}) \parallel q_T(\mathbf{x})\right) \tag{4.13}$$

By defining the score error as $\mathbf{e}^p_t(\mathbf{x}) \overset{\text{def}}{=} s^p(\mathbf{x}, t) - s^p_*(\mathbf{x}, t)$ and $\mathbf{e}^q_t(\mathbf{x}) \overset{\text{def}}{=} s^q_*(\mathbf{x}, t) - s^q(\mathbf{x}, t)$, and considering that $r_T = 0$ it is possible to show (see Appendix B.1.3 for proof) that :

$$\mathbb{KL}\left(p \parallel q\right) - \widetilde{\mathbb{KL}}\left(p \parallel q\right) = \int_0^T \frac{g^2(t)}{2}\mathbb{E}_{\mathbf{x}\sim p_t(\mathbf{x})}\Big[\|\mathbf{e}^p_t(\mathbf{x}) - \mathbf{e}^q_t(\mathbf{x})\|^2 \tag{4.14}$$

$$+ 2\langle\, s^p(\mathbf{x}, t) - s^q(\mathbf{x}, t),\, \mathbf{e}^p_t(\mathbf{x}) - \mathbf{e}^q_t(\mathbf{x})\,\rangle \mathrm{d}t\Big] \tag{4.15}$$

An important property of our estimator is that it is *neither* an upper nor a lower bound of the true KL divergence: indeed the approximation gap Eq. (4.14) can be either positive or negative. This property frees our estimation guarantees from the pessimistic results of (McAllester and Stratos, 2020). Note also that, counter-intuitively, larger errors norms $\|\mathbf{e}^p_t(\mathbf{x})\|$ not necessarily imply larger estimation error of the KL divergence. Indeed, common mode errors (reminiscent of paired statistical tests) cancel out. In the special case where $\mathbf{e}^p_t(\mathbf{x}) = \mathbf{e}^q_t(\mathbf{x})$, the estimation error due to the approximate nature of the score functions is indeed zero.

Qualitatively, we observe that our estimator is affected by two sources of error: score networks that only approximate the true score function and finiteness of T. The approximation gap Eq. (4.14) which is related to the score discrepancy, suggests selection of a small time T (indeed we can expect such mismatch to behave as a quantity that increases with T (Franzese et al., 2023)). It is important however to adopt a sufficiently large diffusion time T such that $\mathbb{KL}\left(p_T \parallel q_T\right)$ vanishes. Typical diffusion schedules satisfy these requirements.

Montecarlo Integration The analytical computation of Eq. (4.13) is, in general, out of reach. However, Montecarlo integration is possible, by recognizing that samples from p_t can be obtained through the sampling scheme $\mathbf{X}_0 \sim p, \mathbf{X}_t \sim p_{0t}(\mathbf{x} \mid \mathbf{x}_0)$. The outer integration w.r.t. to the time instant is similarly possible by sampling $t \sim \mathcal{U}(0, T)$, and multiplying the result of the estimation by T (since $\int_0^T (\cdot)\mathrm{d}t = T\mathbb{E}_{t\sim\mathcal{U}(0,T)}[(\cdot)]$). Alternatively, it is possible to implement importance sampling schemes to reduce the variance, along the lines of what described by (Huang, Lim, and Courville, 2021), by sampling the time instant non-uniformly and modifying accordingly the time-varying constants in Eq. (4.13). In both cases, the Montecarlo estimation error can be reduced to arbitrary small values by collecting enough samples, with guarantees described in (Rainforth et al., 2018).

4.4 Entropy estimation

We now describe how to compute the entropy associated to a given density p. Recall that entropy quantify the randomness of random variable $\mathbf{X}$ associated with the probability density $p(\mathbf{x})$.

$$\mathcal{H}(p) = -\int p(\mathbf{x}) \log p(\mathbf{x}) \mathrm{d}\mathbf{x} \tag{4.16}$$

Using the ideas for estimating the KL divergence, we notice that we can compute $\mathbb{KL}\left(p \parallel \gamma^\sigma\right)$, where $\gamma^\sigma(\mathbf{x})$ stands for the standard Gaussian distribution with mean 0 and covariance $\sigma^2\mathbf{I}$. Then, we can relate the entropy to such divergence through cross entropy:

$$\begin{aligned}
\mathcal{H}(p, \gamma^\sigma) &= \mathcal{H}(p) + \mathbb{KL}\left(p \parallel \gamma^\sigma\right) && (4.17)\\
&= -\int p(\mathbf{x}) \log \gamma^\sigma(\mathbf{x}) \mathrm{d}\mathbf{x} && (4.18)\\
&= \frac{N}{2} \log\left(2\pi\sigma^2\right) + \frac{\mathbb{E}_{p(\mathbf{x})}\left[(\mathbf{X}_0)^2\right]}{2\sigma^2}. && (4.19)
\end{aligned}$$

With N being the dimension of $\mathbf{X}$. Hence we can write :

$$\mathcal{H}(p) = -\mathbb{KL}\left(p \parallel \gamma^\sigma\right) + \frac{N}{2} \log\left(2\pi\sigma^2\right) + \frac{\mathbb{E}_{p(\mathbf{x})}\left[(\mathbf{X}_0)^2\right]}{2\sigma^2}. \tag{4.20}$$

A simple manipulation of Eq. (4.20), using the results from § 4.3, we can obtain the following entropy estimator :

$$\mathcal{H}^{\sigma}(p) \simeq -\int_0^T \mathbb{E}_{\mathbf{x}\sim p_t(\mathbf{x})}\left[\frac{g^2(t)}{2}\left\|s^{p}(\mathbf{x},t)-s^{\gamma^{\sigma}}(\mathbf{x},t)\right\|^2 \mathrm{d}t\right] - \mathbb{KL}\left(p_T \parallel \gamma_T^{\sigma}\right) + \frac{N}{2}\log\left(2\pi\sigma^2\right) + \frac{\mathbb{E}_{p(\mathbf{x})}\left[(\mathbf{X}_0)^2\right]}{2\sigma^2} \tag{4.21}$$

Now, the score function associated to the forward process starting from γ^{σ} is analytically known and has value $s^{\gamma^{\sigma}}(\mathbf{x},t) = -\chi_t^{-1}\mathbf{x}$, where $\chi_t = \left(k_t^2\sigma^2 + k_t^2\int_0^t k_s^{-2}g^2(s)\mathrm{d}s\right) I$, with $k_t = \exp\left\{\left(\int_0^t f(s)\mathrm{d}s\right)\right\}$. Moreover, whenever T is large enough $p_T \simeq \gamma^1$, independently on the chosen value of σ. Consequently $\mathbb{KL}\left(p_T \parallel \gamma_T^{\sigma}\right) \simeq \mathbb{KL}\left(\gamma^1 \parallel \gamma^{\sqrt{\chi_T}}\right)$, which is analytically available as $N/2\left(\log\left(\chi_T\right) - 1 + 1/\chi_T\right)$. Quantification of such approximation is possible following the same lines defined by (Collet and Malrieu, 2008). In summary, we consider the following estimator for the entropy:

$$\mathcal{H}^{\sigma}(p) \simeq -\mathbb{E}_{\mathbf{x}\sim p_t(\mathbf{x})}\left[\int_0^T \frac{g^2(t)}{2}\left\|s^{p}(\mathbf{x},t)+\chi_t^{-1}\mathbf{x}\right\|^2 \mathrm{d}t\right] - \frac{N}{2}\left(\log\left(\chi_T\right) - 1 + \frac{1}{\chi_T}\right) + \frac{N}{2}\log\left(2\pi\sigma^2\right) + \frac{\mathbb{E}_{p(\mathbf{x})}\left[(\mathbf{X}_0)^2\right]}{2\sigma^2} \tag{4.22}$$

For completeness, we note that a related estimator has recently appeared in the literature (Kong, Brekelmans, and Ver Steeg, 2022), although the technical derivation and objectives are different than ours.

4.5 Mutual Information Estimation

In this work, we are interested in estimating the MI between two random variables $\mathbf{X}$ and $\mathbf{Y}$. Consequently, we need to define the joint, conditional, and marginal probability densities. We denote the marginal probability density of the first random variable $\mathbf{X} \in \mathbb{R}^N$ as $p^{\mathbf{X}}(\mathbf{x})$. Similarly, the marginal probability density of the second random variable $\mathbf{Y} \in \mathbb{R}^N$ is denoted by $p^{\mathbf{Y}}(\mathbf{y})$. The joint probability density of the two random variables, $\left[\mathbf{X},\, \mathbf{Y}\right] \in \mathbb{R}^{2N}$ is denoted by $p_{\mathbf{X},\mathbf{Y}}(\mathbf{x},\mathbf{y})$. What remains to be specified are the conditional probability densities. The density of $\mathbf{X}$ given that $\mathbf{Y} = \mathbf{y}$ is denoted by $p^{\mathbf{X}|\mathbf{Y}}(\mathbf{x} \mid \mathbf{y})$, and the density of $\mathbf{Y}$ given that $\mathbf{X} = \mathbf{x}$ is denoted by $p^{\mathbf{Y}|\mathbf{X}}(\mathbf{y} \mid \mathbf{x})$. This choice of notation, along with Bayes' theorem, implies the following set of equivalences: $p^{\mathbf{X},\mathbf{Y}}(\mathbf{x},\mathbf{y}) = p^{\mathbf{X}|\mathbf{Y}}(\mathbf{x} \mid \mathbf{y})\, p^{\mathbf{Y}}(\mathbf{y}) = p^{\mathbf{Y}|\mathbf{X}}(\mathbf{y} \mid \mathbf{x})\, p^{\mathbf{X}}(\mathbf{x})$, and the marginal densities can be recovered as $p^{\mathbf{X}}(\mathbf{x}) = \int p^{\mathbf{X}|\mathbf{Y}}(\mathbf{x} \mid \mathbf{y})\, p^{\mathbf{Y}}(\mathbf{y})\, \mathrm{d}\mathbf{y}$, $\quad p^{\mathbf{Y}}(\mathbf{y}) = \int p^{\mathbf{Y}|\mathbf{X}}(\mathbf{y} \mid \mathbf{x})\, p^{\mathbf{X}}(\mathbf{x})\, \mathrm{d}\mathbf{x}$.

4.5.1 Approximating the Conditional and Joint Score Functions

The marginals $p^{\mathbf{X}}, p^{\mathbf{Y}}$ are associated to diffusion of the form of Eq. (4.1). Similarly, the joint $p^{\mathbf{X}}$ and conditionals $p^{\mathbf{X}|\mathbf{Y}}$ we introduced, are associated to forward diffusion processes:

$$\begin{cases} \mathrm{d}\left[\mathbf{X}_t, \mathbf{Y}_t\right]^\top = f(t)[\mathbf{X}_t, \mathbf{Y}_t]^\top \mathrm{d}t + g(t)\left[\mathrm{d}\mathbf{W}_t, \mathrm{d}\mathbf{W}'_t\right]^\top \\ [\mathbf{X}_0, \mathbf{Y}_0]^\top \sim p^{\mathbf{X},\mathbf{Y}} \end{cases}, \quad \begin{cases} \mathrm{d}\mathbf{X}_t = f(t)\mathbf{X}_t \mathrm{d}t + g(t)\mathrm{d}\mathbf{W}_t \\ \mathbf{X}_0 \sim p^{\mathbf{X}|\mathbf{Y}} \end{cases} \tag{4.23}$$

respectively, where the SDE on the l.h.s. is valid for the real space $\mathbb{R}^{2N}$, as defined in Eq. (4.13).

In this work, we consider two classes of diffusion processes. In the first case, the diffusion model is asymmetric, and the random variable $\mathbf{Y}$ is only considered as a conditioning signal. As such, we learn the score associated to the random variable $\mathbf{X}$, with a conditioning signal $\mathbf{X}$, which is set to some predefined null value when considering the marginal case. This well-known approach (Ho and Salimans, 2022) effectively models the marginal and conditional scores associated to $p_t^{\mathbf{X}}$ and $p_t^{\mathbf{X}|\mathbf{Y}}$ with a unique score network.

Next, we define a new kind of diffusion model for the joint random variable $[\mathbf{X}, \mathbf{Y}]$, which allows modeling the joint and the conditional measures. Inspired by recent trends in multi-modal generative modeling (Bao et al., 2023; Bounoua, Franzese, and Michiardi, 2024), we define a joint diffusion process that allows amortized training of a single score network, instead of considering separate diffusion processes and their respective score networks, for each random variable. To do so, we define the following SDE:

$$\begin{cases} \mathrm{d}\left[\mathbf{X}_t, \mathbf{Y}_t\right]^\top = f(t)[\alpha\mathbf{X}_t, \beta\mathbf{Y}_t]^\top \mathrm{d}t + g(t)\left[\alpha\mathrm{d}\mathbf{W}_t, \beta\mathrm{d}\mathbf{W}'_t\right]^\top \\ [\mathbf{X}_0, \mathbf{Y}_0]^\top \sim p^{\mathbf{X},\mathbf{Y}} \end{cases} \tag{4.24}$$

with extra parameters $\alpha, \beta \in \{0, 1\}$. This SDE extends the l.h.s. of Eq. (4.23), and describes the joint evolution of the variables $\mathbf{X}_t, \mathbf{Y}_t$, starting from the joint $p^{\mathbf{X}\mathbf{Y}}$, with time varying probability density $p_t^{\mathbf{X}\mathbf{Y}}$. The two extra coefficients α, β are used to modulate the *speed* at which the two portions $\mathbf{X}_t, \mathbf{Y}_t$ of the process diffuse towards their steady state. More precisely, $\alpha = \beta = 1$ corresponds to a *classical* simultaneous diffusion (l.h.s. of Eq. (4.23)). On the other hand, the configuration $\alpha = 1, \beta = 0$ corresponds to the case in which the variable $\mathbf{Y}_t$ remains constant throughout all the diffusion (which is used for conditional measures, r.h.s. of Eq. (4.23)). The specular case, $\alpha = 0, \beta = 1$, similarly allows to study the evolution of $\mathbf{Y}_t$ conditioned on a constant value of $\mathbf{X}_0$. Then, instead of learning three separate score networks (for $p^{\mathbf{X},\mathbf{Y}}, p^{\mathbf{X}|\mathbf{Y}}$ and $p^{\mathbf{Y}|\mathbf{X}}$), associated to standard diffusion processes, the key idea is to consider a *unique* parametric score, leveraging the

unified formulation Eq. (4.24), which accepts as inputs two vectors in $\mathbb{R}^N$, the diffusion time t, and the two coefficients α, β. This allows to conflate in a single architecture: i) the score $s^{\mathbf{X},\mathbf{Y}}([\mathbf{x},\mathbf{y}],t)$ associated to the joint diffusion of the variables $\mathbf{X},\mathbf{Y}$ (corresponding to $\alpha=\beta=1$) and ii) the conditional score $s^{\mathbf{X}|\mathbf{Y}}(\mathbf{x},\mathbf{y},t)$ (corresponding to $\alpha=1,\beta=0$). (To simplify the notation, instead of writing $s^{p^{\mathbf{X},\mathbf{Y}}}$ and $s^{p^{\mathbf{X}|\mathbf{Y}}}$, we write $s^{\mathbf{X},\mathbf{Y}}$ and $s^{\mathbf{X}|\mathbf{Y}}$) Additional details are presented in Appendix B.2.

4.5.2 MINDE: a Family of MI estimators

We are now ready to describe our new MI estimator, which we call MINDE. As a starting point, we recognize that the MI between two random variables $\mathbf{X},\mathbf{Y}$ has several equivalent expressions, among which Eqs. (4.25) to (4.28). On the left hand side of these expressions we report well-known formulations for the MI, $\mathcal{I}(\mathbf{X},\mathbf{Y})$, while on the right hand side we express them using the estimators we introduce in this work, where equality is assumed to be valid up to the errors described in § 4.3. We can leverage the different formulations of MI in terms of entropies or as a KL divergence.

We have $\mathcal{I}(\mathbf{X},\mathbf{Y}) = \mathcal{H}(p^{\mathbf{X}}) - \mathcal{H}(p^{\mathbf{X}|\mathbf{Y}})$ or $\mathcal{I}(\mathbf{X},\mathbf{Y}) = \mathcal{H}(p^{\mathbf{X},\mathbf{Y}}) - \mathcal{H}(p^{\mathbf{X}|\mathbf{Y}}) - \mathcal{H}(p^{\mathbf{Y}|\mathbf{X}})$. Note that by substituting the entropy estimator Eq. (4.22) in the MI formulation the other terms vanishes leaving only the difference of scores.

Hence, we can write:

$$\begin{aligned}
\mathcal{I}(\mathbf{X},\mathbf{Y}) &\simeq -\widetilde{\mathbb{KL}}\left(p^{\mathbf{X}} \parallel \gamma^{\sigma}\right) + \widetilde{\mathbb{KL}}\left(p^{\mathbf{X}|\mathbf{Y}} \parallel \gamma^{\sigma}\right) \\
&\simeq -\int_0^T \mathbb{E}_{\mathbf{x}\sim p_t^{\mathbf{X}}}\left[\frac{g^2(t)}{2}\left\|s^{\mathbf{X}}(\mathbf{x},t) + \chi_t^{-1}\mathbf{x}\right\|^2\right] \mathrm{d}t \\
&+ \int_0^T \mathbb{E}_{\substack{\mathbf{y}_0\sim p^{\mathbf{Y}} \\ \mathbf{x}\sim p_t^{\mathbf{X}|\mathbf{Y}=\mathbf{y}_0}}}\left[\frac{g^2(t)}{2}\left\|s^{\mathbf{X}|\mathbf{Y}=\mathbf{y}_0}(\mathbf{x},\mathbf{y}_0,t) + \chi_t^{-1}\mathbf{x}\right\|^2\right] \mathrm{d}t
\end{aligned} \tag{4.25}$$

The joint and conditional scores can also be used :

$$\begin{aligned}\mathcal{I}(\mathbf{X},\mathbf{Y}) &\simeq -\widetilde{\mathbb{KL}}\left(p^{\mathbf{X},\mathbf{Y}} \parallel \gamma^{\sigma}\right) + \widetilde{\mathbb{KL}}\left(p^{\mathbf{X}|\mathbf{Y}} \parallel \gamma^{\sigma}\right) + \widetilde{\mathbb{KL}}\left(p^{\mathbf{Y}|\mathbf{X}} \parallel \gamma^{\sigma}\right) \\ &\simeq -\int_0^T \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p_t^{\mathbf{X},\mathbf{Y}}}\left[\frac{g^2(t)}{2}\left\|s^{\mathbf{X},\mathbf{Y}}(\mathbf{x},\mathbf{y},t) + \chi_t^{-1}[\mathbf{x},\mathbf{y}]^\top\right\|^2\right] \mathrm{d}t \\ &+ \int_0^T \mathbb{E}_{\substack{\mathbf{y}_0\sim p^{\mathbf{Y}} \\ \mathbf{x}\sim p_t^{\mathbf{X}|\mathbf{Y}=\mathbf{y}_0}}}\left[\frac{g^2(t)}{2}\left\|s^{\mathbf{X}|\mathbf{Y}}(\mathbf{x},\mathbf{y}_0,t) + \chi_t^{-1}\mathbf{x}\right\|^2\right] \mathrm{d}t \\ &+ \int_0^T \mathbb{E}_{\substack{\mathbf{x}_0\sim p^{\mathbf{X}} \\ \mathbf{y}\sim p_t^{\mathbf{Y}|\mathbf{X}=\mathbf{x}_0}}}\left[\int_0^T \frac{g^2(t)}{2}\left\|s^{\mathbf{Y}|\mathbf{X}}(\mathbf{y},\mathbf{x}_0,t) + \chi_t^{-1}\mathbf{y}\right\|^2\right] \mathrm{d}t \end{aligned} \tag{4.26}$$

It's also possible to use another formulation of MI using the conditional and unconditional scores without the reference score by starting from the KL formulation. Approximating the true scores with $s^{\mathbf{X}}$, $s^{\mathbf{X}|\mathbf{Y}}$, and assuming that T is large enough such that $\mathbb{KL}\left(p_T^{\mathbf{X}|\mathbf{Y}} \parallel p_T^{\mathbf{X}}\right) \to 0$ (under standard Gaussian terminal), we can obtain another estimator for MI:

$$\begin{aligned}\mathcal{I}(\mathbf{X},\mathbf{Y}) &= \mathbb{E}_{\mathbf{y}\sim p^{\mathbf{Y}}}\,\mathbb{KL}\left(p^{\mathbf{X}|\mathbf{Y}=\mathbf{y}} \parallel p^{\mathbf{X}}\right) \\ &\simeq \int_0^T \mathbb{E}_{\substack{\mathbf{y}_0\sim p^{\mathbf{Y}} \\ \mathbf{x}\sim p_t^{\mathbf{X}|\mathbf{Y}=\mathbf{y}_0}}}\left[\frac{g^2(t)}{2}\left\|s^{\mathbf{X}|\mathbf{Y}}(\mathbf{x},\mathbf{y}_0,t) - s^{\mathbf{X}}(\mathbf{x},t)\right\|^2\right] \mathrm{d}t \end{aligned} \tag{4.27}$$

Another estimator using the joint score and the conditional scores without the reference Gaussian distribution can be obtained (see Appendix B.1.4 for proof):

$$\mathcal{I}(\mathbf{X},\mathbf{Y}) \simeq \int_0^T \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p_t^{\mathbf{X},\mathbf{Y}}}\left[\frac{g^2(t)}{2}\left\|s^{\mathbf{X},\mathbf{Y}}(\mathbf{x},\mathbf{y},t) - [s^{\mathbf{X}|\mathbf{Y}}(\mathbf{x},\mathbf{y}_0,t), s^{\mathbf{Y}|\mathbf{X}}(\mathbf{y},\mathbf{x}_0,t)]^\top\right\|\right] \mathrm{d}t \tag{4.28}$$

4.6 Experimental Validation

We now evaluate the different estimators proposed in § 4.5. In particular, we study conditional and joint models (MINDE-C and MINDE-J respectively), and variants that exploit the difference between the parametric scores *inside* the same norm (Eqs. (4.27) and (4.28)) or *outside* it, adopting the difference of entropies representation along with Gaussian reference distribution (Eqs. (4.25) and (4.26)).

Summarizing, we refer to the different variants as MINDE-C(σ), MINDE-C, and MINDE--J(σ), MINDE-J, for Eqs. (4.25) to (4.28) respectively. Our empirical validation involves a large range of synthetic distributions, which we present in § 4.6.1. We also analyze the behavior of all MINDE variants according to *self-consistency* tests, as discussed in § 4.6.2.

For all the settings, we use a simple, stacked multi-layer perception (MLP) with skip connections adapted to the input dimensions, and adopt VP-SDE diffusion (Song et al., 2021). We apply importance sampling (Huang, Lim, and Courville, 2021; Song et al., 2021) at both training and inference time. More details about the implementation are included in Appendix B.2.

4.6.1 MI Estimation Benchmark

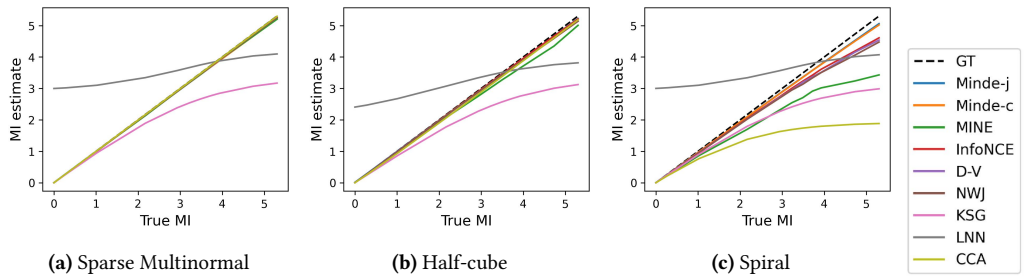
We use the evaluation strategy proposed by Czyż et al. (2023), which covers a range of distributions going beyond what is typically used to benchmark MI estimators, e.g., multivariate normal distributions. In summary, we consider high-dimensional cases with (possibly) long-tailed distributions and/or sparse interactions, in the presence of several non trivial non-linear transformation. Benchmarks are constructed using samples from several base distributions, including Uniform, Normal with either dense or sparse correlation structure, and long-tailed Student distributions. Such samples are further modified by deterministic transformations, including the Half-Cube homeomorphism, which extends the distribution tails, and the Asinh Mapping, which instead shortens them, the Swiss Roll Embedding and Spiral diffeomorphis, which alter the simple linear structure of the base distributions.

We compare MINDE against neural estimators, such as MINE (Belghazi et al., 2018), INFONCE (Oord, Li, and Vinyals, 2018),NWJ (Nguyen, Wainwright, and Jordan, 2007) and DOE (McAllester and Stratos, 2020). To ensure a fair comparison between MINDE and other neural competitors, we consider architectures with a comparable number of parameters. Note that the original benchmark in (Czyż et al., 2023) uses 10k training samples, which are in many cases not sufficient to obtain stable estimates of the MI for our competitors. Here, we use a larger training size (100k samples) to avoid confounding factors in our analysis. In all our experiments, we fix $\sigma = 1.0$ for the MINDE-C(σ), MINDE-J(σ) variants, which results in the best performance (an ablation study is included in Appendix B.3).

Results: The general benchmark consists of 40 tasks (10 unique tasks$\times$ 4 parametrizations) designed by combining distributions and MI-invariant transformations discussed earlier. We average results over 10 seeds for MINDE variants and competitors, following the same protocol as in (Czyż et al., 2023). We present the full set of MI estimation tasks in Table 4.1.

As in the original (Czyż et al., 2023), estimates for the different methods are presented with a precision of 0.1 nats, to improve visualization. For low-dimensional distributions, benchmark results show that all methods are effective in accurate MI estimation. Differences emerge for more challenging scenarios. Overall, all our MINDE variants perform well. MINDE–C stands out as the best estimator with 35/40 estimated tasks with an error within the 0.1 nats quantization range. Moreover, MINDE can accurately estimate the MI for long tailed distributions (Student) and highly transformed distributions (Spiral, Normal CDF), which are instead problematic for most of the other methods. The MINE estimator achieves the second best performance, with an MI estimation within 0.1 nats from ground truth for 24/40 tasks. Similarly to the other neural estimator baselines, MINE is limited when dealing with long tail distributions (Student), and significantly transformed distributions (Spiral).

High MI benchmark: Through this second benchmark, we target high MI distributions. We consider 3×3 multivariate normal distribution with sparse interactions as done in (Czyż et al., 2023). We vary the correlation parameter to obtain the desired MI, and test the estimators when applying Half-cube or Spiral transformations. Results in Figure 4.1 show that while on the non transformed distribution (column (a)) all neural estimators nicely follow the ground truth, on the transformed versions (columns (b) and (c)), MINDE outperforms competitors.



(a) Sparse Multinormal **(b)** Half-cube **(c)** Spiral

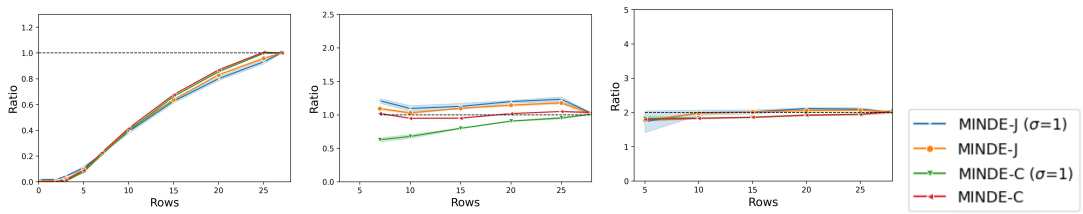
Figure 4.1: High MI benchmark: original (column (a)) and transformed variants (columns (b) and (c)).

4.6.2 Consistency tests

The second set of tests we perform are the self-consistency ones proposed in (Song and Ermon, 2019), which aim at investigating properties of MI estimators on real data. Considering as random variable A a sample from the MNIST (resolution 28×28) dataset, the first set of measurements performed is the estimation of $\mathcal{I}(\mathbf{X}, \mathbf{Y}_i)$, where $\mathbf{Y}(i)$ is equal to $\mathbf{X}$ for the first i rows, and set to 0 afterwards. It is evident that $\mathcal{I}(\mathbf{X}, \mathbf{Y}(i))$ is a quantity that increases with r, where in particular $\mathcal{I}(\mathbf{X}, \mathbf{Y}(0)) = 0$. Testing whether this holds also for the estimated MI is referred to as *independency* test. The second test proposed in (Song and Ermon, 2019) is

the *data-processing* test, where given that $\mathcal{I}(\mathbf{X}; [\mathbf{Y}(i+k), \mathbf{Y}(i)]) = \mathcal{I}(\mathbf{X}; \mathbf{Y}(i+k)), k > 0$, the task is to verify it through estimators for different values of k. Finally, the *additivity* tests aim at assessing whether for two independent images $\mathbf{X}, \hat{\mathbf{X}}$ extracted from the dataset, the property $\mathcal{I}([\mathbf{X}, \hat{\mathbf{X}}]; [\mathbf{Y}(i), \hat{\mathbf{Y}}(i)]) = 2\,\mathcal{I}(\mathbf{X}; \mathbf{Y}(i))$ is satisfied also by the numerical estimations.

For these tests, we consider diffusion models in a latent space, exploiting the invariance of KL divergences to perfect auto-encoding. First, we train deterministic auto-encoders for the considered images in all the tests . Then, through concatenation of the latent variables, as done in (Bao et al., 2023; Bounoua, Franzese, and Michiardi, 2024), we compute the MI with the different schemes proposed in § 4.5. Results of the three tests (averaged over 5 seeds) are reported in Figure 4.2. In general, all MINDE variants show excellent performance, whereas none of the other neural MI estimators succeed at passing simultaneously all tests, as can be observed from Figures 4,5,6 in the original (Song and Ermon, 2019).



(a) Baseline test **(b)** Data processing test **(c)** Additivity test

Figure 4.2: Consistency tests results on the MNIST dataset. *Baseline test Figure 4.2a:* Evaluation of $\frac{\mathcal{I}(\mathbf{X},\mathbf{Y}_i)}{I(\mathbf{X},\mathbf{Y}(0))}$. $\mathbf{X}$ is an image and $\mathbf{Y}(i)$ is an image containing the top i rows of $\mathbf{X}$. *Data processing test Figure 4.2b:* Evaluation of $\frac{\mathcal{I}(\mathbf{X},[\mathbf{Y}_{i+k},\mathbf{Y}(i)])}{I(\mathbf{X},\mathbf{Y}(i+k))}$ (ideal value is 1). *Additivity test Figure 4.2c:* Evaluation of $\frac{\mathcal{I}([\mathbf{X},\hat{\mathbf{X}}];[\mathbf{Y}(i),\hat{\mathbf{Y}}(i)])}{\mathcal{I}(\mathbf{X};\mathbf{Y}(i))}$ (ideal value is 2).

4.6.3 Analysis of conditional diffusion dynamics using MINDE

Diffusion models have achieved outstanding success in generating high-quality images, text, audio, and video across various domains. Recently, the generation of diverse and realistic data modalities (images, videos, sound) from open-ended text prompts (Ramesh et al., 2022; Saharia et al., 2022; Rombach et al., 2022) has projected practitioners into a whole new paradigm for content creation. A remarkable property of our MINDE method is its generalization to any score based model. Then, our method can be considered as a plug and play tool to explore information theoretic properties of score-based diffusion models: in particular, in this section we use MINDE to estimate MI in order to explain the dynamics of image conditional generation, by analyzing the influence of the prompt on the image generation through time.

	Asinh @ *St* 1 × 1 (dof=1)	Asinh @ *St* 2 × 2 (dof=1)	Asinh @ *St* 3 × 3 (dof=2)	Asinh @ *St* 5 × 5 (dof=2)	Bimodal 1 × 1	Bivariate *Nm* 1 × 1	*Hc* @ Bivariate *Nm* 1 × 1	*Hc* @ *Mn* 25 × 25 (2-pair)	*Hc* @ *Mn* 3 × 3 (2-pair)	*Hc* @ *Mn* 5 × 5 (2-pair)	*Mn* 2 × 2 (2-pair)	*Mn* 2 × 2 (dense)	*Mn* 25 × 25 (2-pair)	*Mn* 25 × 25 (dense)	*Mn* 3 × 3 (2-pair)	*Mn* 3 × 3 (dense)	*Mn* 5 × 5 (2-pair)	*Mn* 5 × 5 (dense)	*Mn* 50 × 50 (dense)	*Nm* CDF @ Bivariate *Nm* 1 × 1	*Nm* CDF @ *Mn* 25 × 25 (2-pair)	*Nm* CDF @ *Mn* 3 × 3 (2-pair)	*Nm* CDF @ *Mn* 5 × 5 (2-pair)	*Sp* @ *Mn* 25 × 25 (2-pair)	*Sp* @ *Mn* 3 × 3 (2-pair)	*Sp* @ *Mn* 5 × 5 (2-pair)	*Sp* @ *Nm* CDF @ *Mn* 25 × 25 (2-pair)	*Sp* @ *Nm* CDF @ *Mn* 3 × 3 (2-pair)	*Sp* @ *Nm* CDF @ *Mn* 5 × 5 (2-pair)	*St* 1 × 1 (dof=1)	*St* 2 × 2 (dof=1)	*St* 2 × 2 (dof=2)	*St* 3 × 3 (dof=2)	*St* 3 × 3 (dof=3)	*St* 5 × 5 (dof=2)	*St* 5 × 5 (dof=3)	Swiss roll 2 × 1	Uniform 1 × 1 (additive noise=0.1)	Uniform 1 × 1 (additive noise=0.75)	Wiggly @ Bivariate *Nm* 1 × 1
GT	0.2	0.4	0.3	0.4	0.4	0.4	0.4	1.0	1.0	1.0	1.0	0.3	1.0	1.3	1.0	0.4	1.0	0.6	1.6	0.4	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.2	0.4	0.2	0.3	0.2	0.4	0.3	0.4	1.7	0.3	0.4
MINDE–J ($\sigma = 1$)	0.2	0.4	0.3	0.4	0.4	0.4	0.4	1.1	1.0	1.0	1.0	0.3	0.9	1.2	1.0	0.4	1.0	0.6	1.7	0.4	1.0	1.0	1.0	0.9	0.9	0.9	1.0	0.9	1.0	0.2	0.4	0.2	0.3	0.2	0.5	0.3	0.5	1.6	0.3	0.4
MINDE–J	0.2	0.4	0.3	0.4	0.4	0.4	0.4	1.2	1.0	1.0	1.0	0.3	1.0	1.3	1.0	0.4	1.0	0.6	1.7	0.4	1.1	1.0	1.0	1.0	0.9	0.9	1.1	1.0	1.0	0.1	0.2	0.2	0.3	0.2	0.5	0.3	0.4	1.7	0.3	0.4
MINDE–C ($\sigma = 1$)	0.2	0.4	0.3	0.4	0.4	0.4	0.4	1.0	1.0	1.0	1.0	0.3	1.0	1.3	1.0	0.4	1.0	0.6	1.6	0.4	0.9	1.0	1.0	0.9	0.9	0.9	0.9	1.0	0.9	0.1	0.3	0.2	0.3	0.2	0.4	0.3	0.3	1.7	0.3	0.4
MINDE–C	0.2	0.4	0.3	0.4	0.4	0.4	0.4	1.0	1.0	1.0	1.0	0.3	1.0	1.3	1.0	0.4	1.0	0.6	1.6	0.4	1.0	1.0	1.0	0.9	0.9	0.9	1.0	1.0	1.0	0.1	0.3	0.2	0.3	0.2	0.4	0.3	0.4	1.7	0.3	0.4
MINE	0.2	0.4	0.2	0.4	0.4	0.4	0.4	1.0	1.0	1.0	1.0	0.3	1.0	1.3	1.0	0.4	1.0	0.6	1.6	0.4	0.9	0.9	0.9	0.8	0.7	0.6	0.9	0.9	0.9	0.0	0.0	0.1	0.1	0.1	0.2	0.2	0.4	1.7	0.3	0.4
InfoNCE	0.2	0.4	0.3	0.4	0.4	0.4	0.4	1.0	1.0	1.0	1.0	0.3	1.0	1.3	1.0	0.4	1.0	0.6	1.6	0.4	0.9	1.0	1.0	0.8	0.8	0.8	0.9	1.0	1.0	0.2	0.3	0.2	0.3	0.2	0.4	0.3	0.4	1.7	0.3	0.4
D-V	0.2	0.4	0.3	0.4	0.4	0.4	0.4	1.0	1.0	1.0	1.0	0.3	1.0	1.3	1.0	0.4	1.0	0.6	1.6	0.4	0.9	1.0	1.0	0.8	0.8	0.8	0.9	1.0	1.0	0.0	0.0	0.1	0.1	0.2	0.2	0.2	0.4	1.7	0.3	0.4
NWJ	0.2	0.4	0.3	0.4	0.4	0.4	0.4	1.0	1.0	1.0	1.0	0.3	1.0	1.3	1.0	0.4	1.0	0.6	1.6	0.4	0.9	1.0	1.0	0.8	0.8	0.8	0.9	1.0	1.0	0.0	0.0	0.0	-0.6	0.1	0.1	0.2	0.2	1.7	0.3	0.4
DoE(Gaussian)	0.2	0.5	0.3	0.6	0.4	0.4	0.4	0.7	1.0	1.0	1.0	0.4	0.7	7.8	1.0	0.6	0.9	1.3		0.4	0.7	1.0	1.0	0.5	0.6	0.6	0.6	0.7	0.8	6.7	7.9	1.8	2.5	0.6	4.2	1.2	1.6	0.1	0.4	
DoE(Logistic)	0.1	0.4	0.2	0.4	0.4	0.4	0.4	0.6	0.9	0.9	1.0	0.3	0.7	7.8	1.0	0.6	0.9	1.3		0.4	0.8	1.1	1.0	0.5	0.6	0.6	0.7	0.8	0.8		2.0	0.5	0.8	0.3	1.5	0.6	1.6	0.1	0.4	

Table 4.1: Mean MI estimates over 10 seeds using N = 10k test samples against ground truth (GT). Color indicates relative negative (red) and positive bias (blue). All methods were trained with 100k samples. List of abbreviations (*Mn*: Multinormal, *St*: Student-t, *Nm*: Normal, *Hc*: Half-cube, *Sp*: Spiral)

Prompt influence of conditional sampling. Generative diffusion models can be interpreted as *iterative* schemes in which starting from pure noise, at each iteration, refinements are applied until a sample from the data distribution is obtained. In recent work on **text conditional image generation** (image $\mathbf{X}$, text prompt $\mathbf{Y}$) by Balaji et al. (2022), it has been observed that the role of the text prompt throughout the generative process has not constant importance . Indeed: *"At the early sampling stage, when the input data to the denoising network is closer to the random noise, the diffusion model mainly relies on the text prompt to guide the sampling process. As the generation continues, the model gradually shifts towards visual features to denoise images, mostly ignoring the input text prompt"* (Balaji et al., 2022). Such claim has been motivated by carefully engineered metric analysis such as self and cross attention maps between images and text, as a function of the generation time, as well as visual inspection of the change in generated images when switching the prompt at different stages of the refinement.

Using MINDE, we can refine heuristic-based methods and produce a similar analysis using theoretically sound information theoretic quantities. In particular, we analyze the conditional mutual information $\mathcal{I}(\mathbf{X}, \mathbf{Y} \mid \mathbf{X}_\tau)$, being $\mathbf{X}_\tau$ the result of the generation process at time τ (recall that the time runs backward from T to 0 during generationSuch metric quantifies, given an observation of the generation process at time τ, how much information the prompt $\mathbf{Y}$ carries about the final generated image $\mathbf{X}$. Clearly, when $\tau = T$, the initial sample is independent from both $\mathbf{X}$ and $\mathbf{X}$. Consequently, the conditional mutual information will coincide with $\mathcal{I}(\mathbf{X}, \mathbf{Y})$.

More formally, we consider the following quantity:

$$\begin{aligned}\mathcal{I}(\mathbf{X},\mathbf{Y}\,|\,\mathbf{X}_\tau) &= \mathcal{I}(\mathbf{X},\mathbf{Y}) - [\mathcal{I}(\mathbf{X}_\tau,\mathbf{Y}) - \mathcal{I}(\mathbf{X}_\tau,\mathbf{X}\,|\,\mathbf{Y})]\,, \qquad (4.29)\\ &= \mathcal{I}(\mathbf{X},\mathbf{Y}) - [\mathcal{H}(\mathbf{X}_\tau) - \mathcal{H}(\mathbf{X}_\tau|\mathbf{Y}) - \mathcal{H}(\mathbf{X}_\tau|\mathbf{Y}) + \mathcal{H}(\mathbf{X}_\tau|\mathbf{X},\mathbf{Y})]\\ &= \mathcal{I}(\mathbf{X},\mathbf{Y}) - \mathcal{I}(\mathbf{X}_\tau,\mathbf{Y})\end{aligned}$$

where Eq. (4.29) is simplified due to the Markov chain $\mathbf{X} - \mathbf{X}_0 - \mathbf{X}_\tau$, so $\mathcal{H}(\mathbf{X}_\tau|\mathbf{X},\mathbf{Y}) = \mathcal{H}(\mathbf{X}_\tau|\mathbf{X}_0,\mathbf{Y}) = \mathcal{H}(\mathbf{X}_\tau|\mathbf{Y})$. Next, we use our MINDE estimator, whereby the marginal and conditional entropies can be estimated efficiently. The following approximation of the quantity in interest can be derived:

$$\begin{aligned}\mathcal{I}(\mathbf{X},\mathbf{Y}\,|\,\mathbf{X}_\tau) &\simeq \int_0^T \mathbb{E}_{\substack{\mathbf{y}_0\sim p^{\mathbf{Y}}\\ \mathbf{x}\sim p_t^{\mathbf{X}|\mathbf{Y}=\mathbf{y}_0}}}\left[\frac{g^2(t)}{2}\left\|s^{\mathbf{X}|\mathbf{Y}}(\mathbf{x},\mathbf{y}_0,t) - s^{\mathbf{X}}(\mathbf{x},t)\right\|^2\right]\mathrm{d}t\\ &\quad - \int_\tau^T \mathbb{E}_{\substack{\mathbf{y}_0\sim p^{\mathbf{Y}}\\ \mathbf{x}\sim p_t^{\mathbf{X}|\mathbf{Y}=\mathbf{y}_0}}}\left[\frac{g^2(t)}{2}\left\|s^{\mathbf{X}|\mathbf{Y}}(\mathbf{x},\mathbf{y}_0,t) - s^{\mathbf{X}}(\mathbf{x},t)\right\|^2\right]\mathrm{d}t\\ &\simeq \int_0^\tau \mathbb{E}_{\substack{\mathbf{y}_0\sim p^{\mathbf{Y}}\\ \mathbf{x}\sim p_t^{\mathbf{X}|\mathbf{Y}=\mathbf{y}_0}}}\left[\frac{g^2(t)}{2}\left\|s^{\mathbf{X}|\mathbf{Y}}(\mathbf{x},\mathbf{y}_0,t) - s^{\mathbf{X}}(\mathbf{x},t)\right\|^2\right]\mathrm{d}t \qquad (4.30)\end{aligned}$$

In our experiments, we also include a MINDE-(σ) version which can be obtained similarly to Eq. (4.30).

Experimental setting. We perform our experimental analysis of the influence of a prompt on image generation using Stable Diffusion (Rombach et al., 2022), using the original codebase and pre-trained checkpoints.[1] The original Stable Diffusion model was trained using the DDPM framework (Ho, Jain, and Abbeel, 2020) on images latent space. This framework is equivalent to the discrete-time version of VPSDE (Song et al., 2021). Using the text prompt samples from LAION dataset (Schuhmann et al., 2022), we synthetically generate image samples. We set guidance mechanism to 1.0 to ensure that the images only contain text conditional content. We use 1000 samples and approximate the integral using a Simpson integrator [2] with a discretization over 1000 timesteps.

[1] https://huggingface.co/stabilityai/stable-diffusion-2-1

[2] https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.simpson.html

Results. We report in Figure 4.3 values of $\mathcal{I}(\mathbf{X}, \mathbf{Y} \mid \mathbf{X}_\tau)$ as a function of (reverse) diffusion time, where $\mathbf{X}$ is in the image domain and $\mathbf{Y}$ is in the text domain. In a similar vein to what observed by Balaji et al. (2022), our results indicate that $\mathcal{I}(\mathbf{X}, \mathbf{Y} \mid \mathbf{X}_\tau)$ is very high when $\tau \simeq T$, which indicates that the text prompt has maximal influence during the early stage of image generation. This measurement is relatively stable at high MI values until $\tau \approx 0.8$. Then, the influence of the prompt gradually fades, as indicated by decreasing steadily MI values. This corroborates the idea that mutual information can be adopted as an exploratory tool for the analysis of complex, high dimensional distributions in real use cases.

The intuition pointed out by our MINDE estimator is further consolidated by the qualitative samples in Figure 4.4, where we perform the following experiment: we test whether switching from an original prompt to a different prompt during the backward diffusion semantically impacts the final generated images. We observe that changing the prompt before $\tau \simeq 0.8$ results almost surely with semantically coherent generated image with the second prompt. Instead, when $\tau < 0.8$, the second prompt influence diminishes gradually. We observe that for all the qualitative samples shown in Figure 4.4 the second prompt has no influence on the generated image after $\tau < 0.7$.

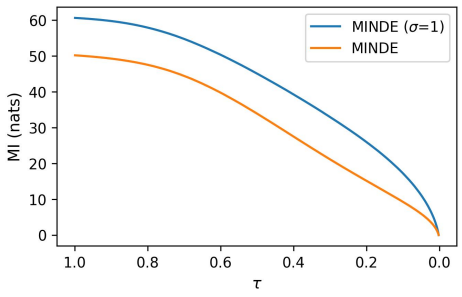


Figure 4.3: $I(\mathbf{X}, \mathbf{Y} \mid \mathbf{X}_\tau)$ as a function of τ.

4.7 Conclusion

The estimation of MI stands as a fundamental goal in many areas of machine learning, as it enables understanding the relationships within data, driving representation learning, and evaluating generative models. Over the years, various methodologies have emerged to tackle the difficult task of MI estimation, addressing challenges posed by high-dimensional, real-world data. Our work introduced a novel method, MINDE, which provides a unique perspective on MI estimation by leveraging the theory of diffusion-based generative models. We expanded the classical toolkit for information-theoretic analysis, and showed how to compute the KL divergence and entropy of random variables using the score of data

Figure 4.4: To validate the explanatory results obtained via the application of our MINDE estimator, we perform the following experiment: Conditional generation is carried out with *Prompt 1* until time τ, whereas after the conditioning signal is switched to *Prompt 2*. We use the same Stable diffusion model as in the previous experiment with guidance scale set to 9.

distributions. We defined several variants of MINDE, which we have extensively tested according to a recent, comprehensive benchmark that simulates real-world challenges, including sparsity, long-tailed distributions, invariance to transformations. Our results indicated that our methods outperform state-of-the-art alternatives, especially on the most challenging tests. Additionally, MINDE variants successfully passed self-consistency tests, validating the robustness and reliability of our proposed methodology.

Our research opens up exciting avenues for future exploration. One compelling direction is the application of MINDE to large-scale multi-modal datasets. The conditional version of our approach enables harnessing the extensive repository of existing pre-trained diffusion models. For instance, it could find valuable application in the estimation of MI for text-conditional image generation. Conversely, our joint modeling approach offers a straightforward path to scaling MI estimation to more than two variables. A scalable approach to MI estimation is particularly valuable when dealing with complex systems involving multiple interacting variables.

Chapter 5

Multi-variate Information Estimation

The analysis of scientific data and complex multivariate systems requires information quantities that capture relationships among multiple random variables. Recently, new information-theoretic measures have been developed to overcome the shortcomings of classical ones, such as mutual information, that are restricted to considering pairwise interactions. Among them, the concept of information synergy and redundancy is crucial for understanding the high-order dependencies between variables. One of the most prominent and versatile measures based on this concept is O-information, which provides a clear and scalable way to quantify the synergy-redundancy balance in multivariate systems. However, its practical application is limited to simplified cases. In this work, we introduce SΩI, which allows to compute O-information without restrictive assumptions about the system while leveraging a unique model. Our experiments validate our approach on synthetic data, and demonstrate the effectiveness of SΩI in the context of a real-world use case.

5.1 Introduction

MI is a fundamental measure which allows investigation of the non-linear dependence between random variables (Shannon, 1948; MacKay, 2003). Despite its success in various domains, classical MI suffers from limitations when analyzing systems composed by more than two variables. This constitutes an important limitation, considering that many scientific endeavors aim at an accurate statistical characterization of systems which are composed of many random variables. Examples includes neuroscience (Latham and Nirenberg, 2005; Ganmor, Segev, and Schneidman, 2011; Gat and Tishby, 1998), climate models (Runge et al., 2019), econometrics (Dosi and Roventini, 2019), and machine learning (Tax, Mediano, and Shanahan, 2017), to name a few.

A recent attempt to overcome such limitations, and to extend the applicability of information-theoretic tools to multivariate systems, is represented by Partial Information Decomposition (PID) (Williams and Beer, 2010). The key idea behind such method is the *decomposition* of the overall MI between a set of source variables and a given target variable into non-negative constituents. In particular, PID quantifies how much of the total information about the target variable is encoded redundantly, synergistically or uniquely into given subsets of variables. *Redundancy* quantifies information that is shared between subsets of the partition, *synergy* describes the additional information that is endowed to all subsets observed jointly but that is not available from individual constituents of the partition, and *uniqueness* quantifies the information that is lost when a given subset is not observed, removing the amount of redundant and synergistic information associated to that subset. The PID method requires partitioning the source system into all its possible subsets and computing the information decomposition of all constituents with respect to the target variable.

Despite its elegance, this measure is not without drawbacks. Indeed, there is no consensus on the best way to define and compute PID, and several variants have emerged, including (Barrett, 2014), who reformulate synergy and redundancy for Gaussian systems (but that has been judged as poorly motivated by (Venkatesh et al., 2023)), (Finn and Lizier, 2020), who use the algebraic structure of information sharing, (Ay, Polani, and Virgo, 2019), who rely on cooperative game theory, (Rosas et al., 2020), who build on concepts related to data privacy and disclosure, (Kolchinsky, 2019), who use set theory, (Enk, 2023), who deal with scalability issues by pooling probabilities, (Gutknecht, Makkeh, and Wibral, 2023), who use a mereological formulation, and (Makkeh, Gutknecht, and Wibral, 2021; Ehrlich et al., 2023), who advocate for methods based on the exclusions of probability mass. Nevertheless, the main limitations of PID persist in all variants. Indeed, computational complexity grows extremely fast, precisely as the Dedekind number of variables (which is more than 10^{31} for 9 variables). Moreover, PID computation relies on a partition of the system into a set of sources and a unique target. This can be an artificial distinction which limits usability and interpretability of the results. This latter problem is partially addressed in (Varley et al., 2023), who introduce Partial Entropy Decomposition (PED).

Motivated by these limitations, (Rosas et al., 2019) introduce the concept of O-information, a measure which captures the synergy-redundancy dominance in multivariate systems. In contrast to PID, this measure does not require the system to be partitioned into sources and a target, and gracefully scales in the number of its components (Martinez Mediano, 2022). Furthermore, recent extensions such as O-information locality (Scagliarini et al., 2021) and gradient computation (Scagliarini et al., 2023) allow a fine-grained analysis of system

behavior. However, O-information measures are accessible only in restricted scenarios. Indeed, existing methods rely on estimation techniques that requires either i) discrete distributions (or binning of continuous ones) or ii) Gaussian distributions. In this work, we show that such limitations can be lifted by using and extending recent methods to estimate MI (Franzese, Bounoua, and Michiardi, 2024; Kong et al., 2024).

Our work is organized as follows: § 5.2 introduces the high-dimensional interaction measures which we investigate in this work, while § 5.3 proposes Score-based O-Information estimation (SΩI), our novel methodology which allows scalable and flexible O-information estimation. § 5.4 validates experimentally our proposed method, where we report a series of compelling results on various synthetic systems, for which ground truth values are known and accessible analytically. Furthermore, we consider a realistic endeavor by revisiting previous studies (Venkatesh et al., 2023) that focus on the analysis of brain activity in mice. Our method allows lifting previous limiting assumptions, and allow synergy-redundancy characterizations that are compatible with observations made by domain experts. Finally, we summarize our findings in § 5.5.

5.2 High dimensional interaction measures

Consider the continuous **multivariate** random variable $\mathbf{X} = \{\mathbf{X}^1, \dots, \mathbf{X}^N\} \sim p(\mathbf{x}^1, \dots, \mathbf{x}^N)$. We indicate the collection of all but the i_{th} random variable with the symbol $\mathbf{X}^{\setminus i} \stackrel{\text{def}}{=} \{\mathbf{X}^1, .., \mathbf{X}^{i-1}, \mathbf{X}^{i+1}, .., \mathbf{X}^N\}$. When necessary, we indicate marginal and conditional distributions by properly specifying the arguments of the distribution, e.g. $\mathbf{X}^i \sim p(\mathbf{x}^i)$ or $\mathbf{X}^{\setminus i} \,|\, \mathbf{X}^i \sim p(\mathbf{x}^1, \dots, \mathbf{x}^{i-1}, \mathbf{x}^{i+1}, \dots, \mathbf{x}^M \,|\, \mathbf{x}^i)$.

A central quantity in this work is the Shannon entropy associated to a given random variable $\mathcal{H}(\mathbf{X}) \stackrel{\text{def}}{=} \mathbb{E}\left[-\log p(\mathbf{X})\right]$ (Cover, Thomas, et al., 1991). Considering the case of bi-variate (i.e. $N = 2$) random variable X, entropy and conditional entropy allow computation of the mutual information (MI) flow $\mathcal{I}$ between the two random variables $\mathbf{X}^1, \mathbf{X}^2$: $\mathcal{I}(\mathbf{X}^1; \mathbf{X}^2) = \mathcal{H}(\mathbf{X}^1) - \mathcal{H}(\mathbf{X}^1|\mathbf{X}^2)$, where $\mathcal{H}(\mathbf{X}^1|\mathbf{X}^2) = \mathbb{E}\left[-\log p(\mathbf{X}^1 \,|\, \mathbf{X}^2)\right]$. Importantly, such quantity can also be expressed as the KL divergence (Cover, Thomas, et al., 1991) between the joint and the product of marginal distributions: $\mathcal{I}(\mathbf{X}^1; \mathbf{X}^2) = \mathbb{KL}\left(p(\mathbf{x}^1, \mathbf{x}^2) \parallel p(\mathbf{x}^1)p(\mathbf{x}^2)\right)$. For the case of $N = 3$, it is possible to define the MI as $\mathcal{I}(\mathbf{X}^1; \mathbf{X}^2; \mathbf{X}^3) = \mathcal{I}(\mathbf{X}^1; \mathbf{X}^2) - \mathcal{I}(\mathbf{X}^1; \mathbf{X}^2|\mathbf{X}^3)$, where $\mathcal{I}(\mathbf{X}^1; \mathbf{X}^2|\mathbf{X}^3) = \mathcal{H}(\mathbf{X}^1 \,|\, \mathbf{X}^3) - \mathcal{H}(\mathbf{X}^1 \,|\, \mathbf{X}^2, \mathbf{X}^3)$. This quantity, also known as co-information or interaction information, can counter-intuitively result in a negative value, and measures the difference between synergistic and redundant interactions (Rosas et al., 2019).

Since, for $N > 3$, interaction information becomes difficult to grasp (Williams and Beer, 2010; Rosas et al., 2019), our goal in this work is to consider extensions to MI, while preserving interpretability. In particular, a measure of the interaction strengths in a system with $N > 3$ can be obtained by studying the summand mutual information between one variable and the rest of the system:

$$\mathcal{S}(\mathbf{X}) \stackrel{\text{def}}{=} \sum_{i=1}^{N} \mathcal{I}(\mathbf{X}^i; \mathbf{X}^{\setminus i}). \tag{5.1}$$

This quantity, named S-information, can be decomposed into the redundant and synergistic components of the considered multivariate system. In particular, since $\mathbf{X}^{\setminus i} = \{\mathbf{X}^{<i}, \mathbf{X}^{>i}\}$, where $\mathbf{X}^{<i} = \{\mathbf{X}^1, \ldots, \mathbf{X}^{i-1}\}$ and $\mathbf{X}^{>i} = \{\mathbf{X}^{i+1}, \ldots, \mathbf{X}^N\}$ (with $\mathbf{X}^{>N} = \emptyset$), we can use the conditional mutual information laws (Cover, Thomas, et al., 1991) and rewrite $\mathcal{S}(\mathbf{X})$ as:

$$\mathcal{S}(\mathbf{X}) = \sum_{i=1}^{N} \mathcal{I}(\mathbf{X}^i; \mathbf{X}^{>i}) + \sum_{i=1}^{N} \mathcal{I}(\mathbf{X}^i; \mathbf{X}^{<i} \,|\, \mathbf{X}^{>i}). \tag{5.2}$$

The two **positive** series which constitute $\mathcal{S}(\mathbf{X})$ are equivalent to the Total Correlation (TC) (Sun, 1975) and the Dual Total Correlation (DTC) (Sun Han, 1980) denoted by $\mathcal{T}(.)$ and $\mathcal{D}(.)$ respectively. Then, $\mathcal{S}(\mathbf{X}) = \mathcal{T}(\mathbf{X}) + \mathcal{D}(\mathbf{X})$, where (proof in Appendix C.1)

$$\mathcal{T}(\mathbf{X}) = \sum_{i=1}^{N} \mathcal{H}(\mathbf{X}^i) - \mathcal{H}(\mathbf{X}), \tag{5.3}$$

$$\mathcal{D}(\mathbf{X}) = \mathcal{H}(\mathbf{X}) - \sum_{i=1}^{N} \mathcal{H}(\mathbf{X}^i | \mathbf{X}^{\setminus i}). \tag{5.4}$$

TC is high in cases where, for each variable $\mathbf{X}^i$, at least one of its "*children*" (variables in $\mathbf{X}^{>i}$) carries information about it. Importantly, the number of children conveying information (whether 1, 2, or $N-1$) is irrelevant. Since $\mathcal{T}(\mathbf{X})$ is permutation invariant, a high value implies that for every ordering of the variables, and hence for all possible combinations of children of a given variable, the summand mutual information between variables and their children remains high. This intuition, which suggests *redundancy*, can similarly be obtained by considering the entropic formulation. Indeed, whenever a system is composed of perfectly independent variables ($\mathbf{X}^i \perp \mathbf{X}^j, i \neq j$) $\mathcal{H}(\mathbf{X}) = \sum_{i=1}^{N} \mathcal{H}(\mathbf{X}^i)$ and consequently $\mathcal{T}(\mathbf{X}) = 0$. On the other hand, a *copy* system ($\mathbf{X}^i = \mathbf{X}^j, \forall i, j$) achieves infinite $\mathcal{T}(\mathbf{X})$, as $\mathcal{H}(\mathbf{X}) = -\infty$, since the support of the joint distribution is on a lower than $N-$dimensional space. TC also admits a representation in terms of KL divergences, $\mathcal{T}(\mathbf{X}) = \mathbb{KL}\left(p(\mathbf{x}) \parallel \prod_{i=1}^{N} p(\mathbf{x}^i)\right)$, which we will exploit later in our proposed methodology.

Similar considerations can be carried out for the DTC. Consider a single MI term $I(\mathbf{X}^i; \mathbf{X}^{<i} | \mathbf{X}^{>i})$. The focus of this conditioning is about quantifying how much **additional** information the variables $\mathbf{X}^{<i}$ carry about $\mathbf{X}^i$ if we are also given access $\mathbf{X}^{>i}$. Whenever the variables are independent or redundant (the copy system), this value is identically zero. However, whenever the aid of the *extra* measurements unlocks new bits of information, which suggests a *synergistic* scenario, its value is positive.

Having recognized that $\mathcal{S}(\mathbf{X})$ in a multivariate system can be decomposed into measures of redundancy $\mathcal{T}(\mathbf{X})$ and synergy $\mathcal{D}(\mathbf{X})$, we can introduce a new information theoretic measure which quantifies the difference between the two behaviours. This quantity, named O-information (Rosas et al., 2019), is defined as

$$\Omega(\mathbf{X}) = \mathcal{T}(\mathbf{X}) - \mathcal{D}(\mathbf{X}). \tag{5.5}$$

In summary, while S-information only quantifies the strength of interactions in a system, O-information also determines the *nature* of these interactions, being them redundant or synergistic. Intuitively, a redundancy-dominated system is the most parsimonious explanation — in an Occam's razor sense — whenever $\Omega(\mathbf{X}) > 0$. Conversely, a negative value $\Omega(\mathbf{X}) < 0$ is associated with a synergy-dominated system. O-information is a natural generalization of MI for more than 3 variables: indeed, it is equal to the co-information for $N = 3$, and is a measure which preserves interpretability for any positive N.

One important property of O-information is that it gracefully scales with the number of random variables composing a system, as opposed to, e.g. the PID measure, which has much worse scalability.

Since O-information measures the *overall* information dynamics among variables, recent work focus on ways to study the *individual* influence of variables to the high-order interactions, and capture the interaction structure of a multivariate system (Scagliarini et al., 2023). The first order difference, called the *gradient* of O-information, captures how much O-information changes when adding or removing a given system variable i:

$$\partial_i \Omega(\mathbf{X}) = \Omega(\mathbf{X}) - \Omega(\mathbf{X}^{\setminus i}). \tag{5.6}$$

A positive value implies that $\mathbf{X}^i$ provides redundant information to the system, while a negative one suggests that its interaction with other variables is mainly synergistic.

5.3 Score-based O-information estimation

O-information and its gradient represent extremely useful information theoretic measures to study multivariate systems. However, as it is clear from Eqs. (5.3) to (5.5), their estimation requires access to entropies, conditional entropies and KL divergence measures. When strict assumptions about the distribution of variables composing the system are possible, such as discrete or Gaussian distributions, existing implementations of O-information estimators have been used successfully in a number of application domains (Varley et al., 2022; Sparacino et al., 2023; Stramaglia et al., 2021; Chiarion et al., 2023). However, in more realistic cases where such assumptions are not valid, there currently does not exist a method to estimate the constituents of O-information in a reliable and scalable manner. In this work, we present the first methodology allowing estimation of O-information for more general scenarios. Our method unfolds according to the observation that all quantities of interest can be expressed in terms of KL divergences, and relies on a technique to estimate such divergences which scales gracefully with the system size. Our key ingredient is the score function associated to data distributions (Vincent, 2011; Song and Ermon, 2019) and the method we present leverages recent advances in the field of MI estimation (Franzese, Bounoua, and Michiardi, 2024; Kong et al., 2024).

5.3.1 Score-based divergence estimation

Consider the generic multivariate random variable $\mathbf{X}$ with associated distribution $p(\mathbf{x})$. Provided that certain minimal regularity assumptions are met (Vincent, 2011), it is always possible to associate the distribution $p(\mathbf{x})$ to its *score function*, defined as the gradient of its logarithm, $\nabla \log p(\mathbf{x})$.

Recently, the community has showed tremendous interest (Song and Ermon, 2019; Song et al., 2021) in a generalization of such concept, which involves computing the score function of a *noised* version of the variable $\mathbf{X}$, due to the possibility of adopting such concept for generative modeling purposes. Accordingly, in this work we define a noised version of the variable $\mathbf{X}$ with corresponding intensity indexed by $t \in [0, \infty)$. Then, the new variable is constructed as $\mathbf{X}_t = \mathbf{X} + \sqrt{2t}W$, where W is a Gaussian random vector with the same dimension of $\mathbf{X}$, zero mean and identity covariance matrix.

This new random variable can be associated to its *time-varying* score function $\nabla \log p_t(\mathbf{x})$. In particular the analytic expression of $p_t(\mathbf{x})$ can be obtained as the solution of the Partial Differential Equation (PDE) $\frac{\mathrm{d}p_t(\mathbf{x})}{\mathrm{d}t} = \Delta p_t(\mathbf{x})$, with initial conditions given by $p_0(\mathbf{x}) = p(\mathbf{x})$.

Next, we consider the KL divergence between two generic distributions and define how it can be computed using score functions, a result which we will use later for computing O-information.

Proposition 2. *(Franzese, Bounoua, and Michiardi, 2024; Kong et al., 2024) The KL divergence between two generic distributions $p(\mathbf{x})$ and $q(\mathbf{x})$, defined as*

$$\mathbb{KL}\left(p(\mathbf{x}) \parallel q(\mathbf{x})\right) = \int p(\mathbf{x}) \log \frac{p(\mathbf{x})}{q(\mathbf{x})} \mathrm{d}\mathbf{x},$$

can be computed considering the time-varying score functions $\nabla \log(p_t)$ and $\nabla \log(q_t)$, according to the following expression:

$$\mathbb{KL}\left(p(\mathbf{x}) \parallel q(\mathbf{x})\right) = \int p_t(\mathbf{x}) \left\| \nabla \log \left(\frac{p_t(\mathbf{x})}{q_t(\mathbf{x})} \right) \right\|^2 \mathrm{d}x\mathrm{d}t.$$

Proof sketch. To avoid clutter, we drop the dependence on $\mathbf{x}$ of the distributions. Let's define $r_t \stackrel{\text{def}}{=} \int p_t \log \frac{p_t}{q_t} \mathrm{d}\mathbf{x}$.

Since it holds that $r_\infty - \mathbb{KL}\left(p \parallel q\right) = \int_0^\infty \frac{\mathrm{d}r_t}{\mathrm{d}t} \mathrm{d}t$, we need

$$\int \frac{\mathrm{d}r_t}{\mathrm{d}t} \mathrm{d}t = \int \frac{\mathrm{d}p_t}{\mathrm{d}t} \log \left(\frac{p_t}{q_t} \right) + p_t \frac{\mathrm{d}}{\mathrm{d}t} \log \left(\frac{p_t}{q_t} \right) \mathrm{d}\mathbf{x}\mathrm{d}t.$$

Note that $\int p_t \frac{\mathrm{d}}{\mathrm{d}t} \log \left(\frac{p_t}{q_t} \right) \mathrm{d}\mathbf{x}\mathrm{d}t = \int \frac{\mathrm{d}}{\mathrm{d}t} p_t - \frac{p_t}{q_t} \Delta q_t \mathrm{d}\mathbf{x}\mathrm{d}t$, and $\int \frac{\mathrm{d}}{\mathrm{d}t} p_t \mathrm{d}\mathbf{x}\mathrm{d}t = 0$ (See Appendix C.1.1 for detailed proof). Then, the expression above can be rewritten as $\int p_t \Delta \log \left(\frac{p_t}{q_t} \right) - \frac{p_t}{q_t} \Delta q_t \mathrm{d}\mathbf{x}\mathrm{d}t$. Integrating by parts we obtain $\int -\nabla p_t \nabla \log \left(\frac{p_t}{q_t} \right) + \nabla (\frac{p_t}{q_t}) \nabla q_t \mathrm{d}\mathbf{x}\mathrm{d}t$. Since $\nabla p_t = p_t \nabla \log p_t$ and $\nabla (\frac{p_t}{q_t}) \nabla q_t = p_t \nabla \log q_t \nabla \log \left(\frac{p_t}{q_t} \right)$, and $r_\infty = 0$ (Franzese et al., 2023; Villani, 2009; Collet and Malrieu, 2008), the proposition follows. □

The result in Proposition 2 allows, in principle, the exact computation of KL divergences, provided knowledge of the score functions $\nabla \log p_t, \nabla \log q_t$. Such knowledge is however out of reach in practical cases, which is why in this work we consider a *parametric* approximation of such vector fields, leading to a KL divergence *estimator*. In particular, we leverage the methodology considered in (Song and Ermon, 2019; Song et al., 2021) where the parametric score s_t is obtained by minimizing the so called *denoising score-matching* loss

$$\int p(\mathbf{x}) p_{0t}(\tilde{\mathbf{x}} \,|\, \mathbf{x}) \| s_t(\tilde{\mathbf{x}}) - \nabla \log (p_{0t}(\tilde{\mathbf{x}} \,|\, \mathbf{x})) \|^2 \mathrm{d}\mathbf{x}\mathrm{d}\tilde{\mathbf{x}}\mathrm{d}t,$$

where $p_{0t}(\tilde{\mathbf{x}} \,|\, \mathbf{x})$ is the conditional distribution of the noised random variable *given* initial conditions $\mathbf{X} = \mathbf{x}$, i.e. $p_t(\tilde{\mathbf{x}}) = \int p_{0t}(\tilde{\mathbf{x}} \,|\, \mathbf{x}) p(\mathbf{x}) \mathrm{d}\mathbf{x}$. Note that p_{0t} has known Gaussian

distribution with mean $\mathbf{x}$ and variance $2t$. This allows, together with the knowledge of the score functions, the implementation of an estimator for the KL divergence.

Informally, learning the score can be understood as learning to *denoise* the variable $\mathbf{X}_t$ to obtain $\mathbf{X}$. Indeed, the score functions have analytic expression $\nabla \log p_t(\mathbf{x}) = \frac{\mathbb{E}[\mathbf{X} \mid \mathbf{X}_t = \mathbf{x}] - \mathbf{x}}{2t}$, where the only unknown is $\mathbb{E}[\mathbf{X} \mid \mathbf{X}_t = \mathbf{x}]$. An alternative, but equivalent parametrization of the problem, consists in estimating the noise W, given $\mathbf{X}_t$. We use this approach in our work since is considered to be more stable numerically (Ho, Jain, and Abbeel, 2020). In practice, the VP-SDE (Song et al., 2021) framework is adopted as the noising process. With such a schedule varying between $[0, T]$, it's valid to assume that $\mathbf{X}_T$ is practically indistinguishable from pure noise (More details in Appendix C.2).

5.3.2 Estimating O-information

Armed with Proposition 2, we can leverage score functions to estimate the information-theoretic quantities introduced in § 5.2. Here we consider an extension of the simple noising process described in § 5.3.1, where we allow i) noising of only certain subsets of the variables or ii) deletion of subset of variables. In practice, the first case corresponds to learning to denoise a portion of the variables, given auxiliary information about the other (noiseless) variables, e.g. to learn $\mathbb{E}[\mathbf{X}^i \mid \mathbf{X}_t^i = \tilde{\mathbf{x}}^i, \mathbf{X}^{\setminus i} = \mathbf{x}^{\setminus i}]$. Instead, the second case amounts to denoising problems akin to $\mathbb{E}[\mathbf{X}^i \mid \mathbf{X}_t^i = \tilde{\mathbf{x}}^i]$. In our implementation, we follow the approach proposed in (Bounoua, Franzese, and Michiardi, 2024) (See Appendix C.2). Next, we use such an intuition to derive a series of propositions that pave the way to O-information computation.

In what follows, we use the compact notation $\left[(\cdot)^i\right]_{i=1}^N$, to indicate a concatenation of N elements in a column vector.

Proposition 3. *Given a multivariate random variable $\mathbf{X} = \{\mathbf{X}^1, \dots, \mathbf{X}^N\} \sim p(\mathbf{x}^1, \dots, \mathbf{x}^N)$, and its corresponding noised version, the Total Correlation $\mathcal{T}(\mathbf{X})$ is equal to:*

$$\int \frac{1}{4t^2} \mathbb{E} \left\| \mathbb{E}[\mathbf{X} \mid \mathbf{X}_t] - \left[\mathbb{E}[\mathbf{X}^i \mid \mathbf{X}_t^i]\right]_{i=1}^N \right\|^2 \mathrm{d}t.$$

Proof Sketch. Recall that $\mathcal{T}(\mathbf{X}) = \mathbb{KL}\left(p(\mathbf{x}) \parallel \prod_{i=1}^N p(\mathbf{x}^i)\right)$. Then, by virtue of Proposition 2, we have that $\mathcal{T}(\mathbf{X})$ equals

$$\int p_t(\mathbf{x}) \left\| \nabla \log p_t(\mathbf{x}) - \left[\tfrac{\partial}{\partial \mathbf{x}^i} \log p_t(\mathbf{x}^i)\right]_{i=1}^N \right\|^2 \mathrm{d}\mathbf{x}\mathrm{d}t.$$

The terms $\frac{\partial}{\partial \mathbf{x}^i} \log p_t(\mathbf{x}^i)$ correspond to $1/2t(\mathbb{E}[\mathbf{X}^i \mid \mathbf{X}_t^i = \mathbf{x}^i] - \mathbf{x}^i)$. Then, the proposition follows. □

Proposition 4. *Given a multivariate random variable* $\mathbf{X} = \{\mathbf{X}^1, \dots, \mathbf{X}^N\} \sim p(\mathbf{x}^1, \dots, \mathbf{x}^N)$, *and its corresponding noised version, the S-information* $\mathcal{S}(\mathbf{X})$ *is equal to:*

$$\int \frac{1}{4t^2} \mathbb{E} \left\| \left[\mathbb{E}[\mathbf{X}^i \mid \mathbf{X}_t^i]\right]_{i=1}^N - \left[\mathbb{E}[\mathbf{X}^i \mid \mathbf{X}_t^i, \mathbf{X}^{\setminus i}]\right]_{i=1}^N \right\|^2 \mathrm{d}t.$$

Proof Sketch. In light of Eq. (5.1), it holds that

$$\mathcal{S}(\mathbf{X}) = \sum_{i=1}^N \int p(\mathbf{x}^{\setminus i}) \mathbb{KL}\left(p(\mathbf{x}^i \mid \mathbf{x}^{\setminus i}) \parallel p(\mathbf{x}^i)\right) \mathrm{d}\mathbf{x}^{\setminus i},$$

where the i_{th} KL term of the sum is equal to (Proposition 2)

$$\int p(\mathbf{x}^i \mid \mathbf{x}^{\setminus i}) p_{0t}(\tilde{\mathbf{x}}^i \mid \mathbf{x}^i) \left\| \frac{\partial}{\partial \tilde{\mathbf{x}}^i} \log \left(\frac{p_t(\tilde{\mathbf{x}}^i)}{\hat{p}_{0t}(\tilde{\mathbf{x}}^i \mid \mathbf{x}^{\setminus i})} \right) \right\|^2 \mathrm{d}\tilde{\mathbf{x}}^i \mathrm{d}\mathbf{x}^i \mathrm{d}t.$$

Now, we can move the terms $p(\mathbf{x}^{\setminus i})$ inside the KL computation integrals and write the sum of the norms as the norm of a vector, which allows computing S-information as

$$\mathcal{S}(\mathbf{X}) = \int p(\mathbf{x}) p_{0t}(\tilde{\mathbf{x}} \mid \mathbf{x}) \left\| \left[\tfrac{\partial}{\partial \tilde{\mathbf{x}}^i} \log p_t(\tilde{\mathbf{x}}^i)\right]_{i=1}^N - \left[\tfrac{\partial}{\partial \tilde{\mathbf{x}}^i} \log p_t(\tilde{\mathbf{x}}^i \mid x^{\setminus i})\right]_{i=1}^N \right\|^2 \mathrm{d}\tilde{\mathbf{x}} \mathrm{d}\mathbf{x} \mathrm{d}t,$$

where $p_t(\tilde{\mathbf{x}}^i \mid \mathbf{x}^{\setminus i}) = \int p_{0t}(\tilde{\mathbf{x}}^i \mid \mathbf{x}^i) p(\mathbf{x}^i \mid \mathbf{x}^{\setminus i}) \mathrm{d}\mathbf{x}^i$.

Finally, the proposition follows since we can interpret the elements inside the square norm in terms of denoisers, with $\frac{\partial}{\partial \tilde{\mathbf{x}}^i} \log p_t(\tilde{\mathbf{x}}^i \mid \mathbf{x}^{\setminus i}) = 1/2t(\mathbb{E}[\mathbf{X}^i \mid \mathbf{X}_t^i = \tilde{\mathbf{x}}^i, \mathbf{X}^{\setminus i} = \mathbf{x}^{\setminus i}] - \tilde{\mathbf{x}}^i)$□

Proposition 5. *Given a multivariate random variable* $\mathbf{X} = \{\mathbf{X}^1, \dots, \mathbf{X}^N\} \sim p(\mathbf{x}^1, \dots, \mathbf{x}^N)$, *and its corresponding noised version, the Dual Total Correlation* $\mathcal{D}(\mathbf{X})$ *equals:*

$$\int \frac{1}{4t^2} \mathbb{E} \left\| \mathbb{E}[\mathbf{X} \mid \mathbf{X}_t] - \left[\mathbb{E}[\mathbf{X}^i \mid \mathbf{X}_t^i, \mathbf{X}^{\setminus i}]\right]_{i=1}^N \right\|^2 \mathrm{d}t$$

Proof Sketch. The starting point to obtain DTC is to recall that $\mathcal{D}(\mathbf{X}) = \mathcal{S}(\mathbf{X}) - \mathcal{T}(\mathbf{X})$. Then, it is sufficient to expand the square norms of $\mathcal{S}(\mathbf{X})$ and $\mathcal{T}(\mathbf{X})$ and combine the different terms, to state that $\mathcal{D}(\mathbf{X})$ equals:

$$\int p(\mathbf{x}) p_{0t}(\tilde{\mathbf{x}} \mid \mathbf{x}) \left\| \nabla \log p_t(\tilde{\mathbf{x}}) - \left[\tfrac{\partial}{\partial \tilde{\mathbf{x}}^i} \log p_t(\tilde{\mathbf{x}}^i \mid \mathbf{x}^{\setminus i})\right]_{i=1}^N \right\|^2 \mathrm{d}\tilde{\mathbf{x}} \mathrm{d}\mathbf{x} \mathrm{d}t.$$

This can be proven considering that :

- $\mathbb{E}\left[\mathbb{E}[\mathbf{X}^i \mid \mathbf{X}_t]\right] \mathbb{E}[\mathbf{X}^i \mid \mathbf{X}_t^i]] = \mathbb{E}[(\mathbb{E}[\mathbf{X}^i \mid \mathbf{X}_t^i])^2]$

- $\mathbb{E}\left[\mathbb{E}[\mathbf{X}^i \mid \mathbf{X}_t^i]\mathbb{E}[\mathbf{X}^i \mid \mathbf{X}_t^i, \mathbf{X}^{\setminus i}]\right] = \mathbb{E}\left[(\mathbb{E}[\mathbf{X}^i \mid \mathbf{X}_t^i])^2\right]$
- $\mathbb{E}\left[\mathbb{E}[\mathbf{X}^i \mid \mathbf{X}_t]]\mathbb{E}[\mathbf{X}^i \mid \mathbf{X}_t^i, \mathbf{X}^{\setminus i}]\right] = \mathbb{E}[(\mathbb{E}[\mathbf{X}^i \mid \mathbf{X}_t^i])^2]$.

Then, the proposition follows. □

Finally, to estimate O-information, it is sufficient to combine Proposition 3 and Proposition 5, and apply Eq. (5.5). In practical terms, our method requires access to *denoisers* for the three following scenarios: i) given $\mathbf{X}_t$ estimate $\mathbf{X}$ ii) given $\mathbf{X}_t^i$ estimate $\mathbf{X}^i$ iii) given $\mathbf{X}_t^i$ and $\mathbf{X}^{\setminus i}$ estimate $\mathbf{X}^i$. To achieve this, we extend the methodology proposed in (Bounoua, Franzese, and Michiardi, 2024), and amortize the three different scenarios with a *unique denoising network*, which takes as input the concatenation of noised and clean variables and outputs the corresponding estimates (see Appendix C.2). Additionally, the estimation of the gradients of O-information requires approximating additional denoising score functions to access Eq. (C.3) (More details in Appendix C.2.2).

5.4 Experimental validation

We evaluate our method according to two strategies. First, we focus on a synthetic setup that allows analytic computation of O-information and full control on system scale. Then, we consider real data collected in a study of brain activity in mice, to demonstrate how SΩI unlocks new avenues in the application of information measures in real systems without the need for restrictive assumptions.

5.4.1 Synthetic benchmark

We consider a canonical Gaussian system, whereby we control the number of variables describing the system N, the dimension of each variable (**Dim**), the inter-dependencies between variables describing how they interact, and the strength of interaction (More details in Appendix C.2). Inspired by (Czyż et al., 2023), we consider more challenging distribution going beyond the Gaussian setting (Please refer to Appendix C.5). No other neural estimator capable of estimating O-information was explored in the literature. Next, we construct an original baseline that relies on neural estimation of MI to access the MI decomposition of O-information.

Baseline. Recent work (Bai et al., 2023) describes a method to compute TC by leveraging a decomposition into pairwise MI terms. Clearly, DTC can also be decomposed into MI terms.

Therefore, we extend (Bai et al., 2023) such that it can be used as a baseline to compute O-information. The main limitation of such a baseline is poor scalability: it requires training an individual model for each MI term in which TC and DTC are decomposed in. We adopt the linear-decomposition method (Bai et al., 2023), which results in $2(N-1)$ MI terms (see Appendix C.3), and propose four variants to estimate MI based on (Belghazi et al., 2018; Nguyen, Wainwright, and Jordan, 2007; Oord, Li, and Vinyals, 2018; Cheng et al., 2020). We label this baseline approach according to the MI estimators: MINE ,NWJ, InfoNCE, and CLUB.

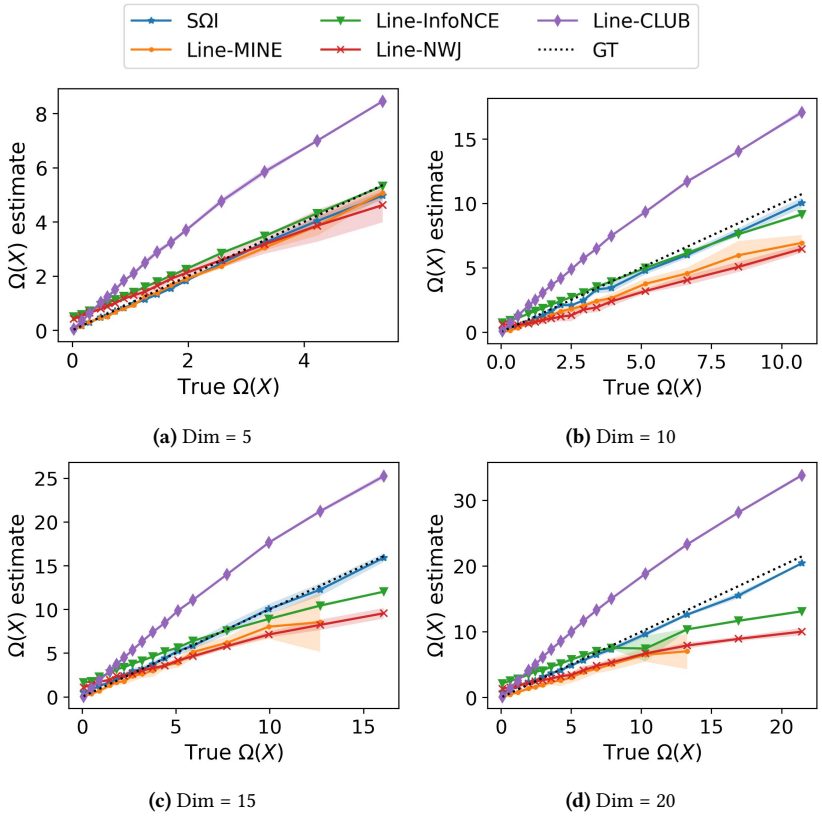
Experimental protocol For each experiment, we use $100k$ samples for training the various neural estimators, and $10k$ samples at inference time, to estimate O-information. For our method SΩI, we use the VP-SDE formulation (Song et al., 2021) and learn a *unique* denoising network to estimate the various score terms. The denoiser is a simple, stacked MLP with skip connections, adapted to the input dimension. We apply importance sampling (Huang, Lim, and Courville, 2021; Song et al., 2021) at both training and inference time. Finally, we use 10-sample Monte Carlo estimates for computing integrals. More details about the implementation are included in Appendix C.3. For the baseline variants, for each MI term we use an MLP that is sufficiently expressive given the data dimension. All results are averaged over 5 seeds. Additional results are included in Appendix C.6.

Our experiments unfold according to three inter-dependency scenarios, for systems characterized by either redundancy, synergy or a mix of both interactions.

Redundancy benchmark. We consider $R = \mathcal{N}(0, \mathbb{I})$ as the redundant information component in the system. All system variables are of the form $\mathbf{X} = \{\mathbf{X}^1, \dots, \mathbf{X}^N\} = \{R+\epsilon_i, \dots, R+\epsilon_N\}$, where $\epsilon_i \sim \mathcal{N}(0, \sigma\mathbb{I})$ are mutually independent random noise samples with standard deviation σ. We use σ to modulate the redundancy level: higher noise levels decrease the strength of redundant interaction, and this has an impact on the value of O-information.

Next, we discuss results for a system with $N = 10$ variables, organized as 3 redundant subsystem, each defined as described above. Figure 5.1 illustrates, for various variable dimension, ranging from 5 to 20 dimensional Gaussians, the ground-truth and the estimated O-information, for SΩI and the various baselines. In this scenario, SΩI and baseline competitors produce fairly accurate O-information estimates, when the dimensionality of each random variable is small. When the dimension of systems variables grows, however, the performance of the baseline methods degrades considerably. This is due to the inherent limitations of the pairwise neural MI estimators, that struggle with high dimensional

data (Czyż et al., 2023). Instead, the performance of SΩI remains stable when increasing variable dimension, and O-information estimates are accurate, even when interaction strength is high.



(a) Dim = 5 **(b)** Dim = 10 **(c)** Dim = 15 **(d)** Dim = 20

Figure 5.1: Redundant system with N =10 variables, organized into subsets of sizes $\{3, 3, 4\}$ and increasing interaction strength.

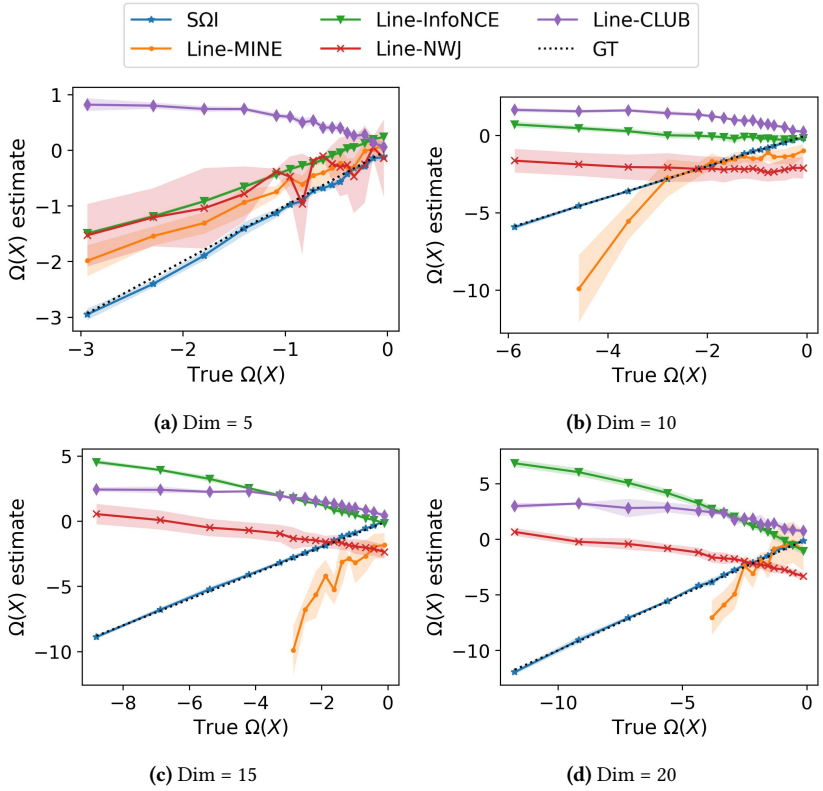
Synergy benchmark. In this case, we synthesize synergistic inter-dependency among system variables by considering the following setup. For simplicity, consider three random variables that behave as follows:

$$\mathbf{X}^1 \sim \mathcal{N}(0, \mathbb{I}), \quad \mathbf{X}^2 = \mathbf{X}^1 + S$$
$$\mathbf{X}^3 = S + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma) \text{ and } \mathbf{X}^1 \perp\!\!\!\perp \mathbf{X}^3$$

with $S \sim \mathcal{N}(0, \mathbb{I})$.

When $\sigma = 0$, the synergy emerges through the Markov chain $\{\mathbf{X}^2, \mathbf{X}^3\} - \mathbf{X}^1$, $\{\mathbf{X}^1, \mathbf{X}^3\} - \mathbf{X}^2$ and $\{\mathbf{X}^1, \mathbf{X}^2\} - \mathbf{X}^3$, since no element alone is sufficient to recover the remaining variables. We modulate σ to achieve different synergistic strengths. More generally, we simulate N synergistic variables as: $\mathbf{X}^1 \sim \mathcal{N}(0, \mathbb{I})$, $\mathbf{X}^2 = \mathbf{X}^1 + S_1 + ... + S_{N-2}$ and $\mathbf{X}^i = S_{i-2} + \epsilon_{i-2} \forall i \in \{3, .., N\}$.

Results in Figure 5.2 show that SΩI achieves consistent results in all scenarios, whereas the baselines behave poorly. Indeed, a synergy-only setting is challenging, as it's dominated by high DTC values required to capture high-order interactions, on which the baselines based on pairwise MI estimator fail.



(a) Dim = 5 **(b)** Dim = 10

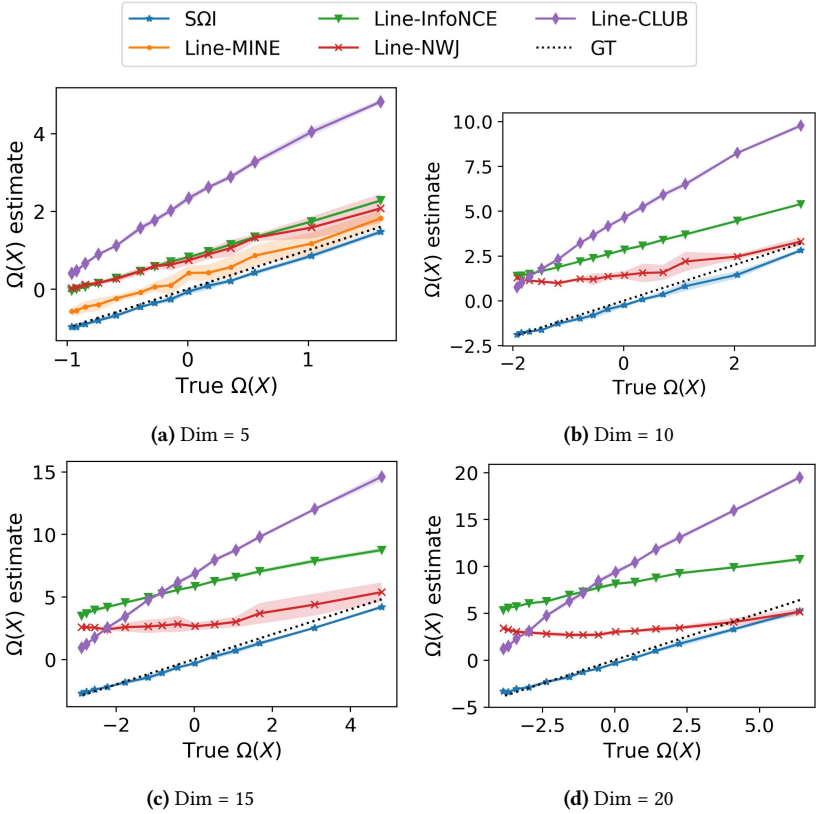
(c) Dim = 15 **(d)** Dim = 20

Figure 5.2: Synergistic system with $N = 10$ variables, organized into subsets of sizes $\{3, 3, 4\}$ and increasing interaction strength.

Mixed benchmark. In general, systems components are characterized by a mix of redundant and synergistic interactions. Then, we synthesize such a system by creating subgroups dominated by redundancy and synergy, respectively, following the procedures defined above.

Results in Figure 5.3, demonstrate that our method SΩI stands out as the best estimator in this challenging scenario. Baseline methods produce poor estimates, especially when the synergistic interaction is dominant. Note that SΩI reports a negative O-information whenever the system is synergy-dominant and also succeeds in capturing interaction strengths, when the system equilibrium changes in favor redundant interactions, by estimating correctly a positive O-information.

Discussion. We attribute the superior performance of SΩI, compared to the baselines, to several factors. Score-based estimators have shown to be extremely successful in fitting



(a) Dim = 5 **(b)** Dim = 10

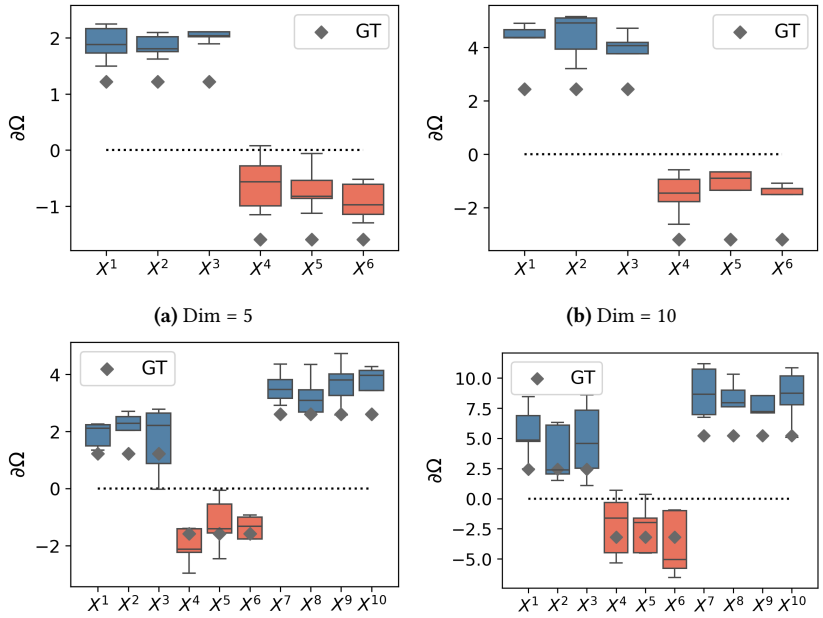
(c) Dim = 15 **(d)** Dim = 20

Figure 5.3: Mixed-interaction system with $N = 10$ variables, organized into 2 redundancy-dominant subsets of size $\{3, 4\}$ variables and one synergy-dominant subset with 3 variables. O-information is modulated by fixing the synergy inter-dependency and increasing the redundancy.

complex distributions, for example in the context of generative modeling (Song and Ermon, 2020; Song et al., 2021). Moreover, our technique relies on Proposition 2, whereby the difference of score functions has been shown to produce an accurate estimate of KL divergences, due to canceling effects of estimation errors (Franzese, Bounoua, and Michiardi, 2024). Note also that the baselines we adopt in our work use MI estimators that produce a bound only. Moreover, using individual models to estimate several MI terms can naturally suffer from cumulative bias, which is avoided in our case by amortizing computation with a unique neural network.

Gradient of O-information While O-information provides global information about dominance of either synergy or redundancy, the contribution of individual variables to either effects is not available. Next, we rely on the gradient of O-information to study individual system components, as introduced in § 5.2. Indeed, our method SΩI can be easily extended to output such gradients, by estimating additional score functions, as described in Appendix C.2. In Figure 5.4, we illustrate gradients of O-information applied to the mixed benchmark scenario discussed above. While O-information of the whole system can be

positive due to the redundancy strength of some subgroup of variables, we notice that three variables report a negative gradient, which is indicative of their synergistic interaction. In Figure 5.4, ground truth gradient values are showed using a diamond marker. Our estimator, despite suffering from some bias, correctly attributes the role and interaction type of each system constituent.



(a) Dim = 5 **(b)** Dim = 10

(c) Dim = 5 **(d)** Dim = 10

Figure 5.4: Gradient of O-information for the mixed benchmark, for a system of $N = 6$ variables, and a system of $N = 10$ variables, and different dimension of variables.

5.4.2 Application to a real system

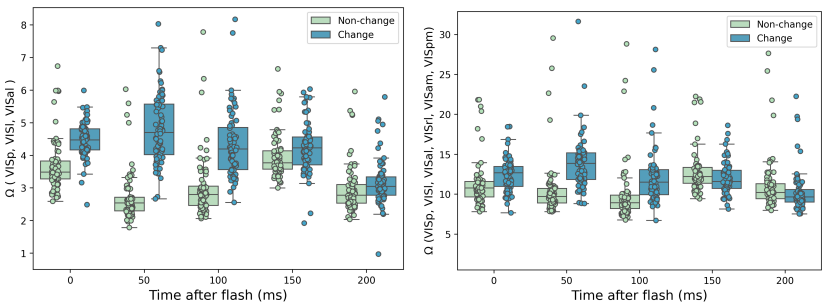
Multivariate analysis is a powerful tool for the field of neuroscience, as it allows scientists to analyze activity patterns of different brain regions. Understanding how the brain processes and transmits information during different stimulus requires analysing the underlying inter-dependencies between different brain regions. To show that SΩI is an effective tool also in practical use cases, we now consider the Visual Behavior project, which used the Allen Brain Observatory to collect a highly standardized dataset consisting of recordings of neural activity in mice that have learned to perform a visually guided task (Allen-Institute, 2022).

A visual change-detection task experiment was conducted on 80 mice using six neuropixels probes tasked to report the activity of different regions of the visual cortex. During the recordings, a set of 8 natural scenes were presented in 250 ms flashes, at intervals of 750 ms. The same image was shown during several flashes before a change to a new image. The mouse had to perform an action to receive a water reward when the image changed.

Ultimately, the purpose of this experiment is to investigate how the different brain region of the mice react to different types of stimulus, such as detecting a new image (*change*) or not (*no change*).

In this work, we follow the prepossessing procedure described by (Venkatesh et al., 2023), where in each experimental session, good quality units from each area are chosen (See Appendix C.3). For each trial, the recorded spikes are binned in 50 ms intervals, starting from the stimulus flash. We consider two types of flashes: *change* and *no change*. For both cases, SΩI is used for each time bin to estimate O-information (O-information). The reported estimation is done using 10 Monte Carlo integration steps and averaged over multiple seeds. We first consider three visual cortex regions VISp, VISl and VISal, as done in (Venkatesh et al., 2023). We then extend the experiment to six brain regions by including VISrl, VISam and VISpm.

We show our results in Figure 5.5, where the distribution of O-information values are reported as box-plots for each bin. We remark that values of O-information are higher in cases of *change* stimulus, and lower for the *no change* stimulus. This suggests that higher amount of redundant information in the visual cortex regions is transmitted in case of a flash with new scene. Interestingly, when considering six areas of the visual cortex, our observations remain valid, suggesting that the measured behaviour is common to these other brain areas as well. Our results are aligned with (Venkatesh et al., 2023). However, prior work rely on the PID measure, which requires the brain regions to be artificially organized into two areas and a target variable, due to scalability issues affecting PID. Our work confirms that SΩI does not have such a limitation and allows a single estimation procedure to obtain the same conclusions.



(a) 3 areas **(b)** 6 areas

Figure 5.5: O-information estimate in the visual cortex region activity after two types of stimulus flash across 72 trial sessions. Top: Analysis using three brain region areas, Bottom: Extended analysis using six brain region areas. The step size is set to $2ms$ which results in 25 dimensional data for each bin per area. Different step sizes led to the same behavior (see Appendix C.6).

5.5 Conclusion

We addressed the problem of analyzing multivariate systems, whereby the essence of complexity does not only lie in the nature of the individual system components, but also in the structure of their inter-dependencies. Indeed, the analysis of high-order interaction among variables has emerged as an important tool to deepen our understanding of such complex systems, with application domains including machine learning, neuroscience, climate modeling, and many more.

Recently, the scientific community has spent considerable effort on extending information theory to allow the study of complex, multivariate systems according to notions of uniqueness, redundancy and synergy. While no consensus exists yet, on a information measure that can fully and reliably characterize high-order interactions, in this work we focused on O-information, which has desirable properties such as interpretability and scalability in number of variables. The current state of the art is however at a roadblock. The existing techniques rely on strong assumptions on the data distribution. Additionally, we explore an exhaustive use of the neural MI estimators to access the O-information which resulted in sub-optimal performance and scalability issues. Then, the endeavour of our work was to present a method to lift such limitations, and endow practitioners and scientists with a flexible and reliable tool to study complex systems associated to natural phenomena.

In this chapter, we proposed SΩI, a novel technique that leverages recent neural estimators of mutual information and uses score functions of joint and conditional distributions to compute divergences. We showed that SΩI can compute O-information by training a unique parametric model, which is efficient and flexible. We validated our technique with a comprehensive experimental protocol, both in synthetic and realistic settings. We demonstrated that SΩI is accurate and robust across different system configurations and complexities. We also applied SΩI to a case study of mice brain activity, where we obtained plausible and interpretable results. This study also demonstrated the scalability of SΩI, showing that it can handle larger systems than previously possible. The method has potential applications across a wide range of multimodal systems, from industrial systems to medical and biological domains. We believe that our work represents a substantial advancement in the computation of information measures and their application to real-world, complex systems.

Chapter 6

Minimum Entropy Coupling

Multimodal data is a precious asset enabling a variety of downstream tasks in machine learning. However, real-world data collected across different modalities is often not paired, which is a significant challenge to learn a joint distribution. A prominent approach to address the modality coupling problem is Minimum Entropy Coupling (MEC), which seeks to minimize the joint Entropy, while satisfying constraints on the marginals. Existing approaches to the MEC problem focus on finite, discrete distributions, limiting their application for cases involving continuous data. In this work, we propose a novel method to solve the continuous MEC problem, using well-known generative diffusion models that learn to approximate and minimize the joint Entropy through a cooperative scheme, while satisfying a relaxed version of the marginal constraints. We empirically demonstrate that our method, DDMEC, is general and can be easily used to address challenging tasks, including unsupervised single-cell multi-omics data alignment and unpaired image translation, outperforming specialized methods.

6.1 Introduction

Nowadays, multimodal data is pervasive thanks to advances in data collection technologies and the crucial need for systems that can learn from the diversity of real-world phenomena. Healthcare, for example, is a domain where patient data often spans electronic health records, radiological images, genetic data, and wearable sensor outputs (Kline et al., 2022; Acosta et al., 2022). Autonomous systems rely on a suite of sensors, including LiDar, cameras, and ultrasonic sensors, to navigate environments effectively (Caesar et al., 2020; Gu et al., 2023; Franchi et al., 2024). Scientific disciplines, such as astronomy and geoscience, employ multimodal datasets combining spatial, spectral, and temporal data to understand complex

systems (Srivastava, Vargas, and Tuia, 2019; Zhang et al., 2024; Klindžić, Šiljeg, and Kalafatić, 2024).

Modeling multimodal data allows for a more comprehensive understanding, reflecting the inherently multi-faceted nature of the real world. Recent works in representation learning (Radford et al., 2021; Lu, 2023; Manzoor et al., 2023; Chen et al., 2023), the study of multivariate systems (Kaplanis, Mediano, and Rosas, 2023; Liang et al., 2023; Bounoua, Franzese, and Michiardi, 2024), generative modeling (Rombach et al., 2022; Tang et al., 2023; Tang et al., 2023; Bounoua, Franzese, and Michiardi, 2024; Esser et al., 2024), and multimodal conversational agents (Li et al., 2023; Liu et al., 2023; Shukor et al., 2023; Xue et al., 2024; Wu et al., 2024), are few examples to illustrate the fervent effort in the machine learning community to address and exploit multimodality. However, the intrinsic complexity of multimodal data introduces several challenges that hinder their application in machine learning research. Modality heterogeneity complicates and sometimes impedes geometric comparisons, requiring for example learning a mapping from ambient to latent spaces (Rombach et al., 2022; Tang et al., 2023; Liu et al., 2023; Bounoua, Franzese, and Michiardi, 2024) or stringent assumptions (Liang et al., 2022; Xia et al., 2023; Dong et al., 2024; Ibrahimi et al., 2024; Zhang, Sui, and Yeung-Levy, 2024). Alignment across modalities at spatial, temporal or semantic levels is another challenge, which calls for costly pre-processing steps such as synchronization (Hanchate et al., 2024; Chen et al., 2024; Scirè, 2024; Martin-Turrero et al., 2024).

The major roadblock we address is that of paired multimodal data, which is an underlying assumption in many works in the literature (Radford et al., 2021; Liu et al., 2023; Li et al., 2023; Bounoua, Franzese, and Michiardi, 2024). Paired data – for a given sample, all its various modalities are available – is either expensive, difficult to obtain, or sometimes impossible. For example, in genetic research, data is inherently unpaired due to the nature of the data acquisition process, such as single-cell RNA sequencing data, where measurements destroy the original cells (Kester and Oudenaarden, 2018; Chen, Lake, and Zhang, 2019; Schiebinger et al., 2019). Similarly, matching image data from different domains is a challenging endeavor when paired data is missing, which calls for specialized methods (Zhu et al., 2017; Huang et al., 2018; Pang et al., 2021; Sasaki, Willcocks, and Breckon, 2021; Yang et al., 2023; Sun et al., 2023; Xie et al., 2023).

In this work, we study the problem of unpaired multimodal data through the lens of *coupling*, a fundamental problem in probability theory, that aims at *determining the optimal joint distribution of random variables given their marginal distributions*, with early attempts at solving it dating back to the work by (Fréchet, 1951). The pairing problem belongs to a broad class of methods (Den Hollander, 2012; Lin et al., 2014; Benes and Stepán, 2012; Yu

and Tan, 2018): some cast it through the lens of information-theoretic quantities, where optimality is defined in terms of Entropy minimization or Mutual Information maximization, others focus on optimal transport (OT) (Villani, 2009; Peyré and Cuturi, 2019), where optimality is defined as minimizing the expected value of a transport cost over the joint distribution. Our focus is the MEC problem, which aims at finding the joint distribution with the smallest Entropy, given the marginal distribution of some random variables. Recent applications include entropic causal inference (Kocaoglu et al., 2017; Javidian et al., 2021; Compton, 2022), communication systems (Sokota et al., 2022), steganography (Witt et al., 2022), random number generation (Li, 2021), dimensionality reduction (Cicalese, Gargano, and Vaccaro, 2016; Vidyasagar, 2012), lossy compression (Ebrahimi, Chen, and Khisti, 2024), and multimodal learning (Liang et al., 2024).

While the MEC problem is known to be NP-Hard (Vidyasagar, 2012; Kovačević, Stanojević, and Šenk, 2012), the literature contains many approximation and greedy algorithms (Painsky, Rosset, and Feder, 2013; Kovacevic, Stanojevic, and Senk, 2013; Cicalese, Gargano, and Vaccaro, 2016; Li, 2021), and theoretical studies about the approximation qualities of such approaches (Cicalese, Gargano, and Vaccaro, 2017; Cicalese, Gargano, and Vaccaro, 2019). Nevertheless, the vast majority of prior work on the MEC problem focus on discrete distributions: instead, we consider the continuous variant of MEC, and propose a flexible and general solution to the coupling problem for arbitrary, continuous distributions. The MEC problem for continuous random variables is much more complex than its discrete counterpart, and can be ill-defined in certain cases due to the properties of differential Entropy and the challenges inherent to continuous distributions living in an infinite dimensional space.

The gist of our method is to consider a parametric class of joint distributions, which we reinterpret as conditional generative models, with additional terms to steer adherence to marginal constraints. Then, the MEC problem requires access to the conditional Entropy, which we rewrite as log-likelihood. Crucially, our method exploits two specular generative models, which cooperate to minimize the joint Entropy, while approximately satisfying the marginal constraints. Our approach materializes as two denoising diffusion probabilistic models (Ho, Jain, and Abbeel, 2020), which we first pre-train on marginal distributions, and then fine-tune according to reward functions, following an alternating optimization process. In summary, our contributions are:

- We propose an approximation of the MEC problem for arbitrary, continuous distributions, which is general, and that does not require stringent assumptions on the marginal distributions, nor the definition of geometric cost functions (§ 6.2).

- We present a practical implementation of our method (§ 6.3), that relies on generative models, that interact through a cooperative scheme aiming at optimizing an information-theoretic cost function related to the Entropy of the joint distribution. Our training procedure overcomes numerical instabilities and degenerate solutions by relying on the application of soft marginal constraints, as well as the natural approximation stemming from a finite-capacity denoising model.

- We illustrate the benefits and performance of our method on two important use cases (§ 6.4). First, we solve the coupling problem between incomparable spaces with a single-cell multi-omics dataset, where we compare our method to state-of-the-art alternatives that rely on OT. Second, we focus on unsupervised image translation between uncoupled pairs, and compare against state of the art.

6.2 Problem Formulation

Given two random variables $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ with marginal probability distributions $p^{\mathbf{X}}(\mathbf{x})$ and $p^{\mathbf{Y}}(\mathbf{y})$ respectively, we consider a *parametric* space $\mathcal{P}^\theta = \{p_\theta^{\mathbf{X},\mathbf{Y}}(x,y)\}$ of *joint* distributions over the space $\mathcal{X}\times\mathcal{Y}$, with induced marginal distributions $p_\theta^{\mathbf{X}}(\mathbf{x}), p_\theta^{\mathbf{Y}}(\mathbf{y})$ (where $p_\theta^{\mathbf{X}}(\mathbf{x}) \triangleq \int_{\mathcal{Y}} p_\theta^{\mathbf{X},\mathbf{Y}}(\mathbf{x},\mathbf{y})\,\mathrm{d}\mathbf{y}$ and similarly for $p_\theta^{\mathbf{Y}}(\mathbf{y})$). The MEC problem between the two original distributions $p^{\mathbf{X}}(\mathbf{x})$ and $p^{\mathbf{Y}}(\mathbf{y})$ consists in finding a joint distribution $p_\theta^{\mathbf{X},\mathbf{Y}}(\mathbf{x},\mathbf{y})$ such that i) the induced marginal distributions $p_\theta^{\mathbf{X}}(\mathbf{x}), p_\theta^{\mathbf{Y}}(\mathbf{y})$ match them either exactly or approximately and ii) the joint distribution is the one with minimal entropy (Kovacevic, Stanojevic, and Senk, 2013; Cicalese, Gargano, and Vaccaro, 2017; Cicalese, Gargano, and Vaccaro, 2019). The constraints over the search space $\mathcal{P}^\theta$ are referred to as *marginal contraints*

Definition 3. *A joint distribution $p_\theta^{\mathbf{X},\mathbf{Y}}(\mathbf{x},\mathbf{y})$ from $\times_\theta$ is said to be an exact coupling iff*

$$p_\theta^{\mathbf{X}}(\mathbf{x}) = p^{\mathbf{X}}(\mathbf{x}), p_\theta^{\mathbf{Y}}(\mathbf{y}) = p^{\mathbf{Y}}(\mathbf{y}). \tag{6.1}$$

In general, exact coupling is not possible (nor wanted, to avoid overfitting) and the goodness of the solution in terms of marginal constraints is approximated through some distance function between the induced and original distributions, e.g. using the Kullback-Leibler divergence $\mathbb{KL}\left(p_\theta^{\mathbf{X}} \parallel p^{\mathbf{X}}\right) \triangleq \mathbb{E}_{\mathbf{x}\sim p_\theta^{\mathbf{X}}}\left[\log \frac{p_\theta^{\mathbf{X}}}{p^{\mathbf{X}}}(\mathbf{x})\right]$. Then, we define the MEC problem with *soft* constraints as follows

$$\min_\theta \mathcal{H}(p_\theta^{\mathbf{X},\mathbf{Y}}) + \lambda_{\mathbf{X}}\mathbb{KL}\left(p_\theta^{\mathbf{X}} \parallel p^{\mathbf{X}}\right) + \lambda_{\mathbf{Y}}\mathbb{KL}\left(p_\theta^{\mathbf{Y}} \parallel p^{\mathbf{Y}}\right), \tag{6.2}$$

where the entropic term is defined as $\mathcal{H}(p_\theta^{\mathbf{X},\mathbf{Y}}) \triangleq -\mathbb{E}_{\mathbf{x},\mathbf{y}\sim p_\theta^{\mathbf{X},\mathbf{Y}}}\left[\log p_\theta^{\mathbf{X},\mathbf{Y}}(\mathbf{x},\mathbf{y})\right]$.

Previous work have mainly focused on the exact MEC in discrete settings, where $p^{\mathbf{X}}$ and $p^{\mathbf{Y}}$ have a finite or countably infinite number of outcomes. Exact solution in such settings is known to be NP-Hard (Vidyasagar, 2011; Kovacevic, Stanojevic, and Senk, 2013). Under our assumption of continuous spaces the problem is more complex. Exact matching is not generally possible due to the finite complexity of the parametric family $\times_\theta$, since in general the distributions $p^{\mathbf{X}}, p^{\mathbf{Y}}$ live in infinite dimensional spaces. Rather than a limitation, enforcing limited complexity is helpful to avoid degenerate, deterministic joint probabilities (e.g. $p_\theta^{\mathbf{X},\mathbf{Y}}(\mathbf{x},\mathbf{y}) = \delta(y - g(\mathbf{x}))p_{\mathbf{X}}(\mathbf{x})$, where $g(\cdot)$ is any mapping which guarantees exact coupling), which would induce infinite joint entropy.

Interestingly, the MEC problem has an intuitive interpretation connected to the problem of Mutual Information maximization. We can write the MI when $\mathbf{X}, \mathbf{Y} \sim p_\theta^{\mathbf{X},\mathbf{Y}}$ as :

$$\mathcal{I}(\mathbf{X},\mathbf{Y}) = -\mathcal{H}(p_\theta^{\mathbf{X},\mathbf{Y}}) + \mathcal{H}(p_\theta^{\mathbf{X}}) + \mathcal{H}(p_\theta^{\mathbf{Y}}) \tag{6.3}$$

and in the exact matching scenario $\mathcal{H}(p_\theta^{\mathbf{X}}) = \mathcal{H}(p^{\mathbf{X}}), \mathcal{H}(p_\theta^{\mathbf{Y}}) = \mathcal{H}(p^{\mathbf{Y}})$. In other words, whenever the marginal constraints are satisfied with reasonable quality, the MEC problem is a good approximation of the information maximization problem.

Early instances of the coupling problem express it through the lenses of OT (Monge, 1781; Kantorovich, 1942). In the simplest (albeit rich and interesting) scenario, the goal is to minimize the transportation cost between distributions $\mathbb{E}_{\mathbf{x},\mathbf{y}\sim p_\theta^{\mathbf{X},\mathbf{Y}}}[||\mathbf{x}-\mathbf{y}||^2]$, with the implicit assumption of $\mathcal{X} = \mathcal{Y} = \mathbb{R}^N$ and under the requirement of exact matching (corresponding to $\lambda_{\mathbf{X}} = \lambda_{\mathbf{Y}} = \infty$). Several interesting extensions, including additional constraints on the joint distribution such as geometry or structural constraints, lead to tailor-made approaches (Villani, 2009; Peyré and Cuturi, 2019). Other than the trivial relaxation of constraints from exact to approximate, a particularly useful extension concerns the *entropy-regularized* version of this problem, where the cost function is complemented by the entropic term $\mathcal{H}(p_\theta^{\mathbf{X},\mathbf{Y}})$. Although MEC is fundamentally different than OT, a link between the two clearly exists. However, a straightforward comparison is not possible, as the entropic term enters the respective minimization problems with different signs. Minimizing $\mathcal{H}(p_\theta^{\mathbf{X},\mathbf{Y}})$ directly over other geometric costs (like the euclidean norm considered in OT) has several advantages in terms of generality, as it does not require geometrically comparable spaces $\mathcal{X}$ and $\mathcal{Y}$.

6.3 Methodology

Consider two random variables in continuous domains, $\mathbf{X} \in \mathcal{X}$ and $\mathbf{Y} \in \mathcal{Y}$. We begin by considering a parametric class for the joint distribution expressed as $p_\theta^{\mathbf{X},\mathbf{Y}}(\mathbf{x},\mathbf{y}) = p_\theta^{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y})\, p^{\mathbf{Y}}(\mathbf{y})$, such that the joint entropy $\mathcal{H}(p_\theta^{\mathbf{X},\mathbf{Y}})$ minimization becomes equivalent to minimizing the conditional entropy $\mathcal{H}(p_\theta^{\mathbf{X}|\mathbf{Y}=\mathbf{y}})$. Note that the marginal constraint on $\mathbf{Y}$ from Eq. (6.2) is verified by construction. To satisfy the marginal constraint on $\mathbf{X}$ we consider the KL. This leads to an alternative definition of the MEC problem with soft constraints, that reads as

Definition 4. *Given random variables $\mathbf{X} \in \mathcal{X}$ and $\mathbf{Y} \in \mathcal{Y}$, the continuous MEC problem with soft marginal constraints corresponds to the optimization problem*

$$\min_\theta \mathbb{E}_{\mathbf{y}\sim p^{\mathbf{Y}}} \left[\mathcal{H}(p_\theta^{\mathbf{X}|\mathbf{Y}=\mathbf{y}})\right] + \lambda_{\mathbf{X}} \mathbb{KL}\left(p_\theta^{\mathbf{X}} \parallel p^{\mathbf{X}}\right). \tag{6.4}$$

Crucially, we note that the parametric portion of the joint distribution, namely $p_\theta^{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y})$, can be interpreted as a conditional generative model of the variable $\mathbf{X}$ given $\mathbf{Y}$. As a consequence, the conditional entropy from Definition 4, can be interpreted as an expected log-likelihood, leading to

$$\min_\theta \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p_\theta^{\mathbf{X},\mathbf{Y}}} \left[-\log\left(p_\theta^{\mathbf{X}|\mathbf{Y}=\mathbf{y}}\right)\right] + \lambda_{\mathbf{X}} \mathbb{KL}\left(p_\theta^{\mathbf{X}} \parallel p^{\mathbf{X}}\right). \tag{6.5}$$

A maximum likelihood solution to the MEC problem in Eq. (6.5) is appealing, because it can be addressed by learning the parameters of an appropriate *conditional* generative model, while approximating the marginal constraints on $\mathbf{X}$ through the unconditional version of the model. Nevertheless, this approach bears several challenges:

- Asymmetry: Eq. (6.5) can be used to minimize the conditional entropy $\mathcal{H}(p_\theta^{\mathbf{X}|\mathbf{Y}=\mathbf{y}})$. The learned conditional generative model can be used to generate samples from variable $\mathbf{X}$ given $\mathbf{Y}$, but not vice-versa.

- Marginal constraint: in principle, exact matching requires $\lambda_{\mathbf{X}} \to \infty$, but this choice leads to degenerate solutions to the MEC problem. The marginal constraint from Definition 4, despite being *soft*, should strive to keep $p_\theta^{\mathbf{X}}$ anchored to $p^{\mathbf{X}}(\mathbf{x})$, which is not known.

To address the first challenge, we introduce a second family of parametric models $p_\phi^{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x})p^{\mathbf{X}}(\mathbf{x})$, this time corresponding to conditional generative model of the variable $\mathbf{Y}$

given observations of $\mathbf{X}$. Then, we can write a specular version of the MEC problem we defined as

$$\min_{\phi} \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p^{\phi}_{\mathbf{X},\mathbf{Y}}} \left[-\log \left(p_{\phi}^{\mathbf{Y}|\mathbf{X}=\mathbf{x}} \right) \right] + \lambda_{\mathbf{Y}} \mathbb{KL} \left(p_{\phi}^{\mathbf{Y}} \parallel p^{\mathbf{Y}} \right). \tag{6.6}$$

Recall that $p_{\theta}^{\mathbf{X},\mathbf{Y}} = p_{\theta}^{\mathbf{X}|\mathbf{Y}} p^{\mathbf{Y}}$ and $p_{\phi}^{\mathbf{X},\mathbf{Y}} = p_{\phi}^{\mathbf{Y}|\mathbf{X}} p^{\mathbf{X}}$: it is then reasonable to strive, among all the possible solutions, for $p_{\theta}^{\mathbf{X},\mathbf{Y}} = p_{\phi}^{\mathbf{X},\mathbf{Y}}$. This *joint constraint* can be approximated with a penalty term proportional to the KL divergence between the two distributions. Interestingly, this coupling allows to implement a practical method that exploits cooperation: we use $p_{\phi}^{\mathbf{Y}|\mathbf{X}}$ to improve $p_{\theta}^{\mathbf{X}|\mathbf{Y}}$, and vice-versa.

To address the second challenge, and pave the way for our practical implementation, we break the optimization problem by first focusing on respecting the marginal constraints. To do so, we pretrain unconditional models such that $p_{\theta_*}^{\mathbf{X}}(\mathbf{x}) \approx p^{\mathbf{X}}(\mathbf{x})$ and $p_{\phi_*}^{\mathbf{Y}}(\mathbf{y}) \approx p^{\mathbf{Y}}(\mathbf{y})$. Then, we use θ_* and ϕ_* to initialize the conditional models, and anchor their parameters throughout the optimization such that they do not deviate too much from the pretrained models.

Overall, the method we propose writes as

$$\begin{aligned} &\min_{\theta} \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p_{\theta}^{\mathbf{X},\mathbf{Y}}} \left[-\log \left(p^{\phi}_{\mathbf{Y}|\mathbf{X}=\mathbf{x}} \right) \right] + \lambda_{\mathbf{X}} \mathbb{KL} \left(p_{\theta}^{\mathbf{X}} \parallel p_{\theta_*}^{\mathbf{X}} \right), \\ &\min_{\phi} \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p^{\phi}_{\mathbf{X},\mathbf{Y}}} \left[-\log \left(p_{\theta}^{\mathbf{X}|\mathbf{Y}=\mathbf{y}} \right) \right] + \lambda_{\mathbf{Y}} \mathbb{KL} \left(p_{\phi}^{\mathbf{Y}} \parallel p_{\phi_*}^{\mathbf{Y}} \right), \end{aligned} \tag{6.7}$$

where we additionally enforce the approximate joint constraint. Notice the difference with Eq. (6.5) and Eq. (6.6): given the structure of the parametric distributions we use, it is possible to show that $\nabla_{\theta} \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p_{\theta}^{\mathbf{X},\mathbf{Y}}} [-\log p_{\theta}^{\mathbf{X}|\mathbf{Y}}] \approx \nabla_{\theta} \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p_{\theta}^{\mathbf{X},\mathbf{Y}}} [-\log p_{\phi}^{\mathbf{Y}|\mathbf{X}}]$ whenever $p_{\theta}^{\mathbf{X},\mathbf{Y}} = p_{\phi}^{\mathbf{X},\mathbf{Y}}$. Strict adherence to the joint constraint, in principle, allows "swapping" the roles of the conditional models without affecting the optimization dynamics, leading to a cooperative method. In practice, we found trough empirical exploration that such a cooperative formulation, albeit approximate, proves to be much more stable than the original problem, and consequently decided to adopt it in our implementation, as described next.

6.3.1 Practical implementation

In our implementation, we consider the parametric class of probability distributions associated to DDPM (Sohl-Dickstein et al., 2015; Ho, Jain, and Abbeel, 2020). These models enjoy excellent performance in fitting complex multimodal data, and allow accurate estimation of information metrics (Franzese, Bounoua, and Michiardi, 2024; Kong et al., 2024; Bounoua, Franzese, and Michiardi, 2024; Dewan et al., 2024).

DDPM. These generative models are characterized by a forward process, that is fixed to a Markov chain that gradually adds Gaussian noise to the data according to a carefully selected variance schedule β_t, i.e. $\mathbf{x}_t = \sqrt{1-\beta_t}\mathbf{x}_{t-1} + \sqrt{\beta_t}\epsilon$ with $\epsilon \sim \mathcal{N}(0, I)$. Interestingly, an arbitrary portion of this forward chain can be efficiently simulated through the equality in distribution $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + (\sqrt{1-\bar{\alpha}_t})\epsilon$, with $\mathbf{x}_0 \sim p^{\mathbf{X}}$ and $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$.

The corresponding discrete-time reverse process, that has a Markov structure as well, is used for generative purposes. The model generates data through the iterative sampling process $p^{\theta}_{\mathbf{X}}(\mathbf{x}_{0\dots T}) = \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)p_\theta(\mathbf{x}_T)$, where $p_\theta(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; 0, I)$ and typically $p^\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is a Gaussian transition kernel with mean $\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon^\theta(\mathbf{x}_t, t)\right)$ and covariance $\beta_t I$. Intuitively, starting from a simple distribution $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, samples are generated by a denoising network ϵ_θ, that removes noise over T denoising steps. A simple way to learn the denoising network ϵ_θ is to consider a re-weighted variational lower bound of the expected marginal likelihood, where the problem $\arg\min_\theta \mathbb{KL}\left(p^{\mathbf{X}} \parallel p^{\mathbf{X}}_\theta\right)$ becomes

$$\arg\min_{\theta} \sum_{t=1}^{T} \mathbb{E}_{\epsilon\sim\mathcal{N}(0,I),\mathbf{x}_0\sim p^{\mathbf{X}}}\left[||\epsilon - \epsilon_\theta(\mathbf{x}_t, t)||^2\right]. \tag{6.8}$$

This simple formulation has been extended to conditional generation (Ho and Salimans, 2022), whereby a conditioning signal y injects "external information" in the iterative denoising process. This requires a simple extension to the denoising network such that it can accept the conditioning signal: $\epsilon_\theta(\mathbf{x}_t, \mathbf{y}, t)$. During training, a randomized approach allows to learn both the conditional and unconditional variants of the denoising network, for example by assigning a null value to the conditioning signal, e.g. $\mathbf{y} = \emptyset$.

In Eq. (6.7), the log-likelihood emerges as a critical quantity to address the MEC problem. In the ideal conditions of a *perfect* denoising network, the difference between predicted and actual noise can be used, in the limit of infinite number of denoising steps, to compute

exactly such quantity (Kong, Brekelmans, and Ver Steeg, 2022). We use these results to compute the log-likelihoods through Monte Carlo estimation techniques

$$-\log p_\theta(\mathbf{x}_0) \approx \text{const} + \frac{1}{2}\sum_{t=1}^{T} \mathbb{E}_\epsilon \left[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\|_2^2\right], \tag{6.9}$$

where the unspecified constant does not depend neither on $\mathbf{x}_0$ nor on θ, and is consequently irrelevant for optimization purposes. This approach can be trivially generalized to the case of a conditional denoising network $\epsilon_\theta(\mathbf{x}_t, \mathbf{y}, t)$.

Our method DDMEC: We being by pretraining *unconditional* models such that $p^{\mathbf{X}}_{\theta_*}(\mathbf{x}) \approx p^{\mathbf{X}}(\mathbf{x})$ and $p^{\mathbf{Y}}_{\phi_*}(\mathbf{y}) \approx p^{\mathbf{Y}}(\mathbf{y})$. Then, we use θ_* and ϕ_* to initialize conditional models $p_\theta^{\mathbf{X}|\mathbf{Y}}$ and $p_\phi^{\mathbf{Y}|\mathbf{X}}$, which use denoising networks that accept additional conditioning signals, following (Zhang, Rao, and Agrawala, 2023). Next, we interpret the optimization expressed in Eq. (6.7) as a model fine-tuning objective, which is reminiscent of the work by (Fan et al., 2023):

$$\begin{aligned}
&\min_\theta \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p_\theta^{\mathbf{X},\mathbf{Y}}} r_\phi(\mathbf{y},\mathbf{x}) + \tilde{\lambda}_{\mathbf{X}} \mathbb{KL}\left(p_\theta^{\mathbf{X}} \parallel p^{\mathbf{X}}_{\theta_*}\right), \\
&\min_\phi \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p_\phi^{\mathbf{X},\mathbf{Y}}} r_\theta(\mathbf{x},\mathbf{y}) + \tilde{\lambda}_{\mathbf{Y}} \mathbb{KL}\left(p_\phi^{\mathbf{Y}} \parallel p^{\mathbf{Y}}_{\phi_*}\right),
\end{aligned} \tag{6.10}$$

where $r_\phi = -\log p_\phi^{\mathbf{Y}|\mathbf{X}}$ and $r_\theta = -\log p_\theta^{\mathbf{X}|\mathbf{Y}}$ are reward signals striving to minimize the conditional entropies, and $\tilde{\lambda}_{\mathbf{X}}, \tilde{\lambda}_{\mathbf{Y}}$ are scaling factors used for fine-tuning, that no longer require to be extremely large. Furthermore, we enforce the joint constraints via extra penalty terms $\mathbb{KL}\left(p_\theta^{\mathbf{X},\mathbf{Y}} \parallel p_\phi^{\mathbf{X},\mathbf{Y}}\right), \mathbb{KL}\left(p_\phi^{\mathbf{X},\mathbf{Y}} \parallel p_\theta^{\mathbf{X},\mathbf{Y}}\right)$.

Fine-tuning DDPM introduces significant computational overhead. To address this, various studies have explored supervised methods (Lee et al., 2023; Wu et al., 2023) or reinforcement learning. In (Clark et al., 2023; Xu et al., 2024), fine-tuning is achieved through direct back-propagation through the reverse process, which can be costly. Alternative methods use proximal policy optimization (PPO) (Fan et al., 2023; Black et al., 2024; Uehara et al., 2024), leading to improved stability. Note that (Fan et al., 2023) incorporates KL-regularization to maximize the reward signal, while ensuring fidelity to the pretrained model, which is analogous to our soft marginal constraints.

In our implementation, we compute gradients of the reward $\nabla_\theta \mathbb{E}_{p_\theta^{\mathbf{X},\mathbf{Y}}} r_\phi(\mathbf{y},\mathbf{x})$ as fol-

lows (Fan et al., 2023)

$$\mathbb{E}_{p_\theta^{\mathbf{X},\mathbf{Y}}} r_\phi(\mathbf{y},\mathbf{x}) \sum_{t=1}^{T} \nabla_\theta \log p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, y), \tag{6.11}$$

while the gradient of the marginal constraints $\nabla_\theta \mathbb{KL}\left(p_\theta^{\mathbf{X}} \parallel p_{\theta_*}^{\mathbf{X}}\right)$ are obtained as the approximate gradient of an upper bound (Fan et al., 2023)

$$\sum_{t=1}^{T} \nabla_\theta \mathbb{E}_{\mathbf{x}_t}\left[||\epsilon_\theta(\mathbf{x}_t, y, t) - \epsilon_{\theta_*}(\mathbf{x}_t, t)||^2\right]. \tag{6.12}$$

Similar expressions apply to the specular model.

Algorithm 1: DDMEC Training Loop

1 **Input:** θ_*, ϕ_*
2 Initialize $\theta \leftarrow \theta_*, \phi \leftarrow \phi_*$
3 **Repeat**
4 Call Algorithm 2 with $\mathbf{y} \sim p^{\mathbf{Y}}, \theta, \theta_*, \phi$
5 Call Algorithm 2 with $\mathbf{x} \sim p^{\mathbf{X}}, \phi, \phi_*, \theta$
6 **Converged**

Given pretrained models $p_{\theta_*}^{\mathbf{X}}, p_{\phi_*}^{\mathbf{Y}}$, the pseudo-code of our DDMEC method in Algorithm 1 is extremely simple, as it materializes as alternating optimization steps, described (for the top Eq. (6.10)) in Algorithm 2. First, we optimize for the parameters θ of the model $p_\theta^{\mathbf{X}|\mathbf{Y}=\mathbf{y}}$, while fixing the parameters ϕ of the specular model $p_\phi^{\mathbf{Y}|\mathbf{X}}$, which we use as a *reward* term. Then we adapt the parameters ϕ to ensure $p_\phi^{\mathbf{X},\mathbf{Y}} \approx p_\theta^{\mathbf{X},\mathbf{Y}}$: this is achieved by noting that we can adapt Eq. (6.8) to this purpose, whereby the parameters θ are now fixed. In the second phase (which can described as the specular version of Algorithm 2), we optimize for the parameters ϕ of the model $p_\phi^{\mathbf{Y}|\mathbf{X}=x}$, while fixing the parameters θ of the model $p_\theta^{\mathbf{X}|\mathbf{Y}}$ using the corresponding reward term. Finally, in a specular manner to the first phase, we ensure coherency of the two models by adapting θ such that $p_\theta^{\mathbf{X},\mathbf{Y}} \approx p_\phi^{\mathbf{X},\mathbf{Y}}$, thus satisfying the joint constraint. For a detailed overview of the DDMECmethodology, we refer the reader to Figure 6.1.

6.4 Experiments

DDMEC is a general method that can be applied across a variety of data domains, as it relies on an information-theoretic measure to match unpaired entities. Next, we demonstrate

Algorithm 2: DDMEC Training Step

1 **Input:** $\mathbf{y}, \theta, \theta_*, \phi$
2 $\mathbf{x} \sim p_\theta^{\mathbf{X}|\mathbf{Y}=\mathbf{y}}, t \sim \mathcal{U}[0,T], \epsilon \sim \mathcal{N}(0,\mathbf{I})$
3 Update θ using Eq. (6.11) and Eq. (6.12)
4 Update ϕ using $\nabla_\phi \mathbb{E}_{\mathbf{y}_t,t}\left[\|\epsilon - \epsilon_\phi(\mathbf{y}_t, \mathbf{x}, t)\|^2\right]$

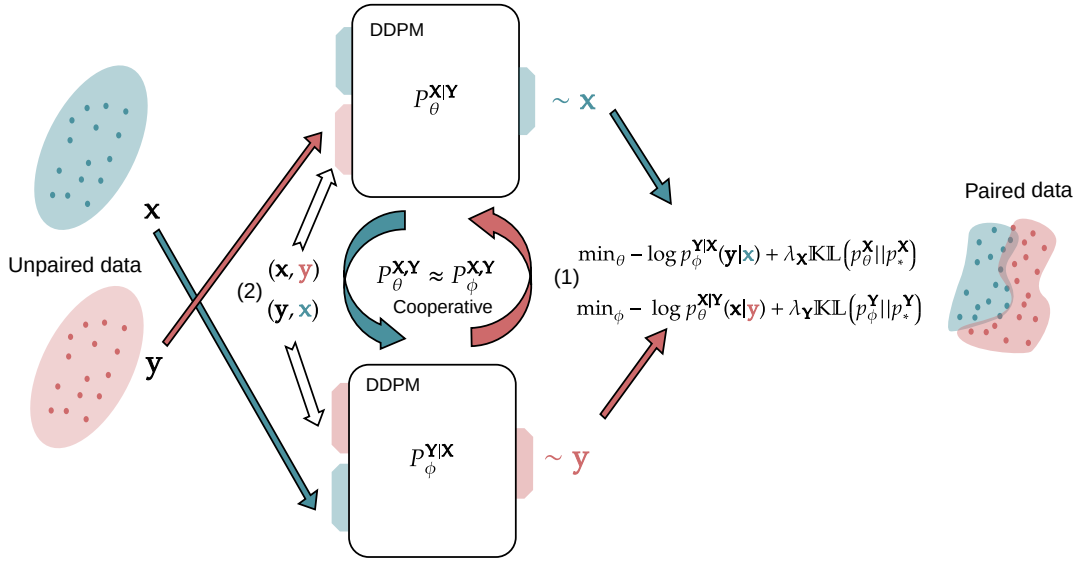


Figure 6.1: Overview of the DDMECmethodology. The phase (1) corresponds to the procedure described in Algorithm 2, Lines 2 and 3, and involves generating samples (depicted in red and blue) conditioned on inputs $\mathbf{x}$ and $\mathbf{y}$ drawn from their respective marginals. These samples are then used to evaluate the loss defined in Eq. (6.10), which is practically optimized using PPO (Fan et al., 2023). The phase (2) corresponds to Line 4 of Algorithm 2, wherein the joint consistency constraint is enforced by updating the model with the previously generated sample pairs. Both phases require coordination between the two models, alternating their roles as outlined in Algorithm 1.

DDMEC versatility using two realistic pairing tasks that use various data modalities, including multi-omics and image data. We compare DDMEC to state-of-the-art methods for each task, and measure performance using domain-specific metrics. Details about DDMEC implementation, and our experimental protocol are given in Appendix D.1.

6.4.1 Multi-omics single-cell alignment

Single-cell measurements techniques, such as mRNA sequencing for whole-transcriptome analysis at the single-cell level (Tang et al., 2009), have been adapted and commercialized by companies which developed platforms to facilitate scalable and efficient single-cell transcriptomics and multi-omics data collection. This data provides a detailed snapshot of the heterogeneous landscape of cells in a sample, and can be used to study the cell developmental

trajectories across time, for example. The availability of multi-omics measurements – capturing various properties of a cell, such as gene expression, mRNA transcriptomes, chromatin accessibility, histone modifications, to name a few – calls for data integration methods to combine a variety of modalities (Xi et al., 2024). Unfortunately, current measurement techniques are destructive: it is hard to obtain multiple types of measurements from the same cell. Furthermore, it is well-known that different cell properties, such as transcriptional and chromatin profiles, cannot be matched using the geometric properties of features in the two domains. Then, pairing single-cell data modalities requires methods that do not rely on either common cells or common features across the data types (Welch, Hartemink, and Prins, 2017; Amodio and Krishnaswamy, 2018; Welch et al., 2019; Stuart et al., 2019).

Baselines. We compare our proposed method DDMEC to several baselines, both from the machine learning and the bio-informatics literature, including INFO-OT (Chuang, Jegelka, and Alvarez-Melis, 2023), SCOT (Demetci et al., 2022), MMD-MA (Liu et al., 2019), and UNIONCOM (Cao et al., 2020). The first two methods, SCOT and INFO-OT, propose variants of an OT formulation, the first based on the Gromov-Wasserstein distance which preserves local neighborhood geometry when moving data points, whereas the second is an information theoretic extension of OT that maximizes the mutual information between domains while minimizing geometric distances. MMD-MA is a global manifold alignment algorithm based on the maximum mean discrepancy measure, whereas UNIONCOM performs unsupervised topological alignment for single-cell multi-omics data that emphasizes both local and global alignment. All alternative methods we consider require geometric distances or similarity measures, which is a pain point that our method DDMEC lifts completely.

Datasets. In our experiments we use the SNARESEQ (Chen, Lake, and Zhang, 2019) dataset, which links chromatin accessibility with gene expression data on a mixture of four cells types. We use the same preprocessing procedures detailed in (Demetci et al., 2022), which deal with filtering spurious data affected by technical errors, and normalization. Data samples have 1–1 correspondence information, which constitute the groud-truth information used for our performance evaluation. We use the average "fraction of samples closer than the true match (FOSCTTM)" metric introduced by (Liu et al., 2019): given a sample in one domain, this amounts to compute the fraction of samples that are positioned more closely to it than its true match after pairing. Results report the average FOSCTTM across all samples, where lower values indicate better performance. We also report the label transfer accuracy as done by (Cao et al., 2020), which measures how well sample labels are transferred between domains based on neighborhood alignment. A k-nearest neighbor classifier is trained on one domain and used to predict labels in the other.

	SNAREseq	
	FOS ↓	Acc ↑
UnionCom⋆	0.265	42.3
MMD-MA⋆	0.150	94.2
SCOT⋆	0.150	98.2
InfoOT⋆	0.156	**98.8**
DDMEC	**0.147**	98.6

Table 6.1: Single-Cell alignment experiments.

Results. Table 6.1 presents the coupling results for the single-cell alignment problem, in which results marked with ⋆ are reported from (Chuang, Jegelka, and Alvarez-Melis, 2023). We remark that DDMEC outperforms alternatives in terms of the FOSCTTM metric, and is on-par with the best method in terms of accuracy. In this experiment, DDMEC is trained once, and inference is conducted five times with different seeds. Unlike other methods, DDMEC is conceptually different as it generates samples rather than learning a deterministic, 1-1 mapping. To compute the different metrics, given a sample from one modality, we use DDMEC to generate a coupling to the other modality, then select the nearest sample from the dataset based on Euclidean distance. Figure 6.2 illustrates the conditional generation of DDMEC in both directions, plotted using PCA: the cell type is correctly transferred from one modality to another, as the generated samples (triangles) largely match the ground truth samples (Circles).

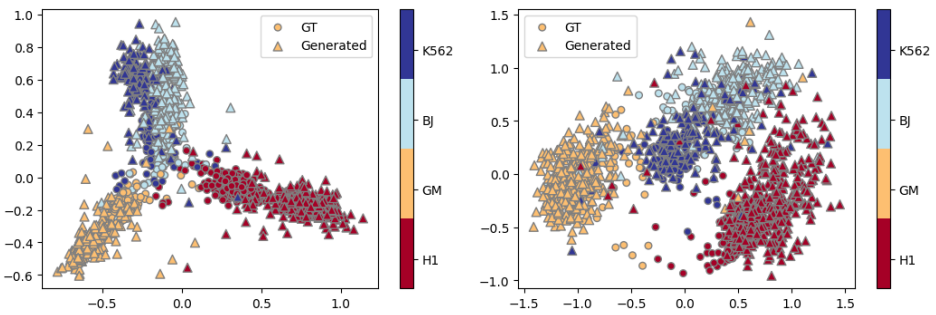


Figure 6.2: Conditional generation using DDMEC on the SNAREseq dataset. The cell types are indicated by colors. **Top:** generation of chromatin accessibility data using gene expression, **Bottom:** generation of gene expression using chromatin accessibility data.

6.4.2 Unpaired image translation

This is a well-known problem in computer vision, where, in the absence of paired data (the joint distribution), the objective is to discover the correct mapping between two image

domains. In this work, we show that unpaired image translation can be framed as a MEC problem, where the goal is to learn the correct joint distribution between two unpaired image domains, $\mathcal{X}$ and $\mathcal{Y}$, respectively. Given the growing popularity of diffusion models in image-related tasks, pretrained weights for various image domains are available: we leverage them in our method DDMEC , as done e.g. by (Zhang, Rao, and Agrawala, 2023).

Baselines. As the literature on image translation is vast, here we primarily focus on the unpaired case, and compare our method to a vast range of alternatives. GAN have been widely applied to this domain (Pang et al., 2021). These methods can be broadly categorized into those focusing on cycle-consistency, which enforces bidirectional mappings between image domains, such as CycleGAN (Zhu et al., 2017), DualGAN (Yi et al., 2017), SCAN (Van Gansbeke et al., 2020), and U-GAT-IT (Kim, 2019); the second category uses distance-based methods, such as DistanceGAN (Benaim and Wolf, 2017), GCGAN (Fu et al., 2019), CUT (Park et al., 2020), and LSeSim (Zheng, Cham, and Cai, 2021). Diffusion-based models, which are related to our method, have also been explored for unpaired image translation. UNIT-DDPM (Sasaki, Willcocks, and Breckon, 2021) learns two conditional models along with two additional domain translation models, incorporating a GAN-like cycle-consistency loss. ILVR (Choi et al., 2021) and SDEdit (Meng et al., 2021) utilize a diffusion model in the target domain while conditioning on a source image to refine the sampling procedure for image translation. EGSDE (Zhao et al., 2022) employs an energy function pretrained on both source and target domains to guide the inference process. Similarly, SDDM (Sun et al., 2023) introduces manifold constraints, forcing distributions at adjacent time steps to be decomposable into denoising and refinement components. Compared to these methods, DDMEC leverages two conditional models, one per domain, which can be initialized using pretrained unconditional diffusion models. By design, our method does not require comparable domains and does not rely on a specific image similarity measure. We report results according to two values for the guidance coefficient, a parameter influencing conditional generation.

Datasets. We adopt the same experimental validation protocol as described by (Zhao et al., 2022), where all images are resized to a resolution of 256×256. We use the AFHQ (Choi et al., 2020) dataset, consisting of high-resolution animal face images across three domains: Cat, Dog, and Wild. This dataset exhibits relatively large variations within and between domains, with 500 test images per domain. We compute the performance of our method DDMEC and compare it to the baselines on Cat→Dog and Wild→Dog tasks. Additionally, we use the CelebA-HQ (Karras, 2017) dataset, featuring high-quality human face images divided into two domains: Male and Female. Each domain $10,000$ and 17000 training

samples for the male and female modalities. Since DDMEC uses two conditional models, translation in the reverse direction is also possible: we include these results in Figure D.2 and Figure D.3.

Results. In Table 6.2, we present the quantitative results of DDMEC: results for alternative methods, marked with $\star$, are reported as obtained in (Sun et al., 2023; Zhao et al., 2022). DDMEC results are reported using 5 seeds. The evaluation is based on generation quality, measured by the FID score (Heusel et al., 2017) (lower is better), and the fidelity to the source domain, assessed using SSIM score (Wang et al., 2004) (higher is better). Note that quality and fidelity can be thought of as divergent objectives: high quality does not imply high fidelity and vice-versa.

GAN-based methods generally suffer from low image quality, except for STARGAN, which achieves a low FID but performs poorly on SSIM. In contrast, diffusion-based methods demonstrate superior performance compared to GAN-based approaches. DDMEC achieves the best FID score in the CAT→DOG task and the highest SSIM in the WILD→DOG task while maintaining comparable results on the remaining metrics. Overall, DDMEC strikes the best balance between high-quality image generation and accurate alignment with the target domain.

Our results on the CELEBA-HQ dataset, demonstrate that DDMECoutperforms competitors on both FID and SSIM even with only 50 sampling steps. Specifically, at 50 steps the FID improves by approximately 1 point and the SSIM by 0.02 points, with an even greater improvement (a 3 point FID reduction) when using 100 sampling steps: it is well-known in the generative modeling literature that these improvements are significant. With the CELEBA-HQ dataset, DDMECbenefits from a larger training set than in AFHQ animal dataset, and achieves state-of-the-art performance on image translation.

This outcome aligns with expectations, as DDMEC is designed to reduce uncertainty and enforce adherence to marginal constraints. In Figure 6.3 and Figure 6.4, qualitative results further confirm the performance of DDMEC in this task. Additional results are available in Figure D.1 and Figure D.3.

6.5 Conclusion

The machine learning community has recently directed substantial effort toward designing multimodal models, as they reflect the inherently multi-faceted nature of the real world. These models often achieve superior performance on downstream tasks compared to uni-

modal counterparts. However, the intrinsic complexity of multimodal data introduces significant challenges. In this work, we addressed the critical problem of coupling data represented by diverse modalities. The coupling problem has been widely studied in the literature, often framed as an optimal transport problem or approached with specialized architectures tailored to specific domains, such as images or language. However, existing methods typically rely on geometric spaces to compute costs, mappings, and similarities between data points.

We proposed a novel method that shifts the focus toward information and uncertainty quantification, thereby circumventing the limiting assumptions of prior approaches. Specifically, we studied the coupling problem through the lens of minimum entropy coupling. Since prior work on MEC has largely been confined to discrete distributions, we extended this framework to continuous distributions. Our key idea lies in introducing a parametric class of joint distributions reinterpreted as conditional generative models, augmented with terms to enforce adherence to marginal constraints. Our approach uses two models, which alternately optimize their objectives while approximately satisfying marginal constraints.

The resulting method enables sampling and generation in either direction between modalities, without requiring specialized embeddings or strict geometric assumptions. Furthermore, it is adaptable to complex settings beyond one-to-one matching between modalities. We validated the performance of our approach in two domains. First, we applied it to multi-omics sequencing data, and we compared our method against several state-of-the-art alternatives that rely on predefined measures for data comparison and coupling cost definition. Our approach, being more general and free from stringent assumptions, achieves performance on par with or superior to these alternatives. Second, we evaluated our method in the image translation domain, comparing it to a range of approaches from the literature. Our method demonstrated superior performance across widely recognized metrics for image quality and coherence, by striking a good balance between these often conflicting measures.

Table 6.2: Quantitative image translation results.

Model	FID↓	SSIM↑
	CAT→DOG	
CycleGAN⋆	85.9	-
MUNIT⋆	104.4	-
DRIT⋆	123.4	-
Distance⋆	155.3	-
SelfDistance⋆	144.4	-
GCGAN⋆	96.6	-
LSeSim⋆	72.8	-
ITTR (CUT)⋆	68.6	-
StarGAN v2⋆	**54.88 ± 1.01**	0.27 ± 0.003
CUT⋆	76.21	**0.601**
SDEdit⋆	74.17 ± 1.01	**0.423 ± 0.001**
ILVR⋆	74.37 ± 1.55	0.363 ± 0.001
EGSDE⋆	65.82 ± 0.77	0.415 ± 0.001
SDDM⋆	62.29 ± 0.63	**0.422± 0.001**
50 Sampling steps		
DDMEC(guidance=9)	**60.70 ± 1.07**	0.410 ± 0.001
DDMEC (guidance=7)	**58.50 ± 0.96**	0.404 ± 0.001
100 Sampling steps		
DDMEC (guidance=9)	**60.51 ± 1.01**	0.403 ± 0.001
DDMEC (guidance=7)	**57.89 ± 0.37**	0.397 ± 0.001
	WILD→DOG	
SDEdit⋆	68.51 ± 0.65	0.343 ± 0.001
ILVR⋆	75.33 ± 1.22	0.287 ± 0.001
EGSDE⋆	59.75 ± 0.62	0.343 ± 0.001
SDDM⋆	**57.38 ± 0.53**	0.328 ± 0.001
50 Sampling steps		
DDMEC(guidance=9)	62.03 ± 1.18	**0.360 ± 0.002**
DDMEC(guidance=7)	60.67 ± 1.01	**0.353 ± 0.004**
100 Sampling steps		
DDMEC(guidance=9)	62.09 ± 0.59	**0.356± 0.001**
DDMEC(guidance=7)	59.22 ± 0.35	**0.346 ± 0.001**
	MALE→FEMALE	
SDEdit⋆	49.43 ± 0.47	0.572 ± 0.000
ILVR⋆	46.12 ± 0.33	0.510 ± 0.001
EGSDE⋆	41.93 ± 0.11	0.574 ± 0.000
SDDM⋆	44.37± 0.23	0.526 ± 0.001
50 Sampling steps		
DDMEC(guidance=2.5)	**40.73** ±0.61	**0.593 ± 0.003**
DDMEC(guidance=2)	**36.99**± 0.83	0.556 ± 0.002
100 Sampling steps		
DDMEC(guidance=2.5)	**38.93 ± 0.37**	**0.588 ± 0.002**
DDMEC(guidance=2)	**34.86 ± 0.70**	0.549 ± 0.002

Figure 6.3: DDMEC (guidance=7) CAT→DOG (*Left*) and WILD→DOG image (*right*) translation examples. Source domain image is used to generate the target dog image.

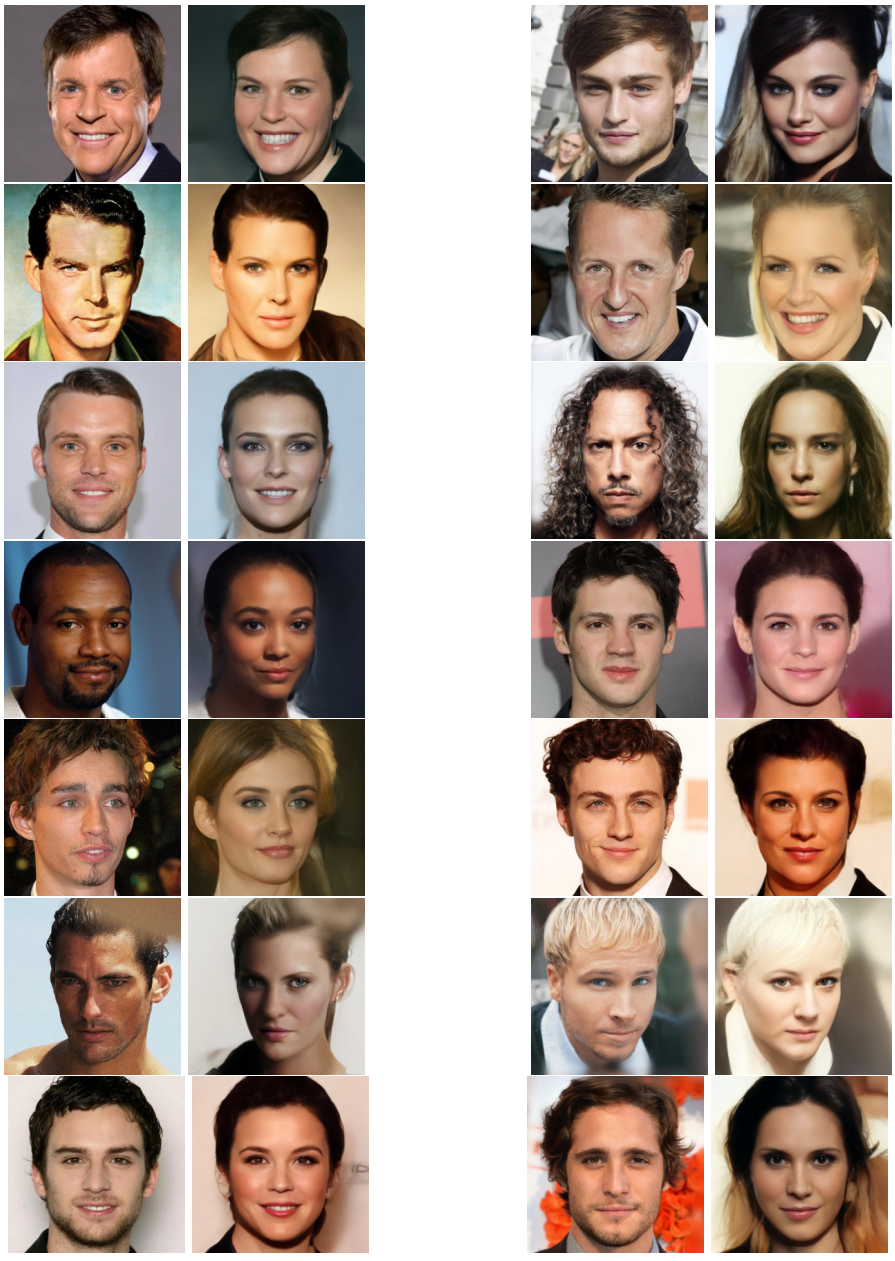


Figure 6.4: DDMEC (guidance=2.5) with 100 sampling steps MALE→FEMALE translation examples. Source domain image is used to generate the target female image.

Chapter 7

Final Remarks and Perspectives

7.1 Summary of Contributions

This thesis makes different contributions, each specialized in a particular area :

- **Multi-modal Generative Modeling** (Chapter 3)

  In this chapter, we tackle multimodal generative modeling, a field long dominated by multimodal VAEs. After analyzing these models and identifying their limitations, we introduced MLD, a novel approach based on score based DMs. MLD employs independently trained, unimodal deterministic autoencoders whose latent variables are concatenated into a shared space and then processed by a masked diffusion model. We also present a new multi-time training method to learn the different conditional scores and allow any-to-any generation. Our method overcomes previous shortcomings to achieve state-of-the-art performance, and we demonstrate its application in the automotive sector by enhancing sensor robustness and improving night vision through the integration of LiDar, RaDar, and camera data.

  Open questions: This work focuses on the generative capabilities of multimodal models, where MLD fully captures joint distributions and their factorization (i.e. conditionals and marginals). A potential that we explore in the remained of the thesis to study the information measures captured by multimodal DM. In contrast to multimodal VAEs, which learn a latent joint representation for downstream tasks, a key open question is how to extract such a representation from MLD. While unimodal approaches have employed strategies like leveraging intermediate layers in the score network or introducing an information bottleneck in the diffusion model, applying similar techniques to MLD could yield useful downstream representations.

- **Mutual Information Estimation** (Chapter 4)

  In this chapter, we present MINDE, a novel method to estimate MI between random variables. By leveraging the Fokker–Planck equation, we reformulate the KL as a difference of score functions which results to the same expression as our published work (which used Girsanov's Theorem in a different framework). We propose a general entropy estimator and naturally a Mutual Information (MI) measurement framework with two strategies: one based on conditional diffusion processes and the other on joint diffusion processes. Experimental results show that our method outperforms current alternatives on challenging distributions and meets MI self-consistency tests. Finally, we demonstrate how pre-trained text-to-image models can compute MI between modalities, aiding the analysis of their generative properties.

  Open questions: Several open questions emerge from this work. A primary concern stems from the inherent limitation of MI in scaling beyond pairwise variable interactions. An important direction, which we explore in Chapter 5, is how to extend the MINDE framework to handle larger sets of variables. Another promising avenue lies in integrating the MINDE estimator within optimization loops aimed at maximizing information-theoretic quantities such as entropy or mutual information. Furthermore, the difference of score functions idea to estimate information measures may be transferable to other generative models closely related to score-based diffusion methods, such as rectified flows.

- **Multi-variate Information Estimation** (Chapter 5)

  in this chapter, we examine multimodal systems where mutual information is insufficient due to multiple random variables. We highlight the importance of synergy and redundancy in capturing higher-order dependencies and identify O-information as a pertinent tool to quantify their balance. Here, we introduce SΩI, a unified model for computing O-information without constraints on the number of modalities. Our experiments on synthetic and real-world data validate the method's effectiveness.

  Open questions: An important open question arises from the lack of consensus regarding optimal multivariate information measures. Although the PID framework (Williams and Beer, 2010) offers an elegant formalism for decomposing information into interpretable components, we have opted for the use of O-information in this thesis, due to its scalability and its adoption of a widely accepted, common definition. Nonetheless, the appealing properties of PID, particularly in contexts where it is necessary to analyze the decomposition of information from (sources to target) perspectives, motivate further the exploration of this methodology. Recent advances have sought to refine the definition of the constituent atoms of PID. A promising direction lies in

connecting these atoms to established information-theoretic measures such as KL or MI. This connection could enable the development of methodologies analogous to SΩI, allowing the computation of PID atoms through the use of score functions. Moreover, multivariate information measures that explicitly incorporate causality and temporal structure, such as those proposed in the ΦID framework (Kaplanis, Mediano, and Rosas, 2023), hold considerable promise for analyzing evolving multimodal systems. Additional open questions, also raised by Chapter 4, include extending our methods to alternative classes of generative models (e.g., rectified flows), and investigating how multivariate information measures can be integrated into the training process.

- **Minimum Entropy Coupling** (Chapter 6)

  In this chapter, we tackle the challenge of learning joint distributions from unpaired multimodal data. We extend MEC problem traditionally applied to discrete distributions to continuous ones via a relaxed formulation. Building on this, we introduce DDMEC, a novel method that uses diffusion models within a reinforcement learning framework to approximate and minimize joint entropy while satisfying relaxed marginal constraints. Our experiments show that DDMECis versatile, outperforming alternatives in tasks like unsupervised single-cell multi-omics alignment and unpaired image translation.

  Open questions: Open questions include extending this method to handle scarce paired or weakly labeled data, a common challenge in semi-supervised settings like single-cell multi-omics alignment. While our domain-agnostic approach avoids reliance on similarity metrics, incorporating additional joint distribution properties might naturally be achieved by integrating them into the reward signal. Moreover, although reinforcement learning based approach introduced in (Fan et al., 2023) effectively manages the iterative sample generation in diffusion models, its computational overhead calls for exploring more efficient training alternatives. Finally, our current focus on pairwise data prompts the question of how to extend the method to multimodal scenarios involving more than two modalities.

7.2 Impact of the Contributions

Our diverse contributions have had a significant impact, inspiring numerous subsequent studies. In this section, we review the various works that stem from the contributions of this thesis. For example, (Di Giacomo et al., 2023; Giacomo et al., 2024) uses the MLD framework presented in Chapter 3, in a multi-view settings, which can be regarded as a subdomain

of multimodality. In this work, MLD serves as the foundation for achieving multi-view cross-generation, while adapting the architecture from (Rombach et al., 2022) as a backbone to attain state-of-the-art performance. This framework was further applied in (Di Giacomo et al., 2024) for data augmentation, resulting in the enhancement of the performance of object classification models.

Furthermore, the difference-of-scores concept developed in Chapter 4 and Chapter 5 has enabled a series of follow-up works. For instance, (Wang et al., 2025) employs the MINDE-estimated MI to fine-tune text-to-image models, achieving improved alignment in a self-supervised manner without relying on external reward signals. These results underscore the utility of the proposed information estimator in quantifying the alignment and dependence between complex data modalities such as text and images. The difference of scores idea has also been extended to discrete data settings in (Foresti, Franzese, and Michiardi, 2025). This work demonstrates the feasibility of constructing mutual information estimators for discrete data distributions by leveraging difference-of-scores associated with noise processes adapted to the discrete data. Furthermore, (Wang et al., 2025) explores rectified flows within this context, showing that this family of generative models can also be employed to estimate information-theoretic quantities. This is achieved by establishing an analogy between velocity fields and score functions, thereby generalizing the difference-of-scores framework to a difference-of-velocity-fields perspective. Finally, (Franzese et al., 2025) introduces a theoretical framework that interprets diffusion models through the lens of nonlinear filtering, revealing how latent abstractions emerge and shape the generative process over time. This work utilizes the mutual information estimator developed in Chapter 4 to quantify the evolving relationship between latent variables and observable data.

7.3 Future Directions

What I cannot create, I do not understand. – **Richard Feynman**

A central focus of this thesis is the exploitation of multimodal data. Echoing Richard Feynman's adage, we start by developing a generative model that captures both the multimodal data distribution and the intricate interactions between modalities. The model's ability to effectively generate data indicates that these interdependencies are well represented. Subsequently, we quantify these dependencies using rigorous information-theoretic measures. Finally, in the concluding chapter, we reverse-engineer this process to design multimodal datasets with tailored interaction properties through information maximization. Several directions can be explored in future work, which we detail next.

Optimized Inference Beyond Diffusion Models In this work, we leverage score-based diffusion models, which, despite their strong performance in generation and information measure estimation, rely on an iterative process that introduces significant computational overhead. This can hinder deployment on low-resource end-user terminals, such as those in automotive systems. To address this challenge, future research should focus on adapting these methods for industrial-scale applications. One promising direction is the exploration of model distillation techniques. For instance, consistency models have been proposed to enable generative sampling with fewer steps by distilling pre-trained diffusion models into more efficient counterparts (Song et al., 2023). Applying our KL estimation via score differences within the framework of consistency models could prove valuable. Another approach is to investigate rectified flows, which streamline the generative trajectories and reduce the number of sampling steps (Lipman et al., 2022), as evidenced by preliminary results in (Wang et al., 2025). Translating our contributions to rectified flows may enhance computational efficiency and facilitate deployment in resource constrained environments.

Multimodal Data Evolving Over Time Multimodal data typically capture the state of an entity from diverse perspectives, resulting in both structural and informational heterogeneity. However, real-world systems are inherently dynamic and evolve over time. A natural extension of this work is to incorporate the temporal dimension into multimodal models, enabling the capture of not only static interdependencies but also their temporal evolution. This direction is particularly promising for applications such as trajectory inference and causal analysis. Incorporating time into multimodal generative modeling would require mechanisms capable of modeling sequential dependencies. One straightforward, yet naïve approach is to treat each modality at each time step as a separate modality. However, this strategy quickly becomes impractical, as the number of modalities scales linearly with the number of time steps, demanding careful architectural and representational design. A more principled alternative is to draw inspiration from recent advances in applying DM to sequential data (Ge et al., 2024; Voleti, Jolicoeur-Martineau, and Pal, 2022). These approaches leverage the flexibility of diffusion models to model temporal dynamics, offering a potential path toward scalable and expressive time-aware multimodal models. Furthermore, integrating autoregressive components within diffusion-based architectures could provide an effective way to model the evolution of multimodal states across time.

Generative Modeling as an Enabler for Industrial applications and Scientific Discovery Generative models offer powerful tools for exploiting multimodal data by enabling cross-modality generation, data augmentation, and the exploration of inter-modal interactions. This thesis has demonstrated applications in fields such as automotive systems, neural

activity, and single-cell multi-omics. Future applications may further advance these areas. In the automotive domain, multimodal generative AI can enhance sensor robustness by integrating multiple sensor inputs and quantifying shared information, ultimately leading to more efficient systems. It may even be possible to generate expensive modalities at test time, effectively creating virtual sensors that reduce hardware costs. In neuroscience, applying multimodal generative models to both brain signals and environmental data could help decode neural activity and uncover underlying brain mechanisms. In biology, particularly in multi-omics studies, generative models that handle unpaired data could facilitate the creation of comprehensive multimodal datasets, offering insights into cellular evolution and dynamics. Extending these techniques to model cellular trajectories over time by incorporating temporal and causal components could open new avenues for research in biological systems.

Appendix

Appendix A

Appendix For Chapter 3

A.1 Additional details about MLD

In this appendix, we provide additional technical details about MLD, including an implementation based on the VPSDE framework (see § 2.2.6). We also explore a naive approach using *inpainting*, which relies solely on the unconditional score network for both joint and conditional generation. Additionally, we introduce an alternative training scheme inspired by techniques from caption-text translation literature (Bao et al., 2023). Finally, we present further technical details on the score network architecture and the sampling techniques used.

A.1.1 Modality Auto-Encoders

Each of the deterministic autoencoders used in the first stage of MLD uses a vector latent space with no size constraints. Instead, VAE-based models generally require the latent space of each individual VAE to be exactly the same size to allow for the definition of a joint latent space. In our approach, the modality-specific latent spaces are *normalized* prior to concatenation using the element-wise mean and standard deviation. In practice, we use the statistics retrieved from the first training batch, which we found to provide sufficient statistical confidence. This operation allows for the harmonization of different modality-specific latent spaces and, thereby facilitates the learning of a joint score network.

A.1.2 Implementation using VPSDE

Following (Ho, Jain, and Abbeel, 2020; Song et al., 2021), we set $\beta_{min} = 0.1$ and $\beta_{max} = 20$. With this configuration, and by substitution of Eq. (3.7), we obtain the following forward SDE:

$$\mathrm{d}\mathbf{Z}_t = -\frac{1}{2}\beta(t)\mathbf{Z}_t\mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}\mathbf{W}_t, \qquad t \in [0, T]. \tag{A.1}$$

The reverse process is described by a different SDE (Eq. (3.8)). When using a variance-preserving SDE, Eq. (3.8) specializes in :

$$\mathrm{d}\mathbf{Z}_t = \left[-\frac{1}{2}\beta(t)\mathbf{Z}_t - \beta(t)\nabla_{\mathbf{z}_t}\log p_t(\mathbf{z}_t)\right]\mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}\bar{\mathbf{W}}_t, \tag{A.2}$$

with $\mathbf{Z}_T \sim \rho(\mathbf{z})$ as the initial condition and time t flowing from T to 0.

Algorithm 3 presents the pseudo-code for the multi-time masked training. The masked diffusion process is applied following randomization with probability d. First, a subset of modalities A_2 is selected randomly to be the conditioning modalities, with A_1 the remaining set of modalities to make up the diffused modalities. The time t is sampled uniformly from $[0, T]$, and the portion of the latent space corresponding to the subset A_1 is diffused accordingly. Using the masking as shown in Algorithm 3, the portion of the latent space corresponding to the subset A_2 is not diffused and is forced to be equal to $\mathbf{z}^{A_2} = \mathbf{z}_0^{A_2}$. The multi-time vector τ is constructed and lastly, the score network is optimized by minimizing a masked loss corresponding to the diffused part of the latent space. With probability $(1 - d)$, all the modalities are diffused at the same time and $A_2 = \emptyset$. We re-weight the loss according to the cardinality of the diffused and frozen portions given the random choice for A_1 and A_2:

$$\omega(A_1, A_2) = 1 + \frac{dim(A_2)}{dim(A_1)}, \tag{A.3}$$

where $\dim(.)$ is the sum of each latent space cardinality of a given subset of modalities with $dim(\emptyset) = 0$.

Algorithm 4 describes the reverse conditional generation pseudo-code. Eq. (3.13) is discretized using a finite time step $\Delta t = {}^{T}/_{N}$. Note that the joint generation can be seen as a special case of Algorithm 4 with $A_2 = \emptyset$.

Algorithm 3: MLD masked multi-time diffusion training step

Data: $\mathbf{x} = \{\mathbf{x}^i\}_{i=1}^M$
Param: d
$\mathbf{z}_0 \leftarrow \{\mathcal{E}_{\phi_i}(\mathbf{x}^i)\}_{i=0}^M$
$A_2 \sim \nu \quad , A_1 \leftarrow \{1, \dots, M\} \setminus A_2$ // Random choice of A_1 and A_2
$t \sim \mathcal{U}[0, T]$
$\mathbf{z}_t \sim p_{0t}(\mathbf{z}_t|\mathbf{z}_0)$ // Apply the perturbation on $\mathbf{z}_0$
$\mathbf{z}_t \leftarrow m(A_1) \odot \mathbf{z}_t + (1 - m(A_1)) \odot \mathbf{z}_0$ // Masked diffusion
$\tau \leftarrow [\mathbb{1}(1 \in A_1)t, \dots, \mathbb{1}(M \in A_1)t]$
Return $\nabla_\chi \left\{\omega(A_1, A_2) \quad \|m(A_1) \odot \quad [s_\chi(\mathbf{z}_t, \tau) - \nabla \log p_{0t}(\mathbf{z}_t \mid \mathbf{z}_0)]\|_2^2 \quad \right\}$

Algorithm 4: MLD conditional generation.

Data: $\mathbf{x}^{A_2}$
$\mathbf{z}_0^{A_2} \leftarrow \{\mathcal{E}_{\phi_i}(\mathbf{x}^i)\}_{i \in A_2}$ // The conditioning modalities
$A_1 \leftarrow \{1, \dots, M\} \setminus A_2$ // The modalities to generate
$\mathbf{z}_t \leftarrow \mathbf{z}_T \leftarrow \mathcal{C}(\mathbf{z}_T^{A_1}, \mathbf{z}_0^{A_2}), \quad \mathbf{z}_T^{A_1} \sim \mathcal{N}(0, \mathbf{I})$ // Compose the initial state
$\Delta t \leftarrow {}^{T}/_{N}$
for $n = N$ ***to*** 1 **do**
  $t \leftarrow n\,\Delta t$
  $\tau(A_1, t) \leftarrow [\mathbb{1}(1 \in A_1)t, \dots, \mathbb{1}(M \in A_1)t]$ // The multi-time vector
  $\epsilon \sim \mathcal{N}(0; I)$ **if** $n > 1$ **else** $\epsilon = 0$
  $\mathbf{z}_t \leftarrow \mathbf{z}_t + \Delta t \left[\frac{1}{2}\beta(t)\mathbf{z}_t + \beta(t)s_\chi(\mathbf{z}_t, \tau)\right] + \sqrt{\beta(t)\Delta t}\epsilon$ // the update step
  $\mathbf{z}_t \leftarrow m(A_1) \odot \mathbf{z}_t + (1 - m(A_1)) \odot \mathbf{z}_T$ // Apply the masking
end
$\hat{\mathbf{z}} \leftarrow \mathbf{z}_t$
Return $\hat{\mathbf{x}}^{A_1} = \{\mathcal{D}_\theta^i(\hat{\mathbf{z}}^i)\}_{i \in A_1}$

A.1.3 Naive Approach : In-painting

As explained in § 3.4.3, the use of the unconditional score network $s_\chi(\mathbf{z}_t, t)$ enables **conditional generation** via the approximation proposed in (Song et al., 2021). Algorithm 5 outlines the procedure for generating a subset of modalities A_1 conditioned on the observed modalities A_2. The available modalities are first encoded into their latent representations $\mathbf{z}^{A_2}$, while the missing components are initialized by sampling from the stationary distribution: $\mathbf{Z}_T^{A_1} \sim \mathcal{N}(0, \mathbf{I})$.

The key idea is to iteratively diffuse the observed latents so that they match the noise level of $\mathbf{Z}_t^{A_1}$, making it possible to apply the unconditional score network across both known and unknown parts. We refer to this method as Multi-modal Latent Diffusion with In-painting (MLD in-paint), and provide detailed comparisons with our proposed MLD approach in Appendix A.2.

Algorithm 5: MLD in-paint conditional generation.

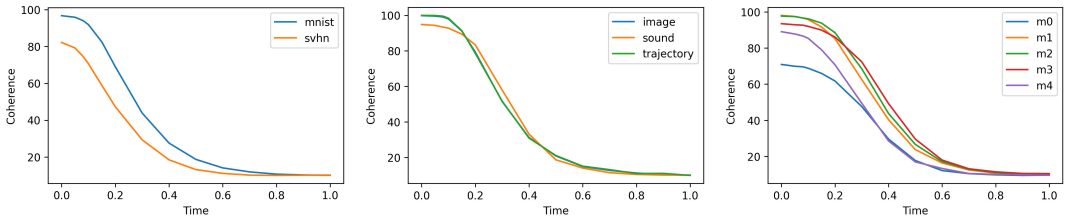
Data: $\mathbf{x}^{A_2}$
$\mathbf{z}_0^{A_2} \leftarrow \{\mathcal{E}_{\phi_i}(\mathbf{x}^i)\}_{i \in A_2}$ // The conditioning modalities
$A_1 \leftarrow \{1, \ldots, M\} \setminus A_2$ // The modalities to generate
$\mathbf{z}_t \leftarrow \mathbf{z}_T \leftarrow \mathcal{C}(\mathbf{z}_T^{A_1}, \mathbf{z}_0^{A_2}), \quad \mathbf{z}_T^{A_1} \sim \mathcal{N}(0, \mathbf{I})$ // Compose the initial state
$\Delta t \leftarrow {}^{T}/_{N}$
for $n = N$ ***to*** 1 **do**
  $t \leftarrow n\,\Delta t$
  $\mathbf{z}_t^{A_2} \sim p_{0t}(\mathbf{z}_t^{A_2} | \mathbf{z}_0^{A_2})$ // Diffuse the conditioning modalities
  $\mathbf{z}_t \leftarrow \mathcal{C}(\mathbf{z}_t^{A_1}, \mathbf{z}_t^{A_2})$
  $\epsilon \sim \mathcal{N}(0; I)$ **if** $n > 1$ **else** $\epsilon = 0$
  $\mathbf{z}_t \leftarrow \mathbf{z}_t + \Delta t \left[\frac{1}{2}\beta(t)\mathbf{z}_t + \beta(t) s_\chi(\mathbf{z}_t, t)\right] + \sqrt{\beta(t)\Delta t}\epsilon$ // the update step
end
$\hat{\mathbf{z}} \leftarrow \mathbf{z}_t$
Return $\hat{\mathbf{x}}^{A_1} = \{\mathcal{D}_\theta^i(\hat{\mathbf{z}}^i)\}_{i \in A_1}$

The approximation enabling the in-painting approach can be efficient in several domains; however, its generalization to the multimodal latent space scenario is not trivial. We argue that this is due to the heterogeneity of modalities. which induce different characteristics on the part of the latent spaces. For different modality-specific latent spaces, the loss of information ratio can vary through the diffusion process. We verify this hypothesis by the following experiment.

Latent space robustness against diffusion perturbation. We analyze the effect of the forward diffusion perturbation on the latent space through time. We encode the modalities using their respective encoders to obtain their latent space $\mathbf{z} = \{\mathcal{E}_{\phi^1}(\mathbf{x}^1)\}_{i=1}^M$. Given a time $t \in [0, T]$, we diffuse the different latent spaces to obtain $\mathbf{z}_t$ being the perturbed version of the latent space at time t. We feed the modality-specific decoders with the perturbed latent space $\hat{\mathbf{x}}_t = \{\mathcal{D}_\theta^i(\mathbf{z}_t^i)\}_{i=1}^M$, with $\hat{\mathbf{x}}_t$ being the output modalities generated using the perturbed latent space. To evaluate the information loss induced by the diffusion process on the different modalities, we assess the coherence preservation in the reconstructed modalities $\hat{\mathbf{x}}_t$ by computing the coherence (in %) as done in § 3.5.

We expect to obtain high coherence results for $t \approx 0$ when compared to $t \approx T$, as the information in the latent space is more preserved at the beginning of the diffusion process than at the last phase of the forward SDE, where all dependencies on initial conditions vanish. Figure A.1 shows the coherence as a function of the diffusion time $t \in [0, 1]$ for different modalities across multiple datasets. It can be observed that, within the same dataset, certain modalities stand out with a specific level of robustness (using the coherence level as a proxy) against the diffusion perturbation in comparison with the remaining modalities

from the same dataset. For instance, we remark that SVHN is less robust than MNIST, which should manifest in under-performance of SVHN-to-MNIST conditional generation. We verify this intuition in Appendix A.2.



(a) MNIST-SVHN **(b)** MHD **(c)** POLYMNIST

Figure A.1: Coherence as a function of the diffusion process time for three datasets. Diffusion perturbation is applied on the modalities' latent space after element-wise normalization.

A.1.4 Uni-Diffuser Training

The work presented in (Bao et al., 2023) is specialized for an image–caption application. The approach is based on a multimodal diffusion model applied to a unified latent embedding obtained via pretrained autoencoders and incorporates pretrained models (CLIP (Radford et al., 2021) and GPT-2 (Radford et al., 2019)). The unified latent space is composed of an image embedding, a CLIP image embedding, and a CLIP text embedding. Note that the CLIP model is pretrained on (image–text) pairs of multimodal data, which is expected to enhance the generative performance. Because it is non-trivial to have a jointly trained encoder similar to CLIP for any type of modality, the evaluation of this model on different modalities across different datasets (e.g., including audio) is not an easy task.

To compare to this work, we adapted the training scheme presented in (Bao et al., 2023) to our MLD method. Instead of applying a masked multimodal SDE to train the score network, every portion of the latent space was diffused according to a different time $t^i \sim \mathcal{U}(0,1)$; therefore, the multi-time vector fed to the score network was $\tau(t) = [t^0 \sim \mathcal{U}(0,1), \ldots, t^M \sim \mathcal{U}(0,1)]$. For fairness, we used the same score network and reverse process sampler as was used for our MLD version with multi-time training; we call this variant Multi-modal Latent Diffusion UniDiffuser (MLD uni).

A.1.5 Technical Details

Sampling Schedule

We used the sampling schedule proposed in (Lugmayr et al., 2022), which has been shown to improve the coherence of conditional and joint generation. We used the best parameters suggested by the authors: $N = 250$ time steps applied $r = 10$ resampling times with jump size $j = 10$. For readability, in Algorithm 5 and Algorithm 4 we present pseudo-code with a linear sampling schedule which can be easily adapted to any other schedule.

Training the Score Network

Inspired by the architecture from (Dupont et al., 2022), we use simple Residual MLP blocks with skip connections as our score network (see Figure A.2). We fix the **width** and **number of blocks** proportionally to the number of the modalities and the latent space size. As in (Song and Ermon, 2020), we use the Exponential moving average (EMA) of the model parameters with a momentum parameter $m = 0.999$.

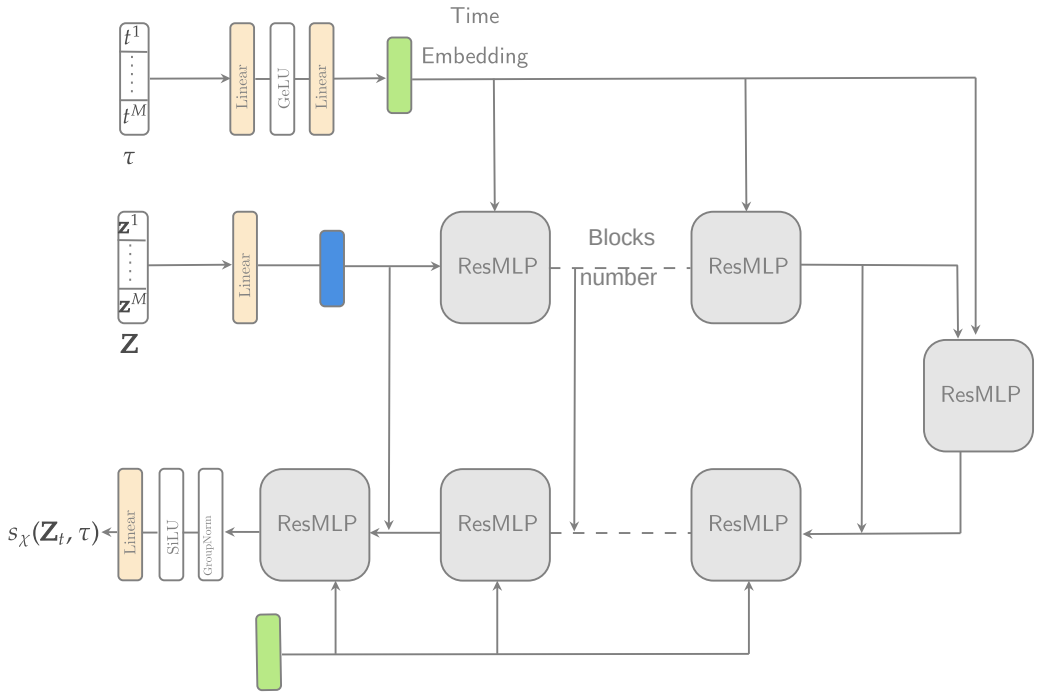


Figure A.2: Score network s_χ architecture used in our MLD implementation. The residual MLP block architecture is shown in Figure A.3.

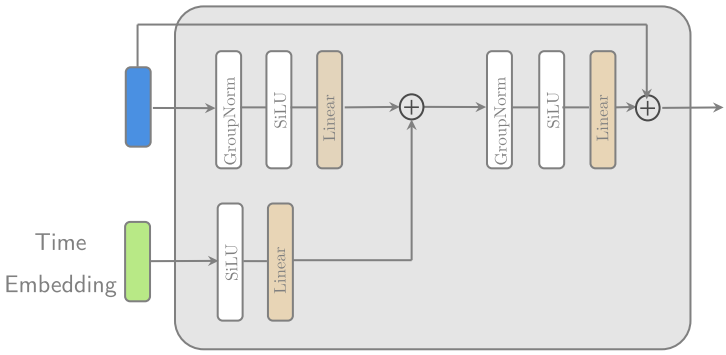


Figure A.3: Architecture of the ResMLP block.

A.2 MLD Ablation Study

In this section, we compare MLD with two variants presented in Appendix A.1: MLD in-paint, a naive approach without our proposed *multi-time masked* SDE, and MLD uni, a variant of our method using the same training scheme from (Bao et al., 2023). In addition, we analyze the effect of the randomization parameter d on the performance of MLD through an ablation study.

A.2.1 MLD and Its Variants

Table A.1 summarizes the different approaches adopted in each variant. All the considered models share the same deterministic autoencoders trained during the first stage.

For fairness, our evaluation was carried out using the same configuration and code basis as MLD. This included the autoencoder architectures and latent space size (similar to § 3.5). The same score network (Figure A.2) was used across experiments, with MLD in-paint using the same architecture with one time dimension instead of the multi-time vector. In all the variants, joint and conditional generation were conducted using the same reverse sampling schedule described in Appendix A.1.

Table A.1: Ablation study of MLD and its variants.

Model	Multi-Time Diffusion	Training	Conditional and Joint Generation
MLD in-paint	×	Eq. (2.28)	Algorithm 5
MLD uni	✓	(Bao et al., 2023)	Algorithm 4
MLD	✓	Algorithm 3	Algorithm 4

In certain cases, the MLD variants were able to match the joint generation performance

of MLD; however, overall they were less efficient and had noticeable weaknesses. MLD in-paint underperforms on conditional generation when considering relatively complex modalities, while MLD uni is not able to leverage the presence of multiple modalities to improve cross-generation, especially for datasets with a large number of modalities. On the other hand, MLD is able to overcome these limitations.

A.2.2 MNIST-SVHN

In Table A.2, MLD achieves the best results and dominates cross=generation performance. It can be observed that MLD in-paint lacks coherence for SVHN-to-MNIST conditional generation, a result we expected based on our analysis of the experiment in Figure A.1. MLD uni, despite the use of a multi-time diffusion process, underperforms our method, which indicates the effectiveness of our masked diffusion process in learning the conditional score network. Because all of the models used the same deterministic autoencoders, their observed generative quality performances are relatively similar (see Figure A.4 for qualitative results).

Table A.2: Generation coherence and quality for MNIST-SVHN (M stands for MNIST and S for SVHN). The generation quality is measured in terms of FMD for MNIST and FID for SVHN. Bold and underlined numbers indicate the best and second best scores respectively.

Models	Coherence (%↑)			Quality (↓)			
	Joint	M → S	S → M	Joint (M)	Joint (S)	M → S	S → M
MLD-Inpaint	**85.53** ± 0.22	<u>81.76</u> ± 0.23	63.28 ± 1.16	**3.85** ± 0.02	60.86 ± 1.27	59.86 ± 1.18	**3.55** ± 0.11
MLD-Uni	82.19 ± 0.97	79.31 ± 1.21	<u>72.78</u> ± 1.81	4.1 ± 0.17	57.41 ± 1.43	<u>57.84</u> ± 1.57	4.84 ± 0.28
MLD	<u>85.22</u> ± 0.5	**83.79** ± 0.62	**79.13** ± 0.38	<u>3.93</u> ± 0.12	**56.36** ± 1.63	**57.2** ± 1.47	<u>3.67</u> ± 0.14

MLD in-paint MLD uni **MLD**

Figure A.4: Qualitative results for **MNIST-SVHN**. For each model, we report MNIST-to-SVHN conditional generation on the left and SVHN-to-MNIST conditional generation on the right.

A.2.3 MHD

Table A.3 shows the performance results for the MHD dataset in terms of generative coherence. MLD achieves the best joint generation coherence, and, dominates the cross-generation

coherence results along with MLD uni. MLD in-paint shows a lack of coherence when conditioning on the sound modality alone, which is a predictable result, as this is a more difficult configuration because the sound modality is loosely correlated to other modalities. It can be observed that MLD in-paint performs worse than the two other alternatives when conditioned on the trajectory modality, which is the smallest modality in terms of latent size. This indicates another limitation of the naive approach regarding coherent generation when handling different latent spaces sizes, a weakness that our MLD method overcomes. Table A.4 presents the qualitative generative performance results, which are homogeneous across the variants, with MLD achieving either the best or second-best performance.

Table A.3: Generation coherence (%↑) for MHD (higher is better). The line above refers to the generated modality, while the subset of observed modalities is presented below. Bold and underlined numbers indicate the best and second best scores respectively.

Models	Joint	I (Image)			T (Trajectory)			S (Sound)		
		T	S	T,S	I	S	I,S	I	T	I,T
MLD-Inpaint	$96.88_{\pm 0.35}$	$63.9_{\pm 1.7}$	$56.52_{\pm 1.89}$	$95.83_{\pm 0.48}$	$\underline{99.58}_{\pm 0.1}$	$56.51_{\pm 1.89}$	$\underline{99.89}_{\pm 0.04}$	$95.81_{\pm 0.25}$	$56.51_{\pm 1.89}$	$96.38_{\pm 0.35}$
MLD-Uni	$\underline{97.69}_{\pm 0.26}$	$\mathbf{99.91}_{\pm 0.04}$	$\mathbf{89.87}_{\pm 0.38}$	$\mathbf{99.92}_{\pm 0.04}$	$\mathbf{99.68}_{\pm 0.1}$	$\mathbf{89.78}_{\pm 0.45}$	$99.38_{\pm 0.31}$	$\underline{97.54}_{\pm 0.2}$	$\underline{97.65}_{\pm 0.41}$	$\underline{97.79}_{\pm 0.41}$
MLD	$\mathbf{98.34}_{\pm 0.22}$	$99.45_{\pm 0.09}$	$\underline{88.91}_{\pm 0.54}$	$\underline{99.88}_{\pm 0.04}$	$\underline{99.58}_{\pm 0.03}$	$\underline{88.92}_{\pm 0.53}$	$\mathbf{99.91}_{\pm 0.02}$	$\mathbf{97.63}_{\pm 0.14}$	$\mathbf{97.7}_{\pm 0.34}$	$\mathbf{98.01}_{\pm 0.21}$

Table A.4: Generation quality for MHD. The metrics reported are FMD for the image and trajectory modalities and FAD for the sound modality (lower is better). Bold and underlined numbers indicate the best and second best scores respectively.

Models	I (Image)				T (Trajectory)				S (Sound)			
	Joint	T	S	T,S	Joint	I	S	I,S	Joint	I	T	I,T
MLD-Inpaint	$5.35_{\pm 1.35}$	$6.23_{\pm 1.13}$	$\underline{4.76}_{\pm 0.68}$	$3.53_{\pm 0.36}$	$\mathbf{1.59}_{\pm 0.12}$	$\mathbf{0.6}_{\pm 0.05}$	$\mathbf{1.81}_{\pm 0.13}$	$\mathbf{0.54}_{\pm 0.06}$	$2.41_{\pm 0.07}$	$2.5_{\pm 0.04}$	$2.52_{\pm 0.02}$	$2.49_{\pm 0.05}$
MLD-Uni	$\mathbf{7.91}_{\pm 2.2}$	$\mathbf{1.65}_{\pm 0.33}$	$6.29_{\pm 1.38}$	$\underline{3.06}_{\pm 0.54}$	$\underline{2.53}_{\pm 0.5}$	$1.18_{\pm 0.26}$	$3.18_{\pm 0.77}$	$2.84_{\pm 1.14}$	$\mathbf{2.11}_{\pm 0.08}$	$\mathbf{2.25}_{\pm 0.05}$	$\mathbf{2.1}_{\pm 0.0}$	$\mathbf{2.15}_{\pm 0.01}$
MLD	$\underline{7.98}_{\pm 1.41}$	$\underline{1.7}_{\pm 0.14}$	$\mathbf{4.54}_{\pm 0.45}$	$\mathbf{1.84}_{\pm 0.27}$	$3.18_{\pm 0.18}$	$\underline{0.83}_{\pm 0.03}$	$\underline{2.07}_{\pm 0.26}$	$\underline{0.6}_{\pm 0.05}$	$\underline{2.39}_{\pm 0.1}$	$\underline{2.31}_{\pm 0.07}$	$\underline{2.33}_{\pm 0.11}$	$\underline{2.29}_{\pm 0.06}$

A.2.4 POLYMNIST

In Figure A.5, we note the superiority of MLD in both generative coherence and quality. MLD-Uni is not able to leverage the presence of a large number of modalities in conditional generation coherence. Interestingly, an increase in the number of input modalities negatively impacts the performance of MLD uni.

A.2.5 CUB

Figure A.6 shows the qualitative results for caption-to-image conditional generation. All of the variants are based on the same first-stage autoencoders, and the generative performance is comparable in terms of quality.

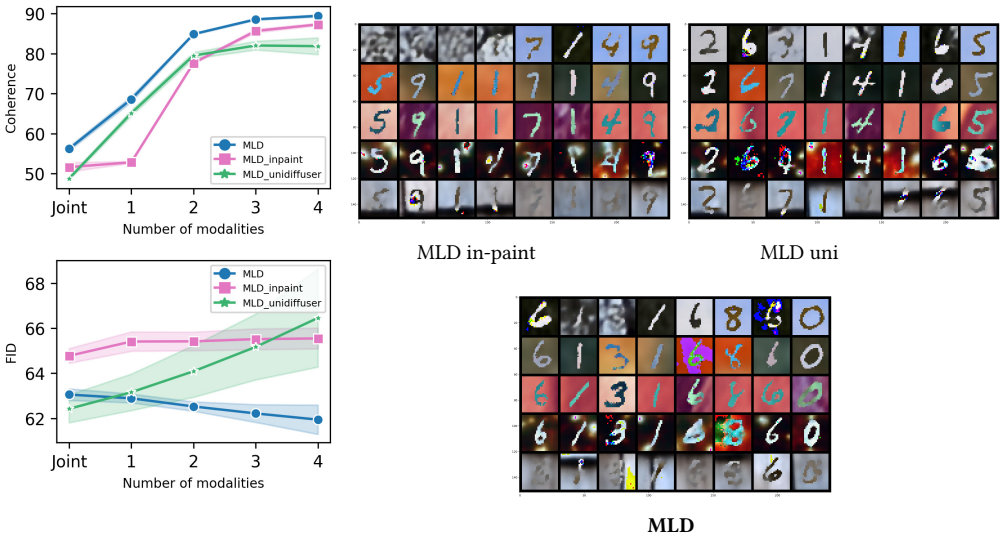


Figure A.5: Results for the **POLYMNIST** dataset. (**Left**): a comparison of the generative coherence (% ↑) and quality in terms of FID (↓)) as a function of the number of modality inputs. We report the average performance following the leave-one-out strategy (see Appendix A.3). (**Right**): qualitative results for joint generation of the five modalities.

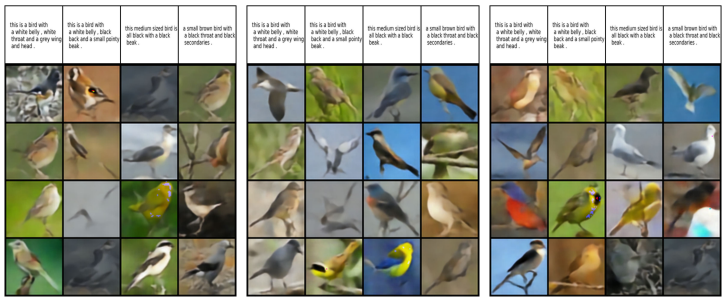


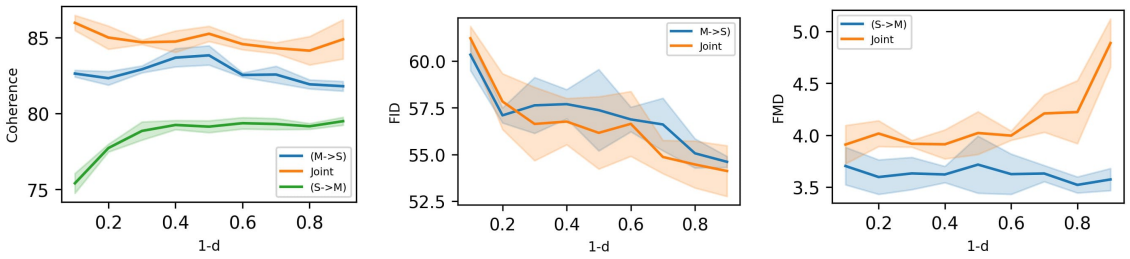
Figure A.6: Qualitative results on the **CUB** dataset. Captions were used as the condition to generate the bird images.

A.2.6 Randomization d-Ablation Study

The d parameter controls the randomization of the *multi-time masked diffusion process* during training in Algorithm 3. With probability d, the concatenated latent space corresponding to all the modalities is diffused at the same time. With probability $(1 - d)$, a portion of the latent space corresponding to a random subset of the modalities is not diffused and is frozen during the training step. To study the d parameter and its effect on the performance of our MLD model, we used $d \in \{0.1, \ldots, 0.9\}$. Figure A.7 shows the results of the d-ablation study on the **MNIST-SVHN** dataset. We report the performance results averaged over five independent seeds as a function of the probability $(1 - d)$: **Left** shows the conditional and joint coherence for the **MNIST-SVHN** dataset; **Middle** shows the quality performance in terms of FID for SVHN generation; and **Right** shows the quality performance in terms of

FMD for MNIST generation.

It can be observed that higher values for $1-d$, indicating a greater probability of applying *multi-time masked diffusion*, improve the coherence of SVHN-to-MNIST conditional generation. This confirms that masked multi-time training enables better conditional generation. Overall, on the **MNIST-SVHN** dataset, MLD shows weak sensibility to the d parameter whenever the value of $d \in [0.2, 0.7]$.



(a) MNIST-SVHN: Coherence (%↑) **(b)** SVHN: FID (↓) **(c)** MNIST: FMD(↓)

Figure A.7: Results of the ablation study for the randomization parameter d on the **MNIST-SVHN** dataset.

A.3 Datasets and Evaluation Protocol

A.3.1 Dataset Description

MNIST-SVHN (Shi et al., 2019) is constructed using pairs of MNIST and SVHN sharing the same digit class (see Figure A.8a). Each instance of a digit class (in either dataset) is randomly paired with 20 instances of the same digit class from the other dataset. SVHN modality samples are obtained from house numbers in Google Street View images, and are characterized by a variety of colors, shapes, and angles. A high number of SVHN samples are noisy, and can contain different digits within the same sample due to the imperfect cropping of the original full house number image. One challenge of this dataset for multimodal generative models is to learn to extract digit number and reconstruct a coherent MNIST modality.

MHD (Vasco et al., 2022) is composed of three modalities: synthetically generated images and motion trajectories of handwritten digits associated with their speech sounds. The images are gray-scale $1 \times 28 \times 28$, and the handwriting trajectories are represented by a 1×200 vector. The spoken digits sounds are $1s$ audio clips processed as Mel-Spectrograms, and are constructed with a hopping window of 512 ms with 128 Mel Bins, resulting in a $1 \times 128 \times 32$ representation. This benchmark is the closest to a real-world scenario involving multimodal sensors because of the presence of three completely different modalities, with

the audio modality representing a complex data type. Therefore, similar to SVHN, the conditional generation of sound to coherent images or trajectories represents a challenging use case.

POLYMNIST (Sutter, Daunhawer, and Vogt, 2021) is a version of the MNIST dataset extended to five modalities. Each modality is constructed using a random set of MNIST digits with an overlay over a random crop from a modality-specific three-channel image background. This synthetic generated dataset allows for the evaluating the scalability of multimodal generative models to large number of modalities. Although this dataset is composed only of images, the different textures of different modality-specific backgrounds results in differing levels of difficulty. In Figure A.8c, the digits are more difficult to distinguish in modalities 1 and 5 than in the other modalities.

CUB (Shi et al., 2019) is comprised of bird images and associated text captions. In (Shi et al., 2019), a simplified version based on precomputed ResNet-features was used. Following (Daunhawer et al., 2022), we conducted all of our experiments on the real image data instead. Each image from the 11,788 photos of birds from Caltech-Birds (Wah et al., 2011) was resized to a $3 \times 64 \times 64$ image and coupled with ten textual descriptions of the respective bird (see Figure A.8d).

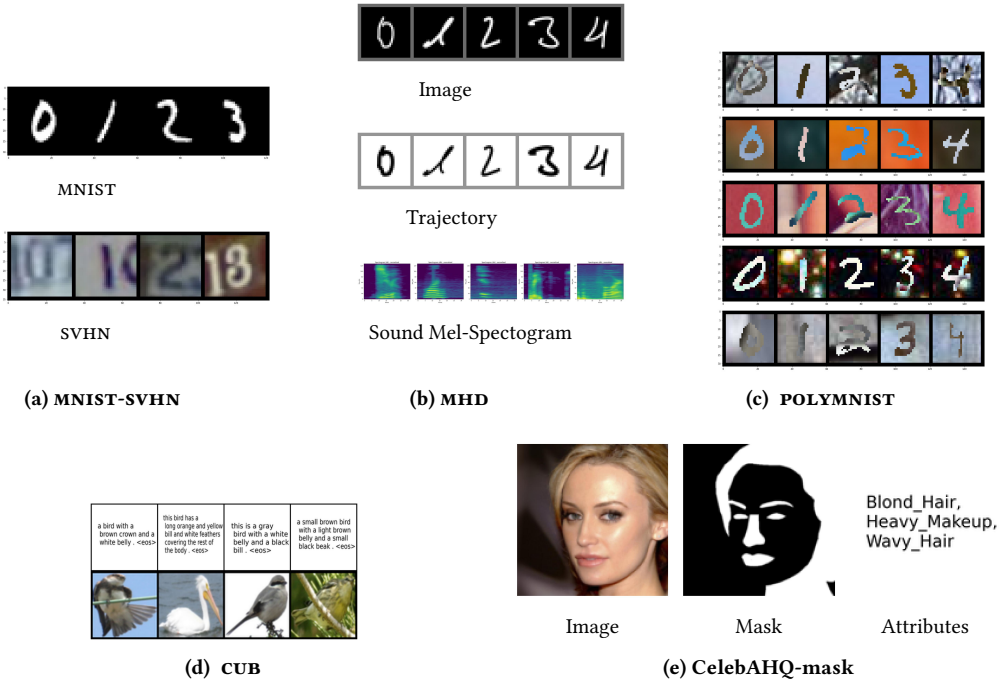
CelebAHQ-mask consists of three modalities: face images, each with a segmentation mask and attributes. We took into account 18 out of 40 attributes from the original dataset and resized the images to 128×128 resolution, as was done in (Wu and Goodman, 2018; Wesego and Rooshenas, 2023).

A.3.2 Evaluation Metrics

The multimodal generative models were evaluated in terms of their generative coherence and quality.

Generation Coherence

We measured *coherence* by verifying that generated data for both joint and conditional generation shared the same information across modalities. Following (Shi et al., 2019; Sutter, Daunhawer, and Vogt, 2021; Hwang et al., 2021; Vasco et al., 2022; Daunhawer et al., 2022), we considered the class label of the modalities as the shared information and used pretrained classifiers to extract the label information from the generated samples and compare it across modalities.



(a) MNIST-SVHN (b) MHD (c) POLYMNIST

(d) CUB (e) CelebAHQ-mask

Figure A.8: Illustrative example of the datasets used for evaluation.

For **MNIST-SVHN**, **MHD**, and **POLYMNIST**, the shared semantic information is the digit class number. Single-modality classifiers are trained to classify the digit number of a given modality sample. To compute the conditional generation of a modality m with a subset of conditioning modalities A, the conditional generated sample $\hat{\mathbf{X}}^m$ is fed to the modality-specific pretrained classifier . The predicted label class is compared to the ground truth label, which is the label of the modalities in subset A. For N samples, the matching rate average establishes the coherence. For all the experiments, N was equal to the length of the test set.

The **joint generation coherence** was measured by feeding the generated samples of each modality to their specific trained classifier. The rate at which all classifiers output the same predicted digit label for N generations was considered the joint generation coherence.

The leave-one-out coherence is the conditional generation coherence using all possible subsets excluding the generated modality (*Coherence*$(\hat{\mathbf{X}}^m|\mathbf{X}^A)$ with $A = \{1, .., M\} \setminus m$). Due to the large number of modalities in **POLYMNIST**, similar to (Sutter, Daunhawer, and Vogt, 2021; Hwang et al., 2021; Daunhawer et al., 2022), we computed the average **leave-one-out coherence** conditional coherence as a function of the subset size of the input modalities. Due to the unavailability of labels in the **CUB** dataset, we used Clip-s (Hessel et al., 2021), a state-of-the art metric for image captioning evaluation.

Generation Quality

For each modality, we considered the following metrics:

- **RGB Images**: FID (Heusel et al., 2017) is the state-of-the-art standard metric for evaluating the image generation quality of generative models.
- **Audio**: FAD (Kilgour et al., 2019) is a state-of-the-art standard metric for the evaluation of audio generation. FAD performs well in terms of robustness against noise, and is consistent with human judgments (Vinay and Lerch, 2022). Similar to FID, the Fréchet distance is computed, except that VGGish (audio classifer model) embeddings are used instead.
- **Other modalities**: For other modality types, we derived the FMD (Fréchet Modality Distance), a similar metric to FID and FAD. We computed the **Fréchet distance** between the statistics retrieved from the activations of the modality-specific pretrained classifiers used for coherence evaluation. FMD was used to evaluate the generative quality of the MNIST modality on the **MNIST-SVHN** dataset and the image and trajectory modalities on the **MHD** dataset.

For conditional generation, we computed the quality metric (FID,FAD, or FMD) using the conditionally generated modality and the real data. For joint generation, we used the randomly generated modality and randomly selected the same number of samples from the real data. For **CUB**, we used $10,000$ samples to evaluate the generation quality in terms of FID. In the remaining experiments, we used 5000 samples to evaluate the performance in terms of FID, FAD, or FMD.

A.4 Implementation Details

In this section, we report the implementation details for each benchmark. We used the same unified code base for all the baselines, relying on the *PyTorch* framework. The VAE implementation was adapted from the official code whenever available (MVAE, MMVAE and MoPoE, as in [1], Multi-view Total Correlation Autoencoder (MVTCAE) [2] , and NEXUS (NEXUS) [3]. To ensure fairness, MLD and all VAE-based models used the same autoencoder architecture. We used the best hyperparameters suggested by the authors. Across all the datasets, we used the *Adam optimizer* (Kingma and Ba, 2014) for training.

[1] https://github.com/thomassutter/MoPoE
[2] https://github.com/gr8joo/MVTCAE
[3] https://github.com/miguelsvasco/nexus_pytorch

A.4.1 MLD

MLD uses the same autoencoder architecture as for VAE-based models, except that the latter are deterministic autoencoders. The autoencoders were trained using the same reconstruction loss term as for the VAE-based models. Tables A.5 and A.6 summarize the hyper-parameters used during the two phases of MLD training. Note that data augmentation was necessary for the image modality in the CUB dataset in order to overcome overfitting when training the deterministic autoencoder. For this, we used *TrivialAugmentWide* from the Torchvision library.

Table A.5: MLD: hyperparameters used for the deterministic autoencoders.

Dataset	Modality	Latent Space	Batch Size	Lr	Epochs	Weight Decay
MNIST-SVHN	MNIST SVHN	16 64	128	1×10^{-3}	150	
MHD	Image Trajectory Sound	64 16 128	64	1×10^{-3}	500	
POLYMNIST	All modalities	160	128	1×10^{-3}	300	
CUB	Caption Image	32 64	128	1×10^{-3} 1×10^{-4}	500 300	1×10^{-6}
CelebAMask-HQ	Image Mask Attributes	256 128 32	64	1×10^{-3}	200	

Table A.6: MLD: score network hyperparameters.

Dataset	d	Blocks	Width	Time Embed	Batch Size	Lr	Epochs
MNIST-SVHN	0.5	2	512	256	128		150
MHD	0.3	2	1024	512	128		3000
POLYMNIST	0.5	2	1536	512	256	1×10^{-4}	3000
CUB	0.7	2	1024	512	64		3000
CelebAMask-HQ	0.5	2	1536	512	64		3000

A.4.2 VAE-Based Models

For **MNIST-SVHN**, we followed (Sutter, Daunhawer, and Vogt, 2021; Shi et al., 2019) and used the same autoencoder architecture and pretrained classifier. The latent space size was

set to 20, $\beta = 5.0$. For MVTCAE $\alpha = \frac{5}{6}$. For both modalities, the likelihood was estimated using the Laplace distribution. For NEXUS, we used the same modality latent space size as in MLD, the joint NEXUS latent space was set to 20, $\beta_i = 1.0$, and $\beta_c = 5.0$. We trained all the VAE-models for 150 epochs with a batch size of 256 and learning rate of 1×10^{-3}.

For **MHD**, we reused the autoencoder architecture and pretrained classifier from (Vasco et al., 2022). We adopted the hyperparameters from (Vasco et al., 2022) to train the NEXUS model with the same settings while discarding the label modality. For the remaining VAE-based models, the latent space size was set to 128, $\beta = 1.0$, and $\alpha = \frac{5}{6}$ for MVTCAE. For all the modalities, Mean square error (MSE) was used to compute the reconstruction loss, similar to (Vasco et al., 2022). The models were trained for 600 epochs with a batch size of 128 and learning rate of 1×10^{-3}.

For **POLYMNIST**, we used the same autoencoder architecture and pretrained classifier used by (Sutter, Daunhawer, and Vogt, 2021; Hwang et al., 2021). We set the latent space size to 512, $\beta = 2.5$, and $\alpha = \frac{5}{6}$ for MVTCAE. For all the modalities, the likelihood was estimated using the Laplace distribution. For NEXUS, we used the same modality latent space size as in MLD, the joint NEXUS latent space was 64, $\beta_i = 1.0$, and $\beta_c = 2.5$. We trained all the models for 300 epochs with a batch size of 256 and learning rate of 1×10^{-3}.

For **CUB**, we used the same autoencoder architecture and implementation settings as in (Daunhawer et al., 2022). The Laplace and one-hot categorical distributions were used to estimate the likelihoods of the image and caption modalities, respectively. The latent space size was set to 64, $\beta = 9.0$ for MVAE, MVTCAE, and MoPoE, and $\beta = 1$ for MMVAE. We set $\alpha = \frac{5}{6}$ for MVTCAE. For NEXUS, we used the same modality latent space sizes as in MLD, the joint NEXUS latent space was set to 64, $\beta_i = 1.0$, and $\beta_c = 1$. We trained all the models for 150 epochs with a batch size of 64. We used a learning rate of $5e-4$ for MVAE, MVTCAE, and MoPoE and 1×10^{-3} for the remaining models.

Finally, we note that in the official implementation of (Sutter, Daunhawer, and Vogt, 2021) and (Hwang et al., 2021) on the **POLYMNIST** and **MNIST-SVHN** datasets, the classifiers were used for evaluation with dropout. In our implementation, we made sure to deactivate dropout during the evaluation step.

For **CelebAMask-HQ**, in our MLD experiments we used deterministic autoencoders instead of variational autoencoders (Lee et al., 2019).

A.4.3 MLD with Powerful Autoencoder

Here, we provide more detail about the CUB experiment using a more powerful autoencoder, denoted MLD* in Figure 3.5. We used an architecture similar to (Rombach et al., 2022) adapted to (64×64) resolution images. We modified the autoencoder architecture to be deterministic and trained the model with a simple mean square error loss. We kept the same configuration as the CUB experiment described in the previous experiment on the same dataset, including the text autoencoder, score network, and hyperparameters. We performed further experiments with the same settings on 128×128 resolution images. We include the qualitative results in Fig. A.21.

	Day	Night
Train	114251	16620
Test	3840	524

Table A.7: nuScenesdataset sample size after preprocessing.

A.4.4 MLD for improved Night Vision

The model is initialized with weights pre-trained on the ImageNet dataset (Deng et al., 2009). We extend the model by adding additional tokens for each modality and fine-tuned using multimodal sensor data. The fine-tuning process is conducted on $512x512$ image resolution for $200,000$ iterations, with a batch size of 24 and a learning rate of $1e^{-4}$. We employ the randomized approach described in § 3.4.3 to learn the reconstruction of the camera modalities given the availability of either LiDar, RaDar, or both. In all the experiments, we used classifier free guidance (Ho and Salimans, 2022) with guidance set to 2.0 and DDIM (Song, Meng, and Ermon, 2020) sampler with 50 steps.

A.4.5 Computation Resources

In our experiments, we used four A100 GPUs for a total of roughly four months of experiments.

A.5 Additional Results

In this section, we report detailed results for all of our experiments, including the standard deviation and additional qualitative samples for all the datasets and all the methods we

compared in our work.

A.5.1 MNIST-SVHN

Self-Reconstruction

In Table A.8, we report the results on *self-coherence*, which we use to support the arguments from § 3.3. This metric is used to measure the loss of information due to latent collapse by showing the ability of all competing models to reconstruct an arbitrary modality given the same modality or a set thereof as an input. For our MLD model, self-reconstruction is done without using the diffusion model component; the modality is encoded using its deterministic encoder, and the decoder is fed the latent space to obtain the reconstruction.

We observe that the VAE-based models fail to reconstruct SVHN given SVHN. This is especially visible for the models based on the product-of-experts approach (MVAE and MVTCAE). In MLD, the deterministic autoencoders do not suffer from such weakness, and achieve the best overall performance.

Figure A.9 shows the qualitative self-generation results. We remark that the digits in certain samples generated using VAE-based models differ from those in the input sample (for example, generation of the MNIST digit 3 in the case of MVAE and the SVHN digit 2 in the case of MVTCAE), indicating information loss due to latent collapse.

Table A.8: Self-generation coherence and quality for **MNIST-SVHN** (M: MNIST, S: SVHN). The generation quality is measured in terms of FMD for MNIST and FID for SVHN. Bold and underlined numbers indicate the best and second best scores respectively.

Models	Coherence (%↑)				Quality (↓)			
	M → M	**M,S → M**	**S → S**	**M,S → S**	**M → M**	**M,S → M**	**S → S**	**M,S → S**
MVAE	$86.92_{\pm 0.8}$	$88.03_{\pm 0.78}$	$40.62_{\pm 0.99}$	$68.01_{\pm 1.29}$	$10.75_{\pm 1.04}$	$10.79_{\pm 1.02}$	$60.22_{\pm 1.01}$	$59.0_{\pm 0.6}$
MMVAE	$87.22_{\pm 1.87}$	$77.35_{\pm 4.19}$	$67.31_{\pm 6.93}$	$39.44_{\pm 3.43}$	$12.15_{\pm 1.25}$	$20.24_{\pm 1.04}$	$58.1_{\pm 3.14}$	$171.42_{\pm 4.55}$
MoPoE	$89.95_{\pm 0.84}$	$91.71_{\pm 0.77}$	$67.26_{\pm 0.8}$	$\underline{83.58}_{\pm 0.44}$	$9.39_{\pm 0.76}$	$10.1_{\pm 0.73}$	$53.19_{\pm 1.06}$	$57.34_{\pm 1.35}$
NEXUS	$92.63_{\pm 0.45}$	$93.59_{\pm 0.4}$	$\underline{68.31}_{\pm 0.46}$	$83.13_{\pm 0.58}$	$4.92_{\pm 0.61}$	$5.16_{\pm 0.59}$	$85.67_{\pm 2.74}$	$97.86_{\pm 2.86}$
MVTCAE	$\underline{94.33}_{\pm 0.18}$	$\underline{95.18}_{\pm 0.19}$	$47.47_{\pm 0.76}$	$\mathbf{86.6}_{\pm 0.23}$	$\underline{4.67}_{\pm 0.35}$	$\underline{4.94}_{\pm 0.37}$	$\underline{52.29}_{\pm 1.17}$	$\underline{53.55}_{\pm 1.19}$
MLD	$\mathbf{96.73}_{\pm 0.0}$	$\mathbf{96.73}\pm 0.0$	$\mathbf{82.19}_{\pm 0.0}$	$82.19_{\pm 0.0}$	$\mathbf{2.25}_{\pm 0.03}$	$\mathbf{2.25}\pm 0.03$	$\mathbf{48.47}_{\pm 0.63}$	$\mathbf{48.47}_{\pm 0.63}$

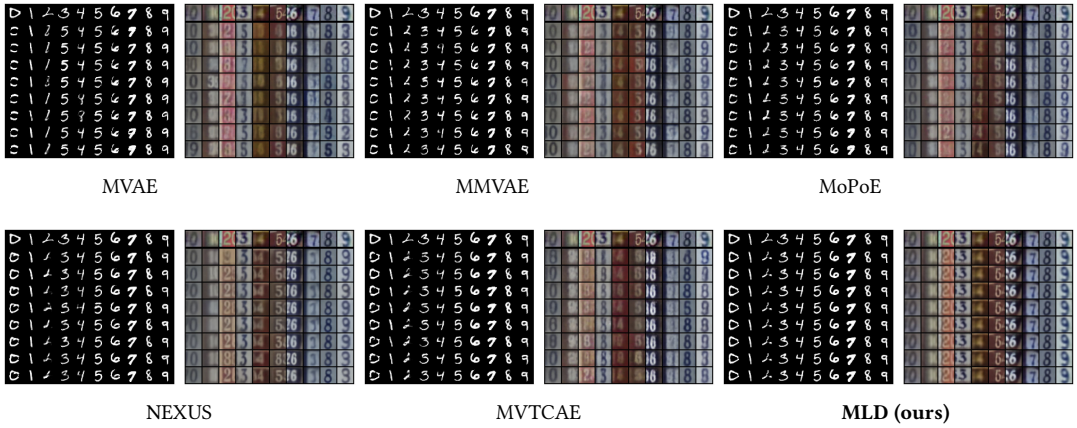


Figure A.9: Self-generation qualitative results for **MNIST-SVHN**. For each model, we report MNIST-to-MNIST conditional generation on the left and SVHN-to-SVHN conditional generation on the right.

Detailed Results

Table A.9: Generative coherence for **MNIST-SVHN**. We report the detailed version of Table 3.1 with the standard deviation for five independent runs with different seeds. Bold and underlined numbers indicate the best and second best scores respectively.

Models	Coherence (%↑)			Quality (↓)			
	Joint	**M → S**	**S → M**	**Joint(M)**	**Joint(S)**	**M → S**	**S → M**
MVAE	$38.19_{\pm 2.27}$	$48.21_{\pm 2.56}$	$28.57_{\pm 1.46}$	$13.34_{\pm 0.93}$	$68.0_{\pm 0.99}$	$68.9_{\pm 1.84}$	$13.66_{\pm 0.95}$
MMVAE	$37.82_{\pm 1.19}$	$11.72_{\pm 0.33}$	$67.55_{\pm 9.22}$	$25.89_{\pm 0.46}$	$146.82_{\pm 4.76}$	$393.33_{\pm 4.86}$	$53.37_{\pm 1.87}$
MoPoE	$39.93_{\pm 1.54}$	$12.27_{\pm 0.68}$	$68.82_{\pm 0.39}$	$20.11_{\pm 0.96}$	$129.2_{\pm 6.33}$	$373.73_{\pm 26.42}$	$43.34_{\pm 1.72}$
NEXUS	$40.0_{\pm 2.74}$	$16.68_{\pm 5.93}$	$70.67_{\pm 0.77}$	$13.84_{\pm 1.41}$	$98.13_{\pm 5.9}$	$281.28_{\pm 16.07}$	$53.41_{\pm 1.54}$
MVTCAE	$48.78_{\pm 1}$	$\underline{81.97}_{\pm 0.32}$	$49.78_{\pm 0.88}$	$12.98_{\pm 0.68}$	$\mathbf{52.92}_{\pm 1.39}$	$69.48_{\pm 1.64}$	$13.55_{\pm 0.8}$
MMVAE+ (MMVAE+)	$17.64_{\pm 4.12}$	$13.23_{\pm 4.96}$	$29.69_{\pm 5.08}$	$26.60_{\pm 2.58}$	$121.77_{\pm 37.77}$	$240.90_{\pm 85.74}$	$35.11_{\pm 4.25}$
MMVAE+(K = 10)	$41.59_{\pm 4.89}$	$55.3_{\pm 9.89}$	$56.41_{\pm 5.37}$	$19.05_{\pm 1.10}$	$67.13_{\pm 4.58}$	$75.9_{\pm 12.91}$	$18.16_{\pm 2.20}$
MLD	$\underline{85.22}_{\pm 0.5}$	$\mathbf{83.79}_{\pm 0.62}$	$\mathbf{79.13}_{\pm 0.38}$	$\underline{3.93}_{\pm 0.12}$	$\underline{56.36}_{\pm 1.63}$	$\mathbf{57.2}_{\pm 1.47}$	$\underline{3.67}_{\pm 0.14}$

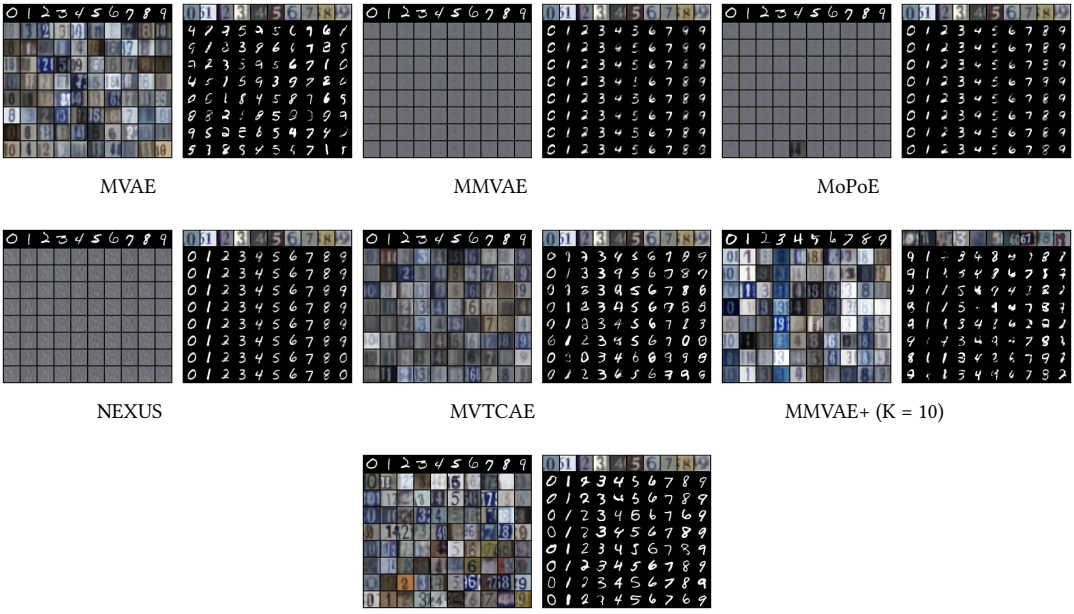


Figure A.10: Additional qualitative results for **MNIST-SVHN**. For each model, we report MNIST-to-SVHN conditional generation on the left and SVHN-to-MNIST conditional generation on the right.

A.5.2 MHD

Table A.10: Generative coherence for **MHD**. We report the detailed version of Table 3.2 with the standard deviation for five independent runs with different seeds. Bold and underlined numbers indicate the best and second best scores respectively.

Models	Joint	I (Image)			T (Trajectory)			S (Sound)		
		T	S	T,S	I	S	I,S	I	T	I,T
MVAE	$37.77_{\pm 3.32}$	$11.68_{\pm 0.35}$	$26.46_{\pm 1.84}$	$28.4_{\pm 1.47}$	$95.55_{\pm 1.39}$	$26.66_{\pm 1.72}$	$96.58_{\pm 1.06}$	$58.87_{\pm 4.89}$	$10.39_{\pm 0.42}$	$58.16_{\pm 5.24}$
MMVAE	$34.78_{\pm 0.83}$	$\mathbf{99.7}_{\pm 0.03}$	$69.69_{\pm 1.66}$	$84.74_{\pm 0.95}$	$\underline{99.3}_{\pm 0.07}$	$85.46_{\pm 1.57}$	$92.39_{\pm 0.95}$	$49.95_{\pm 0.79}$	$50.14_{\pm 0.89}$	$50.17_{\pm 0.99}$
MoPoE	$48.84_{\pm 0.36}$	$\underline{99.64}_{\pm 0.08}$	$68.67_{\pm 2.07}$	$\underline{99.69}_{\pm 0.04}$	$99.28_{\pm 0.08}$	$\underline{87.42}_{\pm 0.41}$	$99.35_{\pm 0.04}$	$50.73_{\pm 3.72}$	$51.5_{\pm 3.52}$	$56.97_{\pm 6.34}$
NEXUS	$26.56_{\pm 1.71}$	$94.58_{\pm 0.34}$	$\underline{83.1}_{\pm 0.74}$	$95.27_{\pm 0.52}$	$88.51_{\pm 0.64}$	$76.82_{\pm 3.63}$	$93.27_{\pm 0.91}$	$70.06_{\pm 2.83}$	$75.84_{\pm 2.53}$	$89.48_{\pm 3.24}$
MVTCAE	$42.28_{\pm 1.12}$	$99.54_{\pm 0.07}$	$72.05_{\pm 0.95}$	$99.63_{\pm 0.05}$	$99.22_{\pm 0.08}$	$72.03_{\pm 0.48}$	$\underline{99.39}_{\pm 0.02}$	$\underline{92.58}_{\pm 0.47}$	$\underline{93.07}_{\pm 0.36}$	$\underline{94.78}_{\pm 0.25}$
MMVAE+	$41.67_{\pm 2.3}$	$98.05_{\pm 0.19}$	$84.16_{\pm 0.57}$	$91.88_{\pm}$	$97.47_{\pm 0.89}$	$81.16_{\pm 2.24}$	$89.31_{\pm 1.54}$	$64.34_{\pm 4.46}$	$65.42_{\pm 5.42}$	$64.88_{\pm 4.93}$
MMVAE+(K = 10)	$42.60_{\pm 2.5}$	$99.44_{\pm 0.07}$	$\mathbf{89.75}_{\pm 0.75}$	$94.7_{\pm 0.72}$	$99.44_{\pm 0.18}$	$\mathbf{89.58}_{\pm 0.4}$	$95.01_{\pm 0.30}$	$87.15_{\pm 2.81}$	$87.99_{\pm 2.55}$	$87.57_{\pm 2.09}$
MLD	$\mathbf{98.34}_{\pm 0.22}$	$99.45_{\pm 0.09}$	$\underline{88.91}_{\pm 0.54}$	$\mathbf{99.88}_{\pm 0.04}$	$\mathbf{99.58}_{\pm 0.03}$	$\underline{88.92}_{\pm 0.53}$	$\mathbf{99.91}_{\pm 0.02}$	$\mathbf{97.63}_{\pm 0.14}$	$\mathbf{97.7}_{\pm 0.34}$	$\mathbf{98.01}_{\pm 0.21}$

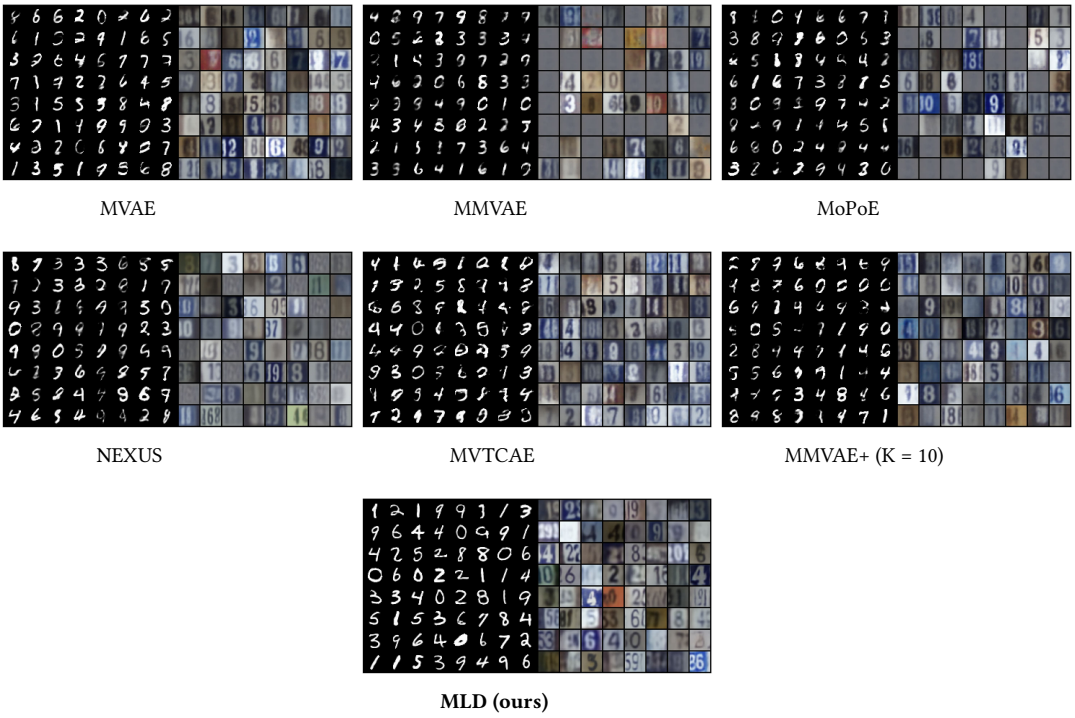


Figure A.11: Qualitative results for **MNIST-SVHN** joint generation.

Table A.11: Generative quality for **MHD**. We report the detailed version of Table 3.3 with the standard deviation for five independent runs with different seeds. Bold and underlined numbers indicate the best and second best scores respectively.

Models	I (Image)				T (Trajectory)				S (Sound)			
	Joint	T	S	T,S	Joint	I	S	I,S	Joint	I	T	I,T
MVAE	$\underline{94.9}_{\pm 7.37}$	$93.73_{\pm 5.44}$	$92.55_{\pm 7.37}$	$91.08_{\pm 10.24}$	$39.51_{\pm 6.04}$	$20.42_{\pm 4.42}$	$38.77_{\pm 6.29}$	$19.25_{\pm 4.26}$	$14.14_{\pm 0.25}$	$\underline{14.13}_{\pm 0.19}$	$14.08_{\pm 0.24}$	$14.17_{\pm 4.26}$
MMVAE	$224.01_{\pm 12.58}$	$22.6_{\pm 4.3}$	$789.12_{\pm 12.58}$	$170.41_{\pm 8.06}$	$16.52_{\pm 1.17}$	$\mathbf{0.5}_{\pm 0.05}$	$30.39_{\pm 1.38}$	$6.07_{\pm 0.37}$	$22.8_{\pm 0.39}$	$22.61_{\pm 0.75}$	$23.72_{\pm 0.86}$	$23.01_{\pm 0.67}$
MoPoE	$147.81_{\pm 10.37}$	$16.29_{\pm 0.85}$	$838.38_{\pm 10.84}$	$15.89_{\pm 1.96}$	$\underline{13.92}_{\pm 0.96}$	$\underline{0.52}_{\pm 0.12}$	$33.38_{\pm 1.14}$	$\mathbf{0.53}_{\pm 0.1}$	$18.53_{\pm 0.27}$	$24.11_{\pm 0.4}$	$24.1_{\pm 0.41}$	$23.93_{\pm 0.87}$
NEXUS	$281.76_{\pm 12.69}$	$116.65_{\pm 9.99}$	$282.34_{\pm 12.69}$	$117.24_{\pm 8.53}$	$18.59_{\pm 2.16}$	$6.67_{\pm 0.23}$	$33.01_{\pm 3.41}$	$7.54_{\pm 0.29}$	$\underline{13.99}_{\pm 0.9}$	$19.52_{\pm 0.14}$	$18.71_{\pm 0.24}$	$16.3_{\pm 0.59}$
MVTCAE	$121.85_{\pm 3.44}$	$\underline{5.34}_{\pm 0.33}$	$\underline{54.57}_{\pm 7.79}$	$\underline{3.16}_{\pm 0.26}$	$19.49_{\pm 0.67}$	$0.62_{\pm 0.1}$	$\underline{13.65}_{\pm 1.24}$	$0.75_{\pm 0.13}$	$15.88_{\pm 0.19}$	$14.22_{\pm 0.27}$	$\underline{14.02}_{\pm 0.14}$	$\underline{13.96}_{\pm 0.28}$
MMVAE+	$97.19_{\pm 12.37}$	$2.80_{\pm 0.42}$	$128.56_{\pm 4.47}$	$114.3_{\pm 11.4}$	$22.37_{\pm 1.87}$	$1.21_{\pm 0.22}$	$21.74_{\pm 3.49}$	$15.2_{\pm 1.15}$	$16.12_{\pm 0.40}$	$17.31_{\pm 0.62}$	$17.92_{\pm 0.19}$	$17.56_{\pm 0.48}$
MMVAE+(K = 10)	$85.98_{\pm 1.25}$	$1.83_{\pm 0.26}$	$70.72_{\pm 1.76}$	$62.43_{\pm 3.4}$	$21.10_{\pm 1.25}$	$1.38_{\pm 0.34}$	$8.52_{\pm 0.79}$	$7.22_{\pm 1.6}$	$14.58_{\pm 0.47}$	$14.33_{\pm 0.51}$	$14.34_{\pm 0.42}$	$14.32_{\pm 0.6}$
MLD (ours)	$\mathbf{7.98}_{\pm 1.41}$	$\mathbf{1.7}_{\pm 0.14}$	$\mathbf{4.54}_{\pm 0.45}$	$\mathbf{1.84}_{\pm 0.27}$	$\mathbf{3.18}_{\pm 0.18}$	$0.83_{\pm 0.03}$	$\mathbf{2.07}_{\pm 0.26}$	$\underline{0.6}_{\pm 0.05}$	$\mathbf{2.39}_{\pm 0.1}$	$\mathbf{2.31}_{\pm 0.07}$	$\mathbf{2.33}_{\pm 0.11}$	$\mathbf{2.29}_{\pm 0.06}$

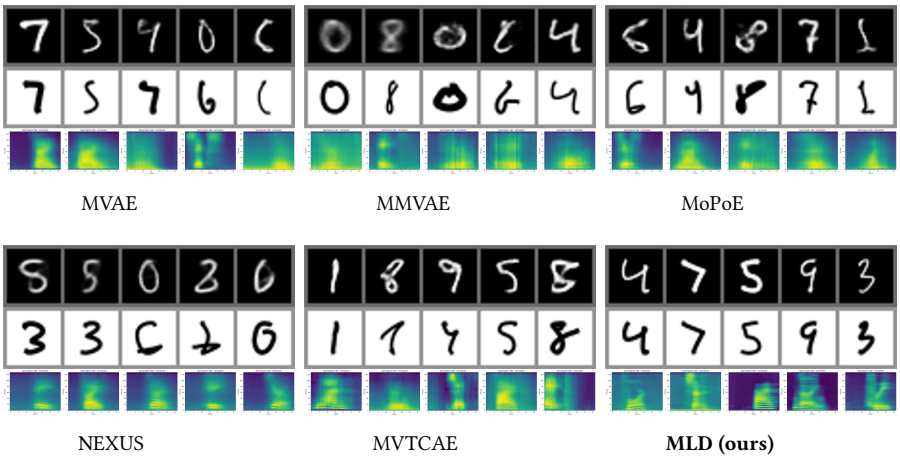


Figure A.12: Joint generation qualitative results for **MHD**. The three modalities were randomly generated simultaneously. **Top row**: image; **Middle row**: trajectory vector converted into image; **Bottom row**: sound mel-spectrogram).

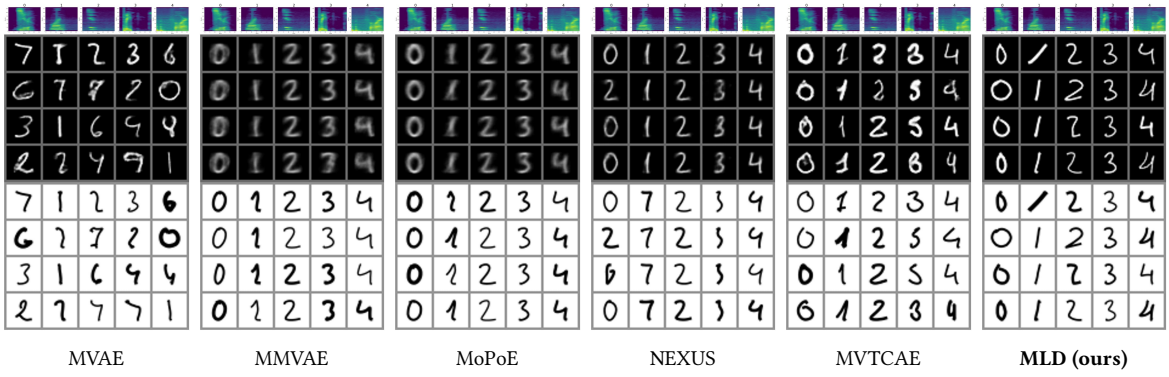


Figure A.13: Sound-to-image and trajectory conditional generation qualitative results for **MHD**. For each model, the **Top row** reports the sound mel-spectrograms of the digits {0,1,2,3,4} from left to right and the **Lower rows** report the generated image and trajectory samples.

A.5.3 POLYMNIST

Table A.12: Generation coherence (%) for **POLYMNIST** (higher is better) used for the plots in Figure 3.3 and Figure A.5. We report the average *leave-one-out coherence* as a function of the number of observed modalities. *Joint* refers to random generation of the five modalities simultaneously. Bold and underlined numbers indicate the best and second best scores respectively.

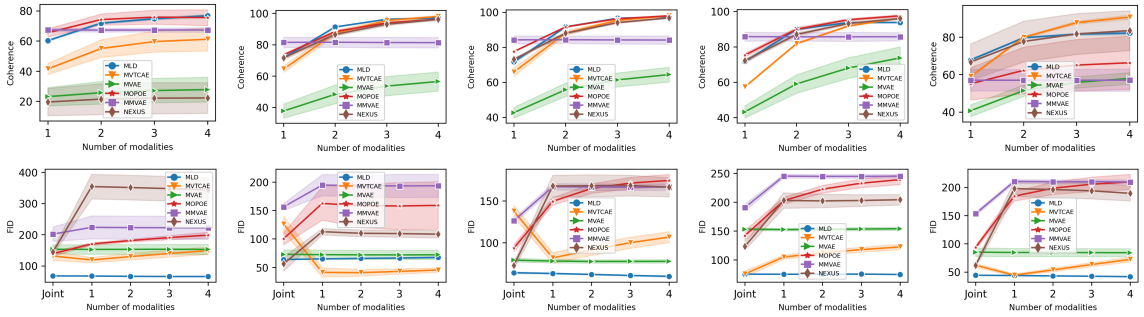
Models	Coherence (%↑)				
	Joint	**1**	**2**	**3**	**4**
MVAE	$4.0_{\pm 1.49}$	$37.51_{\pm 3.16}$	$48.06_{\pm 3.55}$	$53.19_{\pm 3.37}$	$56.09_{\pm 3.31}$
MMVAE	$25.8_{\pm 1.43}$	$\mathbf{75.15}_{\pm 2.54}$	$75.14_{\pm 2.47}$	$75.09_{\pm 2.6}$	$75.09_{\pm 2.58}$
MoPoE	$17.32_{\pm 2.47}$	$\underline{69.37}_{\pm 1.85}$	$\underline{81.29}_{\pm 2.34}$	$85.26_{\pm 2.36}$	$86.7_{\pm 2.39}$
NEXUS	$18.24_{\pm 0.89}$	$60.61_{\pm 2.51}$	$72.14_{\pm 2.79}$	$76.81_{\pm 2.75}$	$78.92_{\pm 2.64}$
MVTCAE	$0.21_{\pm 0.05}$	$57.66_{\pm 1.06}$	$78.44_{\pm 1.31}$	$\underline{85.97}_{\pm 1.43}$	$\underline{88.81}_{\pm 1.49}$
MMVAE+	$26.28_{\pm 2.19}$	$54.74_{\pm 0.5}$	$54.06_{\pm 0.33}$	$55.2_{\pm 1.32}$	$53.17_{\pm 0.75}$
MMVAE+ (K = 10)	$14.53_{\pm 4.94}$	$58.93_{\pm 6.3}$	$59.42_{\pm 8.8}$	$60.77_{\pm 8.03}$	$58.24_{\pm 7.42}$
MLD in-paint	$\underline{51.65}_{\pm 1.16}$	$52.85_{\pm 0.23}$	$77.65_{\pm 0.24}$	$85.66_{\pm 0.43}$	$87.29_{\pm 0.29}$
MLD uni	$48.79_{\pm 0.43}$	$65.12_{\pm 0.7}$	$79.52_{\pm 0.8}$	$82.03_{\pm 1.19}$	$81.86_{\pm 2.09}$
MLD	$\mathbf{56.23}_{\pm 0.52}$	$68.58_{\pm 0.72}$	$\mathbf{84.87}_{\pm 0.19}$	$\mathbf{88.56}_{\pm 0.12}$	$\mathbf{89.43}_{\pm 0.27}$

Table A.13: Generation quality (FID ↓) for **POLYMNIST** (lower is better) used for the plots in in Figure 3.3 and Figure A.5. Similar to Table A.12, we report the average *leave-one-out* FID as a function of the number of observed modalities. *Joint* refers to random generation of the five modalities simultaneously. Bold and underlined numbers indicate the best and second best scores respectively.

Models	Quality (↓)				
	Joint	**1**	**2**	**3**	**4**
MVAE	$108.74_{\pm 2.73}$	$108.06_{\pm 2.79}$	$108.05_{\pm 2.73}$	$108.14_{\pm 2.71}$	$108.18_{\pm 2.85}$
MMVAE	$165.74_{\pm 5.4}$	$208.16_{\pm 10.41}$	$207.5_{\pm 10.57}$	$207.35_{\pm 10.59}$	$207.38_{\pm 10.58}$
MoPoE	$113.77_{\pm 1.62}$	$173.87_{\pm 7.34}$	$185.06_{\pm 10.21}$	$191.72_{\pm 11.26}$	$196.17_{\pm 11.66}$
NEXUS	$91.66_{\pm 2.93}$	$207.14_{\pm 7.71}$	$205.54_{\pm 8.6}$	$204.46_{\pm 9.08}$	$202.43_{\pm 9.49}$
MVTCAE	$106.55_{\pm 3.83}$	$78.3_{\pm 2.35}$	$85.55_{\pm 2.51}$	$92.73_{\pm 2.65}$	$99.13_{\pm 2.72}$
MMVAE+	$168.88_{\pm 0.12}$	$165.67_{\pm 0.14}$	$166.5_{\pm 0.18}$	$165.53_{\pm 0.55}$	$165.3_{\pm 0.33}$
MMVAE+ (K = 10)	$156.55_{\pm 3.58}$	$154.42_{\pm 2.73}$	$153.1_{\pm 3.01}$	$153.06_{\pm 2.88}$	$154.9_{\pm 2.9}$
MLD in-paint	$64.78_{\pm 0.33}$	$65.41_{\pm 0.43}$	$65.42_{\pm 0.41}$	$65.52_{\pm 0.46}$	$\underline{65.55}_{\pm 0.46}$
MLD uni	$\mathbf{62.42}_{\pm 0.62}$	$\underline{63.16}_{\pm 0.81}$	$\underline{64.09}_{\pm 1.15}$	$\underline{65.17}_{\pm 1.46}$	$66.46_{\pm 2.18}$
MLD	$\underline{63.05}_{\pm 0.26}$	$\mathbf{62.89}_{\pm 0.2}$	$\mathbf{62.53}_{\pm 0.21}$	$\mathbf{62.22}_{\pm 0.39}$	$\mathbf{61.94}_{\pm 0.65}$

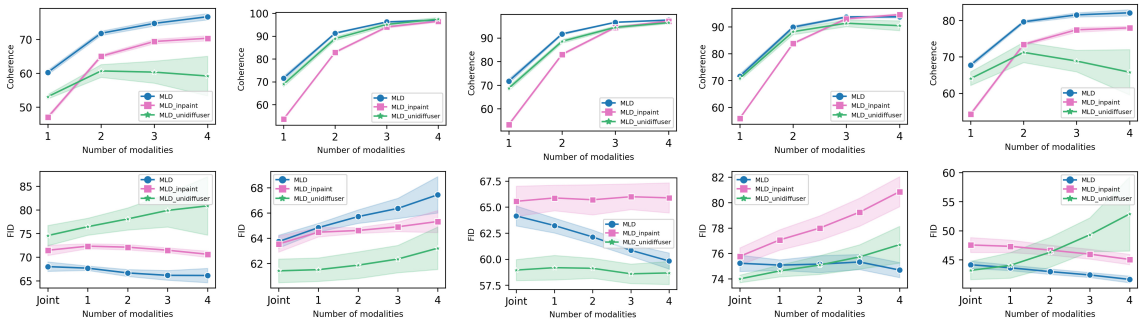
Additional Experiments with the Architecture from (Palumbo, Daunhawer, and Vogt, 2023)

In our experiments on POLYMNIST, we used the same architecture as in (Sutter, Daunhawer, and Vogt, 2021; Hwang et al., 2021) in order to ensure a fair settings for all the



(a) X^0 (b) X^1 (c) X^2 (d) X^3 (e) X^4

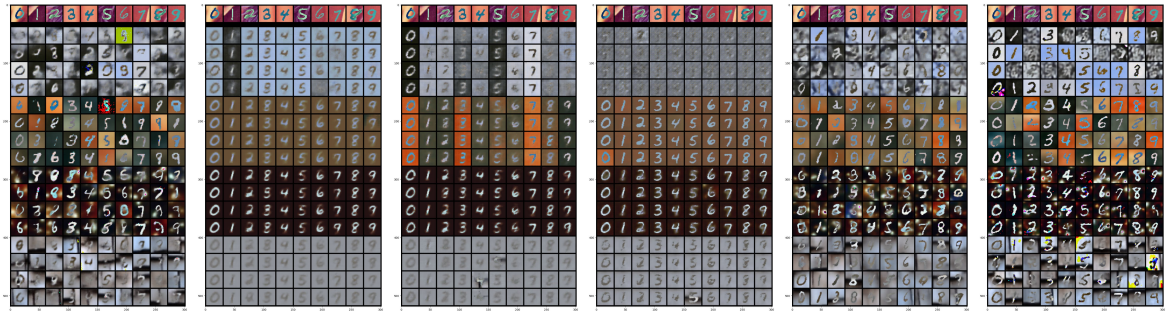
Figure A.14: Top: Generation coherence (%) for **POLYMNIST** (higher is better). **Bottom:** Generation quality (FID) (lower is better). We report the average *leave-one-out* performance as a function of the number of observed modalities for each modality X^i. *Joint* refers to random generation of the five modalities simultaneously.



(a) X^0 (b) X^1 (c) X^2 (d) X^3 (e) X^4

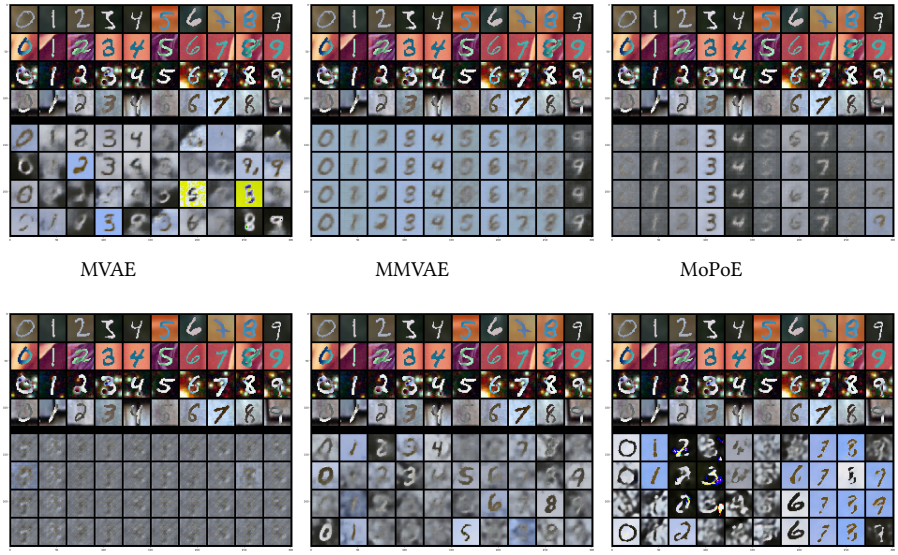
Figure A.15: Top: Generation coherence (%) for **POLYMNIST** (higher is better).**Bottom:** Generation quality (FID) (lower is better). We report the average *leave-one-out* performance as a function of the number of observed modalities for each modality X^i. *Joint* refers to random generation of the five modalities simultaneously.

baselines. In (Palumbo, Daunhawer, and Vogt, 2023), the experiments on POLYMNIST were conducted using a different autoencoder architecture based on Resnet instead of a sequence of autoencoder-based convolutional layers. In this section, we investigate the performance of MMVAE+ and our MLD using this architecture. For MMVAE+, we kept the same settings as in (Palumbo, Daunhawer, and Vogt, 2023), including the autoencoder architecture, latent size, and importance sampling K = 10 with doubly reparameterized gradient estimator (DReG). For MLD, we used the same autoencoder architecture with a latent size equal to 160. In Figure A.18, can be observed that while the new autoencoder architecture enhances the performance of MMVAE+, the performance our MLD is improved as well. Similar to the previous results, MLD simultaneously achieves the best generative coherence and the best quality.



MVAE MMVAE MoPoE NEXUS MVTCAE **MLD (ours)**

Figure A.16: Conditional generation qualitative results for **POLYMNIST**. Modality X^2 (first row) is used as the condition to generate the four remaining modalities (the rows below).



MVAE MMVAE MoPoE

NEXUS MVTCAE **MLD (ours)**

Figure A.17: Conditional generation qualitative results for **POLYMNIST**. The subset of modalities X^1, X^2, X^3, X^4 (first four rows) are used as the condition to generate modality X^0 (the rows below).

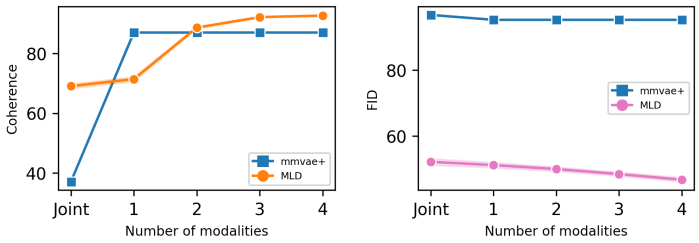


Figure A.18: Results for the POLYMNIST dataset. Left: Comparison of the generative coherence (↑) and quality in terms of FID (↓) as a function of the number of inputs.

A.5.4 CUB

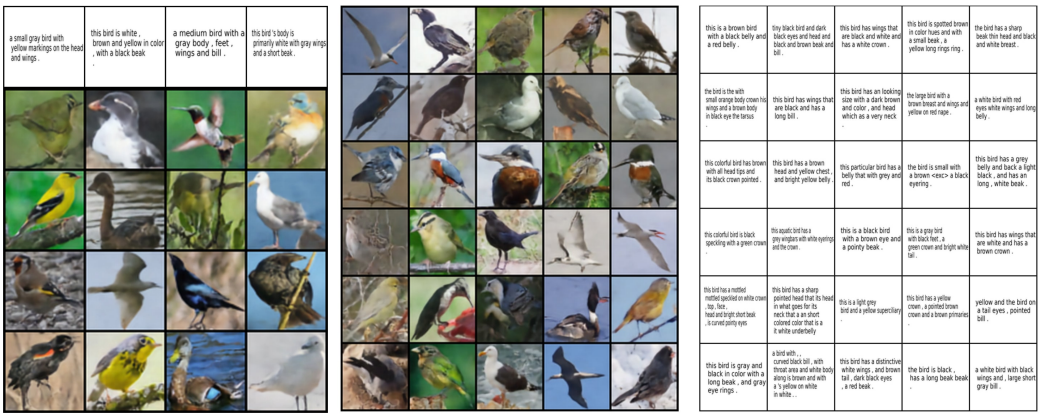
Table A.14: Generation coherence (Clip-s: higher is better) and quality (FID: ↓ lower is better) for the CUB dataset. **MLD*** denotes the version of our method using a more powerful image autoencoder. Bold numbers indicate the best scores.

Models	Coherence (↑)		Quality (↓)		
	JointImage	→ CaptionCaption	→ ImageJoint	→ ImageCaption	→ Image
MVAE	0.66	**0.70**	0.64	158.91	158.88
MMVAE	0.66	0.69	0.62	277.8	212.57
MoPoE	0.64	0.68	0.55	279.78	179.04
NEXUS	0.65	0.69	0.59	147.96	262.9
MVTCAE	0.65	**0.70**	0.65	155.75	168.17
MMVAE+	0.61	0.68	0.65	188.63	247.44
MMVAE+ (K=10)	0.63	0.68	0.62	172.21	178.88
MLD in-paint	**0.69**	0.69	0.68	69.16	68.33
MLD uni	**0.69**	0.69	**0.69**	64.09	**61.92**
MLD	**0.69**	0.69	**0.69**	**63.47**	62.62
MLD*	**0.70**	0.69	**0.69**	**22.19**	**22.50**

A.5.5 CelebAMask-HQ

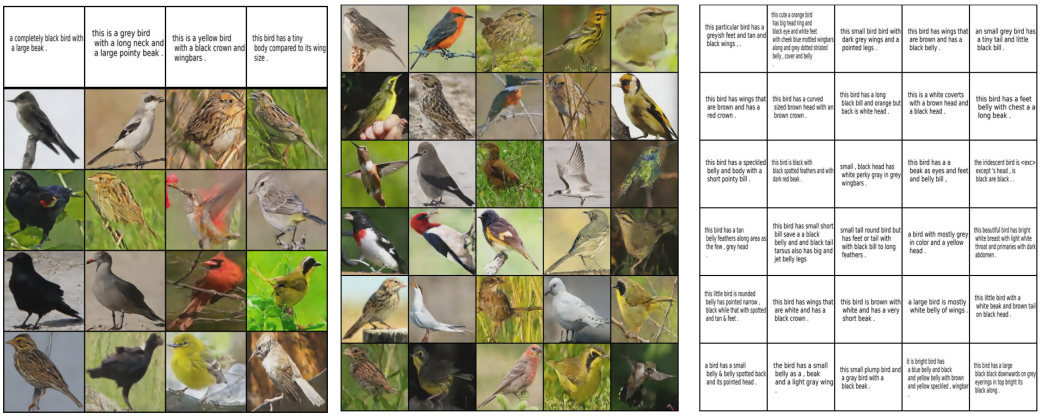
In this section, we present additional experiments on the CelebAMask-HQ dataset (Lee et al., 2019).

Figure A.19: Qualitative results for joint generation on the **CUB** dataset.(Better viewed zoomed)



(a) Conditional generation. **(b)** Joint generation.

Figure A.20: Qualitative results of **MLD*** on the **CUB** dataset with powerful image autoencoder. (Better viewed zoomed)



(a) Conditional Generation **(b)** Joint generation

Figure A.21: Qualitative results of **MLD*** on the **CUB** dataset with 128 × 128 resolution images and powerful image autoencoder. (Better viewed zoomed)

Figure A.22: (Attributes → Image). Conditional generation of **MLD** on CelebAMask-HQ.

Figure A.23: (Mask → Image) Conditional generation of **MLD** on CelebAMask-HQ.

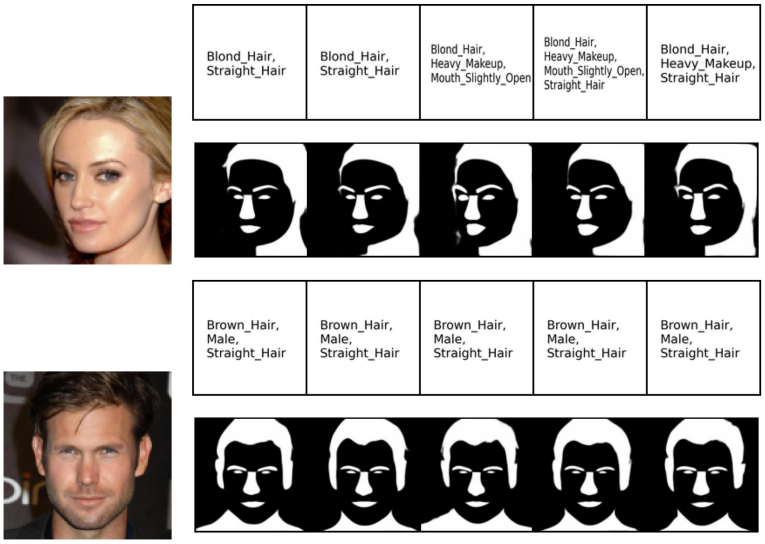


Figure A.24: (Image → Attribute, Mask). Conditional generation of **MLD** on CelebAMask-HQ.

Appendix B

Appendix for Chapter 4

B.1 Proofs

B.1.1 Details of Eq. (4.2)

For an SDE of the form:

$$\mathrm{d}\mathbf{X}_t = \mu(\mathbf{X}_t, t)\mathrm{d}t + \sigma(\mathbf{X}_t, t)\mathrm{d}W_t, \tag{B.1}$$

the probability density $p_t(\mathbf{x})$ satisfies the general Fokker–Planck equation:

$$\frac{\partial p}{\partial t} = -\frac{\partial}{\partial x}\big(\mu(x,t)p_t(\mathbf{x})\big) + \frac{1}{2}\frac{\partial^2}{\partial x^2}\big(\sigma^2(x,t)p_t(\mathbf{x})\big). \tag{B.2}$$

Comparing with the given SDE (4.1), we identify: $\mu(x,t) = f(t)x, \quad \sigma(x,t) = g(t)$. The drift term in (B.2) is:

$$-\frac{\partial}{\partial \mathbf{x}}\big(f(t)\mathbf{x}p_t(\mathbf{x})\big). \tag{B.3}$$

Applying the product rule:

$$\frac{\partial}{\partial \mathbf{x}}(xp_t) = p_t + \mathbf{x}\frac{\partial p_t}{\partial \mathbf{x}}, \tag{B.4}$$

we obtain:

$$-\frac{\partial}{\partial \mathbf{x}}(f(t)\mathbf{x}p_t) = -f(t)\big(p_t + \mathbf{x}\frac{\partial p_t}{\partial \mathbf{x}}\big). \tag{B.5}$$

Then we compute Second-Order Derivative:

$$\frac{1}{2}\frac{\partial^2}{\partial \mathbf{x}^2}\big(g(t)^2 p_t(\mathbf{x})\big). \tag{B.6}$$

Since $g(t)$ depends only on t, it factors out:

$$\frac{1}{2}g(t)^2\frac{\partial^2 p_t}{\partial \mathbf{x}^2}. \tag{B.7}$$

Substituting these into (B.2), we obtain:

$$\frac{\partial p_t}{\partial t} = -f(t)\big(p_t + \mathbf{x}\frac{\partial p_t}{\partial \mathbf{x}}\big) + \frac{1}{2}g(t)^2\frac{\partial^2 p_t}{\partial \mathbf{x}^2}. \tag{B.8}$$

Using vector notation:

$$\frac{\partial p_t}{\partial t} = -\nabla(f(t)\mathbf{x}p_t) + \frac{1}{2}g(t)^2\Delta p_t. \tag{B.9}$$

B.1.2 Proof of Proposition 1

We aim to demonstrate that :

$$-\int_0^T \frac{\mathrm{d}\mathbf{r}_t}{\mathrm{d}t}\mathrm{d}t = \frac{1}{2}\int p_t\, g(t)^2\, \|\nabla \log p_t - \nabla \log q_t\|^2 \mathrm{d}\mathbf{x}\mathrm{d}t \tag{B.10}$$

Using the Leibniz rule (Under smoothness assumptions), we can write:

$$\frac{\mathrm{d}\mathbf{r}_t}{\mathrm{d}t} = \frac{\mathrm{d}}{\mathrm{d}t}\int p_t\, \log\frac{p_t}{q_t}\, d\mathbf{x} = \int \frac{\partial}{\partial t}\Big(p_t\, \log\frac{p_t}{q_t}\Big) d\mathbf{x}. \tag{B.11}$$

Using the partial derivative product rule:

$$\frac{\partial}{\partial t}\Big(p_t\, \log\frac{p_t}{q_t}\Big) = \frac{\partial p_t}{\partial t}\, \log\frac{p_t(\mathbf{x})}{q_t} + p_t\, \frac{\partial}{\partial t}\log\frac{p_t}{q_t}. \tag{B.12}$$

Substituting back into the integral and integrating with respect to t we get :

$$C = \int \frac{\mathrm{d}\mathbf{r}_t}{\mathrm{d}t}\,\mathrm{d}t = \int \frac{\partial p_t}{\partial t}\, \log\frac{p_t}{q_t} + \underbrace{p_t\, \frac{\partial}{\partial t}\Big(\log\frac{p_t}{q_t}\Big)}_{(1)}\, \mathrm{d}\mathbf{x}\,\mathrm{d}t. \tag{B.13}$$

(1) is simplified as follows :

$$(1) = \int p_t \frac{\partial}{\partial t} \log \frac{p_t}{q_t} = \int p_t \left(\frac{\partial \log p_t}{\partial t} - \frac{\partial \log q_t}{\partial t} \right) \mathrm{d}x\mathrm{d}t \tag{B.14}$$

$$= \int p_t \frac{\partial p_t}{\partial t} \frac{1}{p_t} - p_t \frac{\partial q_t}{\partial t} \frac{1}{q_t} \mathrm{d}x\mathrm{d}t \tag{B.15}$$

$$= \int \frac{\partial p_t}{\partial t} \mathrm{d}x\mathrm{d}t \; - \int \frac{p_t}{q_t} \frac{\partial q_t}{\partial t} \, \mathrm{d}x\mathrm{d}t \tag{B.16}$$

The last simplification in Eq. (B.16) is possible by noticing that :

$$\int \frac{\partial p_t}{\partial t} \mathrm{d}x\mathrm{d}t = \int \frac{\partial \left(\int p_t \mathrm{d}x \right)}{\partial t} \mathrm{d}t = \int \frac{\partial \left(1 \right)}{\partial t} \mathrm{d}t = 0 \tag{B.17}$$

We have then :

$$C = \int \frac{\partial p_t}{\partial t} \log \frac{p_t}{q_t} - \int \frac{p_t}{q_t} \frac{\partial q_t}{\partial t} \, \mathrm{d}x\mathrm{d}t \tag{B.18}$$

p_t and q_t evolve according to the Fokker–Planck equations which allows to write:

$$\frac{\partial p_t}{\partial t} = -\nabla \cdot \Big(f(t)\, \mathbf{x}\, p_t \Big) + \frac{1}{2}\, g(t)^2\, \Delta p_t, \tag{B.19}$$

$$\frac{\partial q_t}{\partial t} = -\nabla \cdot \Big(f(t)\, \mathbf{x}\, q_t \Big) + \frac{1}{2}\, g(t)^2\, \Delta q_t. \tag{B.20}$$

Substituting Eq. (B.20) into Eq. (B.18) and rearranging the different the terms we can write $C = C_1 + C_2$ with :

$$C_1 = -\int \log \frac{p_t}{q_t} \nabla \Big(f(t)\, \mathbf{x}\, p_t \Big) \; + \frac{p_t}{q_t} \nabla \cdot \Big(f(t)\, \mathbf{x}\, q_t \Big) \, \mathrm{d}\mathbf{x}\, \mathrm{d}t. \tag{B.21}$$

$$C_2 = \frac{1}{2} \int g(t)^2\, \Delta p_t \, \log \frac{p_t}{q_t} - \frac{p_t}{q_t}\, g(t)^2\, \Delta q_t \, \mathrm{d}\mathbf{x}\, \mathrm{d}t. \tag{B.22}$$

The first term C_1 We should first recall that densities p_t, q_t are equal to zero at infinite values of $|\mathbf{x}| \to \infty$. This means that p_t, q_t and their products with any polynomial factors (like $\mathbf{x}$) decay sufficiently fast so that the boundary integral is zero which permits us to apply integration by parts.

By applying integration by parts, we can show that the first term equal to zero :

$$C_1 = \int -\log \frac{p_t}{q_t} \nabla\Big(f(t)\,\mathbf{x}\,p_t\Big) \; + \frac{p_t}{q_t}\,\nabla\Big(f(t)\,\mathbf{x}\,q_t\Big)\,\mathrm{d}\mathbf{x}\,\mathrm{d}t \tag{B.23}$$

$$= 0 \tag{B.24}$$

Indeed by manipulating the first term in Eq. (B.24): applying integration by part two times in Eq. (B.25) and Eq. (B.28) then substituting the logarithm partial derivative in Eq. (B.26):

$$\int \log \frac{p_t}{q_t} \nabla\Big(f(t)\,\mathbf{x}\,p_t\Big)\mathrm{d}\mathbf{x}\mathrm{d}t = -\int \nabla\Big(\log \frac{p_t}{q_t}\Big) f(t)\,\mathbf{x}\,p_t \mathrm{d}\mathbf{x}\mathrm{d}t \tag{B.25}$$

$$= -\int \nabla\Big(\frac{p_t}{q_t}\Big) \frac{q_t}{p_t} f(t)\,\mathbf{x}\,p_t \mathrm{d}\mathbf{x}\mathrm{d}t \tag{B.26}$$

$$= -\int \nabla\Big(\frac{p_t}{q_t}\Big)\,\mathbf{x}\,q_t \mathrm{d}\mathbf{x}\mathrm{d}t \tag{B.27}$$

$$= \int \nabla\Big(f(t)\mathbf{x}\,q_t\Big)\,\frac{p_t}{q_t}\mathrm{d}\mathbf{x}\mathrm{d}t \tag{B.28}$$

The term C_2 can be simplified in a similar manner :

$$C_2 = \frac{1}{2}\int g(t)^2\,\Delta p_t\,\log\frac{p_t}{q_t} - \frac{p_t}{q_t}\,g(t)^2\,\Delta q_t\,\mathrm{d}\mathbf{x}\,\mathrm{d}t.$$

$$= \frac{1}{2}\int g(t)^2 \left[-\nabla p_t\,\nabla \log\frac{p_t}{q_t} + \nabla\frac{p_t}{q_t}\,\nabla q_t\right]\,\mathrm{d}\mathbf{x}\,\mathrm{d}t. \tag{B.29}$$

$$= \frac{1}{2}\int g(t)^2 \left[-\nabla p_t\,\nabla \log\frac{p_t}{q_t} + \frac{p_t}{q_t}\nabla q_t \nabla \log\frac{p_t}{q_t}\right]\,\mathrm{d}\mathbf{x}\,\mathrm{d}t.$$

$$= \frac{1}{2}\int p_t\,g(t)^2 \nabla \log\frac{p_t}{q_t}\left[-\frac{\nabla p_t}{p_t} + \frac{\nabla q_t}{q_t}\right]\,\mathrm{d}\mathbf{x}\,\mathrm{d}t.$$

$$= \frac{1}{2}\int p_t\,g(t)^2 \nabla \log\frac{p_t}{q_t}\left[-\nabla \log\frac{p_t}{q_t}\right]\,\mathrm{d}\mathbf{x}\,\mathrm{d}t. \tag{B.30}$$

In Eq. (B.29), we start by applying integration by parts and then in the remaining operations, we use $\nabla \log \frac{p_t}{q_t} = \frac{\nabla p_t}{p_t} - \frac{\nabla q_t}{q_t}$. We obtain :

$$C2 = -\frac{1}{2}\int p_t(\mathbf{x})\,g(t)^2\,\|\nabla \log p_t(\mathbf{x}) - \nabla \log q_t(\mathbf{x})\|^2 \mathrm{d}\mathbf{x}\mathrm{d}t \quad \square \tag{B.31}$$

B.1.3 Proof for Eq. (4.14)

To prove such claim, it is sufficient to start from the r.h.s. of Eq. (4.13), substitute to the parametric scores their definition with the errors $s^p(\mathbf{x},t) = \mathbf{e}_t^p(\mathbf{x}) + s_*^p(\mathbf{x},t)$, and $s^q(\mathbf{x},t) = s_*^q(\mathbf{x},t) + \mathbf{e}_t^q(\mathbf{x})$, and expand the square. We consider that T is large enough such that $\widetilde{\mathbb{KL}}\left(p_T \parallel q_T\right) = 0$ is a vanishing term.

$$
\begin{aligned}
\widetilde{\mathbb{KL}}\left(p \parallel q\right) &= \int_0^T \mathbb{E}_{\mathbf{x}\sim p_t(\mathbf{x})}\left[\frac{g^2(t)}{2}\|s^p(\mathbf{x},t) - s^q(\mathbf{x},t)\|^2\right]\mathrm{d}t \\
&= \int_0^T \mathbb{E}_{\mathbf{x}\sim p_t(\mathbf{x})}\left[\frac{g^2(t)}{2}\|\mathbf{e}_t^p(\mathbf{x}) + s_*^p(\mathbf{x},t) - s_*^q(\mathbf{x},t) - \mathbf{e}_t^q(\mathbf{x})\|^2\right]\mathrm{d}t \\
&= \int_0^T \mathbb{E}_{\mathbf{x}\sim p_t(\mathbf{x})}\left[\frac{g^2(t)}{2}\|s_*^p(\mathbf{x},t) - s_*^q(\mathbf{x},t)\|^2 - \frac{g^2(t)}{2}\|\mathbf{e}_t^p(\mathbf{x}) - \mathbf{e}_t^q(\mathbf{x})\|^2\right. \\
&\quad \left. - 2\langle\, s^p(\mathbf{x},t) - s^q(\mathbf{x},t),\, \mathbf{e}_t^p(\mathbf{x}) - \mathbf{e}_t^q(\mathbf{x})\,\rangle\right]\mathrm{d}t \\
&= \mathbb{KL}\left(p \parallel q\right) - \int_0^T \mathbb{E}_{\mathbf{x}\sim p_t(\mathbf{x})}\left[\frac{g^2(t)}{2}\|\mathbf{e}_t^p(\mathbf{x}) - \mathbf{e}_t^q(\mathbf{x})\|^2\right. \\
&\quad \left. - 2\langle\, s^p(\mathbf{x},t) - s^q(\mathbf{x},t),\, \mathbf{e}_t^p(\mathbf{x}) - \mathbf{e}_t^q(\mathbf{x})\,\rangle\right]\mathrm{d}t
\end{aligned}
\tag{B.32}
$$

B.1.4 Proof for Eq. (4.28)

Starting from the three-term score-based KL decomposition (cf. (4.26)), and letting $\chi_t^{-1} \to 0$ as $\sigma \to \infty$, we drop all reference-score terms to obtain

$$
\begin{aligned}
\mathcal{I}(\mathbf{X},\mathbf{Y}) \simeq \int_0^T \mathbb{E}\Big[& -\tfrac{g^2(t)}{2}\left\|s^{\mathbf{X},\mathbf{Y}}(\mathbf{x}_t,\mathbf{y}_t,t)\right\|^2 && \text{(B.33)} \\
& + \tfrac{g^2(t)}{2}\left\|s^{\mathbf{X}|\mathbf{Y}}(\mathbf{x}_t,\mathbf{y}_0,t)\right\|^2 && \text{(B.34)} \\
& + \tfrac{g^2(t)}{2}\left\|s^{\mathbf{Y}|\mathbf{X}}(\mathbf{y}_t,\mathbf{x}_0,t)\right\|^2\Big]\,\mathrm{d}t. && \text{(B.35)}
\end{aligned}
$$

Define for brevity:

$$
\begin{aligned}
a &= s^{\mathbf{X},\mathbf{Y}}(\mathbf{x}_t,\mathbf{y}_t,t), && \text{(B.36)} \\
c &= s^{\mathbf{X}|\mathbf{Y}}(\mathbf{x}_t,\mathbf{y}_0,t), && \text{(B.37)} \\
d &= s^{\mathbf{Y}|\mathbf{X}}(\mathbf{y}_t,\mathbf{x}_0,t). && \text{(B.38)}
\end{aligned}
$$

Then algebraically

$$-\|a\|^2 + \|c\|^2 + \|d\|^2 = \big\|a - [c;d]\big\|^2 + 2\,\langle a, [c;d]\rangle - 2\,\langle c, d\rangle. \tag{B.39}$$

However, each score has zero mean under its own sampling law. Indeed, for any density p with it's score function $s = \nabla_x \log p(x)$.

$$\begin{aligned}
\mathbb{E}_{x\sim p}\big[s(x)\big] &= \int p(x)\,\nabla_x \log p(x)\,\mathrm{d}x && \text{(B.40)}\\
&= \int \nabla_x p(x)\,\mathrm{d}x \qquad\qquad \text{since } \nabla_x \log p(x) = \tfrac{\nabla_x p(x)}{p(x)}) && \text{(B.41)}\\
&= 0 && \text{(B.42)}
\end{aligned}$$

The last operation can be verified by applying integration by parts by choosing $u(\mathbf{x}) = 1 \text{and} v(\mathbf{x}) = p(\mathbf{x})$.

Applying this to the score functions $s^{\mathbf{X},\mathbf{Y}}$, $s^{\mathbf{X}|\mathbf{Y}}$, and $s^{\mathbf{Y}|\mathbf{X}}$, each of which has zero mean under its own sampling distribution, and using the independence of the conditional processes given the initial condition $(\mathbf{x}_0, \mathbf{y}_0)$, all cross-terms in the expanded square vanish in expectation.

$$\begin{aligned}
\mathbb{E}\langle a, [c;d]\rangle &= \mathbb{E}[a]^\top \mathbb{E}[c;d] = 0, && \text{(B.43)}\\
\mathbb{E}\langle c, d\rangle &= \mathbb{E}[c]^\top \mathbb{E}[d] = 0. && \text{(B.44)}
\end{aligned}$$

It follows that

$$-\mathbb{E}\|a\|^2 + \mathbb{E}\|c\|^2 + \mathbb{E}\mathbb{E}\|d\|^2 = \mathbb{E}\big\|a - [c;d]\big\|^2. \tag{B.45}$$

Substituting back into the time-integral yields the single-difference we recover (4.28). □

B.2 Implementation details

In this Section, we provide additional technical details of MINDE. We discuss the different variants of our method their implementation details, including detailed information about the MI estimators alternatives considered in § 4.6.

Algorithm 6: MINDE–C (Single Training Step)

Data: $[\mathbf{x}_0, \mathbf{y}_0] \sim p^{\mathbf{X},\mathbf{Y}}$
parameter: $net\theta()$, with θ current parameters
$t \sim \mathcal{U}[0,T]$ $\mathbf{x}_t \sim p_{0t}(.|\mathbf{x}_0)$ // diffuse the variable X to timestep t
$c \sim$ Bernoulli(d) // Sample binary variable c with probability d
if $c = 0$ **then**
  $\tilde{s}^{\mathbf{X}} \leftarrow net_\theta([\mathbf{x}_t, 0], t, c = 0)$ // Estimated unconditional score
else
  $\tilde{s}^{\mathbf{X}|\mathbf{Y}} \leftarrow net_\theta([\mathbf{x}_t, \mathbf{y}_0], t, c = 1)$ // Estimated conditional score
$\mathcal{L} \leftarrow$ *Eq.* (4.12) // Compute the denoising score matching loss
return Update θ according to gradient of $\mathcal{L}$

Algorithm 7: MINDE–C

Data: $[\mathbf{x}_0, \mathbf{y}_0] \sim p^{\mathbf{X},\mathbf{Y}}$
parameter: σ, *option*
$t \sim \mathcal{U}[0,T]$ // Importance sampling can be used to reduce variance
$\mathbf{x}_t \sim p_{0t}(.|\mathbf{x}_0)$ // diffuse the variable X to timestep t
$\tilde{s}^{\mathbf{X}} \leftarrow net_\theta([\mathbf{x}_t, 0], t, c = 0)$ // The marginal score
$\tilde{s}^{\mathbf{X}|\mathbf{Y}} \leftarrow net_\theta([\mathbf{x}_t, \mathbf{y}_0], t, c = 1))$ // The conditional score
if *option* $= 1$ **then**
  // Eq. (4.27)
  $\hat{\mathcal{I}} \leftarrow T\frac{g^2(t)}{2}\left\|\tilde{s}^{\mathbf{X}} - \tilde{s}^{\mathbf{X}|\mathbf{Y}}\right\|^2$
else
  // Eq. (4.25)
  $\hat{\mathcal{I}} \leftarrow T\frac{g^2(t)}{2}\left[\left\|\tilde{s}^{\mathbf{X}} + \mathbf{x}_t\,\chi_t^{-1}\right\|^2 - \left\|\tilde{s}^{\mathbf{X}|\mathbf{Y}} + \mathbf{x}_t\,\chi_t^{-1}\right\|^2\right]$ // See § 4.4 for χ_t formulation which depends on σ.
return $\hat{\mathcal{I}}$

B.2.1 MINDE-c

In all experiments, we consider the first variable as the main variable and the second variable as the conditioning signal. A single neural network is used to model the conditional and unconditional score. It accepts as inputs the two variables, the diffusion time t, and an additionally binary input c which enable the conditional mode. To enable the conditional mode, we set $c = 1$ and feed the network with both the main variable and the conditioning signal, obtaining $\tilde{s}^{\mathbf{X}|\mathbf{Y}}$. To obtain the marginal score $\tilde{s}^{\mathbf{X}}$, we set $c = 0$ and the conditioning signal is set to zero value.

A randomized procedure is used for training. For each training step, with probability d, the main variable is diffused and the score network is fed with the diffused variable, the conditioning variable, the diffusion time signal and the conditioning signal is set to $c = 1$. On the contrary, with probability $1 - d$, to enable the network to learn the unconditional

score, the network is fed only with the diffused modality, the diffusion time and $c = 0$. In contrast to the first case, the conditioning is not provided to the score network and replaced with a zero value vector. Pseudo-code is presented in Algorithm 6.

Actual estimation of the MI is then possible either by leveraging Eq. (4.27) or Eq. (4.25), referred to in the main text as difference *outside* or *inside* the score respectively (MINDE-C(σ), MINDE-C). A pseudo-code description is provided in Algorithm 7.

B.2.2 MINDE-J

The joint variant of our method, MINDE-J is based on the parametrized joint processes in Eq. (4.24). Also in this case, instead of training a separate score network for each possible combination of conditional modalities, we use a single architecture that accepts both variables, the diffusion time t and the coefficients α, β. This approach allows modeling the joint score network $\tilde{s}_t^{\mathbf{X},\mathbf{Y}}$ by setting $\alpha = \beta = 1$. Similarly, to obtain the conditional scores it is sufficient to set $\alpha = 1, \beta = 0$ or $\alpha = 0, \beta = 1$, corresponding to $\tilde{s}^{\mathbf{X}|\mathbf{Y}}$ and $\tilde{s}^{\mathbf{Y}|\mathbf{X}}$ respectively.

Training is carried out again through a randomized procedure. At each training step, with probability d, both variables are diffused. In this case, the score network is fed with diffusion time t, along with $\mathbf{X}_t$, $\mathbf{Y}_t$ and the two parameters $\alpha = \beta = 1$. With probability $1 - d$, instead, we randomly select one variable to be diffused, while we keeping constant the other. For instance, if A is the one which is diffused, we set $\alpha = 1$ and $\beta = 0$. Further details are presented in Algorithm 8.

Once the score network is trained, MI estimation can be obtained following the procedure explained in Algorithm 9. Two options are possible, either by computing the difference between the parametric scores outside the same norm (Eq. (4.26) MINDE-J(σ) or inside (Eq. (4.28) MINDE-J). Similarly to the conditional case, an *option* parameter can be used to switch among the two.

B.2.3 Technical settings for MINDE-C and MINDE-J

We follow the implementation of (Bounoua, Franzese, and Michiardi, 2024) which uses stacked multi-layer perception (MLP) with skip connections. We adopt a simplified version of the same score network architecture: this involves three Residual MLP blocks. We use the *Adam optimizer* (Kingma and Ba, 2014) for training and Exponential moving average (EMA) with a momentum parameter $m = 0.999$. We use importance sampling at train and test-time. We returned the mean estimate on the test data set over 10 runs.

Algorithm 8: MINDE–J (Single Training Step)

Data: $[\mathbf{x}_0, \mathbf{y}_0] \sim p^{\mathbf{X},\mathbf{Y}}$
parameter: $net_\theta()$, with θ current parameters
$t \sim \mathcal{U}[0, T]$
$[\mathbf{x}_t, \mathbf{y}_t] \sim p_{0_t}(.|[\mathbf{x}_0, \mathbf{y}_0])$ `// Diffuse modalities to timestep` t
$c \sim$ Bernoulli(d) `// Sample binary variable` c `with probability` d
if $c = 0$ **then**
  $\tilde{s}^{\mathbf{X},\mathbf{Y}} \leftarrow net_\theta([\mathbf{x}_t, \mathbf{y}_t], t, [1, 1])$ `// Estimated Joint score`
else
  if *Bernoulli(0.5)* **then**
    $\tilde{s}^{\mathbf{X}|\mathbf{Y}} \leftarrow net_\theta([\mathbf{x}_t, \mathbf{y}_0], t, [1, 0])$ `// Estimated conditional score`
  else
    $\tilde{s}^{\mathbf{Y}|\mathbf{X}} \leftarrow net_\theta([\mathbf{x}_0, \mathbf{y}_t], t, [0, 1])$ `// Estimated conditional score`
$\mathcal{L} \leftarrow$ *Eq.* (4.12) `// Compute the denoising score matching loss`
return Update θ according to gradient of $\mathcal{L}$

The hyper-parameters are presented in Table B.1 and Table B.2 for MINDE-J and MINDE-C respectively. Concerning the consistency tests (§ 4.6.2), we independently train an autoencoder for each version of the MNIST dataset with r rows available.

B.2.4 Neural estimators implementation

We use the package *benchmark-mi*[1] implementation to study the neural estimators. We use MLP architecture with 3 layers of the same width as in MINDE. We use the same training procedure as in (Czyż et al., 2023), including early stopping strategy. We return the highest estimate on the test data.

B.3 Ablations study

B.3.1 σ Ablation study

We hereafter report in Table B.3 the results of all the variants of MINDE, including different values of σ parameter. For completeness in our experimental campaign, we report also the results of non neural competitors, similarly to the work in (Czyż et al., 2023). In summary, the MINDE-C/J versions ("*difference inside*") of our estimator prove to be more robust than the MINDE-C/J(σ) ("*difference outside*") counterpart, especially for the joint variants. Nevertheless, it is interesting to notice that the "*difference outside*" variants are stable and

[1] https://github.com/cbg-ethz/bmi

Algorithm 9: MINDE–J

Data: $[\mathbf{x}_0, \mathbf{y}_0] \sim p^{\mathbf{X},\mathbf{Y}}$
parameter: σ, *option*
$t \sim \mathcal{U}[0, T]$ // Importance sampling can be used to reduce variance
$[\mathbf{x}_t, \mathbf{y}_t] \sim p_{0_t}(.|[\mathbf{x}_0, \mathbf{y}_0])$ // Diffuse modalities to timestep t
$\tilde{s}^{\mathbf{X},\mathbf{Y}} \leftarrow net_\theta([\mathbf{x}_t, \mathbf{y}_t], t, [1, 1])$ // Estimated Joint score
$\tilde{s}^{\mathbf{X}|\mathbf{Y}} \leftarrow net_\theta([\mathbf{x}_t, \mathbf{y}_0], t, [1, 0])$ // Estimated conditional score
$\tilde{s}^{\mathbf{Y}|\mathbf{X}} \leftarrow net_\theta([\mathbf{x}_0, \mathbf{y}_t], t, [0, 1])$ // Estimated conditional score
if *option* $= 1$ **then**
  // Eq. (4.28)
  $\hat{\mathcal{I}} \leftarrow T\frac{g^2(t)}{2}\left\|\tilde{s}^{\mathbf{X},\mathbf{Y}} - [\tilde{s}^{\mathbf{X}|\mathbf{Y}}, \tilde{s}^{\mathbf{Y}|\mathbf{X}}]\right\|^2$
else
  // Eq. (4.26)
  $\hat{\mathcal{I}} \leftarrow T\frac{g^2(t)}{2}\left[\left\|\tilde{s}^{\mathbf{X},\mathbf{Y}} + [\mathbf{x}_t, \mathbf{y}_t]\,\chi_t^{-1}\right\|^2 - \left\|\tilde{s}^{\mathbf{X}|\mathbf{Y}} + \mathbf{x}_t\,\chi_t^{-1}\right\|^2 - \left\|\tilde{s}^{\mathbf{Y}|\mathbf{X}} + \mathbf{y}_t\,\chi_t^{-1}\right\|^2\right]$
  // See § 4.4 for χ_t formulation which depends on σ.
return $\hat{\mathcal{I}}$

Table B.1: MINDE-J score network training hyper-parameters. Dim of the task correspond the sum of the two variables dimensions, whereas d corresponds to the randomization probability.

	d	Width	Time embed	Batch size	Lr	Iterations	Number of params
Benchmark ($Dim \leq 10$)	0.5	64	64	128	1e-3	234k	55490
Benchmark ($Dim = 50$)	0.5	128	128	256	2e-3	195k	222100
Benchmark ($Dim = 100$)	0.5	256	256	256	2e-3	195k	911204
Consistency tests	0.5	256	256	64	1e-3	390k	1602080

competitive against a very wide range of values of σ (ranging from 0.5 to 10), with their best value typically achieved for $\sigma = 1.0$.

Table B.2: MINDE-C score network training hyper-parameters. Dim of the task correspond the sum of the two variables dimensions, and d corresponds to the randomization probability.

	d	Width	Time embed	Batch size	Lr	Iterations	Number of params
Benchmark ($Dim \leq 10$)	0.5	64	64	128	1e-3	390k	55425
Benchmark ($Dim = 50$)	0.5	128	128	256	2e-3	290k	220810
Benchmark ($Dim = 100$)	0.5	256	256	256	2e-3	290k	898354
Consistency tests	0.5	256	256	64	1e-3	390k	1597968

	Asinh @ *St* 1 × 1 (dof=1)	Asinh @ *St* 2 × 2 (dof=1)	Asinh @ *St* 3 × 3 (dof=2)	Asinh @ *St* 5 × 5 (dof=2)	Bimodal 1 × 1	Bivariate *Nm* 1 × 1	*Hc* @ Bivariate *Nm* 1 × 1	*Hc* @ *Mn* 25 × 25 (2-pair)	*Hc* @ *Mn* 3 × 3 (2-pair)	*Hc* @ *Mn* 5 × 5 (2-pair)	*Mn* 2 × 2 (2-pair)	*Mn* 2 × 2 (dense)	*Mn* 25 × 25 (2-pair)	*Mn* 25 × 25 (dense)	*Mn* 3 × 3 (2-pair)	*Mn* 3 × 3 (dense)	*Mn* 5 × 5 (2-pair)	*Mn* 5 × 5 (dense)	*Mn* 50 × 50 (dense)	*Nm* CDF @ Bivariate *Nm* 1 × 1	*Nm* CDF @ *Mn* 25 × 25 (2-pair)	*Nm* CDF @ *Mn* 3 × 3 (2-pair)	*Nm* CDF @ *Mn* 5 × 5 (2-pair)	*Sp* @ *Mn* 25 × 25 (2-pair)	*Sp* @ *Mn* 3 × 3 (2-pair)	*Sp* @ *Mn* 5 × 5 (2-pair)	*Sp* @ *Nm* CDF @ *Mn* 25 × 25 (2-pair)	*Sp* @ *Nm* CDF @ *Mn* 3 × 3 (2-pair)	*Sp* @ *Nm* CDF @ *Mn* 5 × 5 (2-pair)	*St* 1 × 1 (dof=1)	*St* 2 × 2 (dof=1)	*St* 2 × 2 (dof=2)	*St* 3 × 3 (dof=2)	*St* 3 × 3 (dof=3)	*St* 5 × 5 (dof=2)	*St* 5 × 5 (dof=3)	Swiss roll 2 × 1	Uniform 1 × 1 (additive noise=0.1)	Uniform 1 × 1 (additive noise=0.75)	Wiggly @ Bivariate *Nm* 1 × 1
GT	0.2	0.4	0.3	0.4	0.4	0.4	0.4	1.0	1.0	1.0	1.0	0.3	1.0	1.3	1.0	0.4	1.0	0.6	1.6	0.4	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.2	0.4	0.2	0.3	0.2	0.4	0.3	0.4	1.7	0.3	0.4
MINDE–j	0.2	0.4	0.3	0.4	0.4	0.4	0.4	1.2	1.0	1.0	1.0	0.3	1.0	1.3	1.0	0.4	1.0	0.6	1.7	0.4	1.1	1.0	1.0	1.0	0.9	0.9	1.1	1.0	1.0	0.1	0.2	0.2	0.3	0.2	0.5	0.3	0.4	1.7	0.3	0.4
MINDE–c	0.2	0.4	0.3	0.4	0.4	0.4	0.4	1.0	1.0	1.0	1.0	0.3	1.0	1.3	1.0	0.4	1.0	0.6	1.6	0.4	1.0	1.0	1.0	0.9	0.9	0.9	1.0	1.0	1.0	0.1	0.3	0.2	0.3	0.2	0.4	0.3	0.4	1.7	0.3	0.4
MINDE–j ($\sigma = 0.5$)	0.2	0.4	0.3	0.4	0.4	0.4	0.4	1.3	1.0	1.1	1.0	0.3	0.9	1.1	1.0	0.4	1.0	0.6	1.7	0.4	1.0	1.0	1.0	0.8	0.9	0.9	1.0	1.0	1.0	0.2	0.3	0.2	0.3	0.2	0.5	0.3	0.5	1.7	0.3	0.4
MINDE–j ($\sigma = 1$)	0.2	0.4	0.3	0.4	0.4	0.4	0.4	1.1	1.0	1.0	1.0	0.3	0.9	1.2	1.0	0.4	1.0	0.6	1.7	0.4	1.0	1.0	1.0	0.9	0.9	0.9	1.0	0.9	1.0	0.2	0.4	0.2	0.3	0.2	0.5	0.3	0.5	1.6	0.3	0.4
MINDE–j ($\sigma = 1.5$)	0.2	0.4	0.3	0.4	0.4	0.4	0.4	1.0	1.0	1.0	1.0	0.3	0.9	1.2	1.0	0.4	1.0	0.6	1.7	0.4	0.9	1.0	1.0	0.9	0.9	0.9	0.9	0.9	1.0	0.2	0.4	0.2	0.2	0.2	0.5	0.3	0.5	1.6	0.3	0.4
MINDE–j ($\sigma = 2$)	0.2	0.4	0.3	0.4	0.4	0.4	0.4	1.0	1.0	0.9	1.0	0.3	0.9	1.3	1.0	0.4	1.0	0.6	1.8	0.4	0.9	1.0	1.0	0.9	0.9	0.9	0.9	0.9	0.9	0.2	0.4	0.2	0.3	0.2	0.5	0.3	0.5	1.7	0.3	0.4
MINDE–j ($\sigma = 3$)	0.2	0.4	0.2	0.4	0.4	0.4	0.4	0.9	1.0	0.9	1.0	0.3	0.9	1.3	1.0	0.4	0.9	0.6	1.8	0.4	0.9	1.0	0.9	0.9	0.9	0.8	0.9	0.9	0.9	0.2	0.4	0.2	0.2	0.1	0.5	0.3	0.5	1.6	0.3	0.4
MINDE–j ($\sigma = 5$)	0.2	0.4	0.2	0.4	0.4	0.4	0.4	0.9	1.0	0.9	1.0	0.3	1.0	1.4	1.0	0.4	0.9	0.6	1.8	0.4	0.9	1.0	0.9	0.9	0.9	0.8	0.8	0.9	0.9	0.2	0.4	0.2	0.2	0.1	0.5	0.2	0.5	1.6	0.3	0.4
MINDE–j ($\sigma = 10$)	0.2	0.4	0.2	0.4	0.4	0.4	0.4	0.8	1.0	0.9	1.0	0.3	1.0	1.4	1.0	0.4	0.9	0.6	1.9	0.4	0.9	1.0	0.9	0.9	0.9	0.8	0.8	0.9	0.9	0.2	0.4	0.2	0.2	0.1	0.5	0.2	0.5	1.6	0.3	0.4
MINDE–c ($\sigma = 0.5$)	0.2	0.4	0.3	0.4	0.4	0.4	0.4	0.9	1.0	1.0	1.0	0.3	1.0	1.3	1.0	0.4	1.0	0.6	1.7	0.4	0.9	1.0	1.0	0.9	1.0	0.9	0.9	1.0	1.0	0.1	0.3	0.2	0.3	0.2	0.4	0.3	0.3	1.6	0.3	0.4
MINDE–c ($\sigma = 1$)	0.2	0.4	0.3	0.4	0.4	0.4	0.4	1.0	1.0	1.0	1.0	0.3	1.0	1.3	1.0	0.4	1.0	0.6	1.6	0.4	0.9	1.0	1.0	0.9	0.9	0.9	0.9	1.0	0.9	0.1	0.3	0.2	0.3	0.2	0.4	0.3	0.3	1.7	0.3	0.4
MINDE–c ($\sigma = 1.5$)	0.2	0.4	0.3	0.4	0.4	0.4	0.4	1.0	1.0	1.0	1.0	0.3	1.0	1.3	1.0	0.4	1.0	0.6	1.6	0.4	0.9	1.0	0.9	0.9	0.9	0.9	0.9	1.0	0.9	0.1	0.3	0.2	0.3	0.2	0.4	0.3	0.3	1.7	0.3	0.4
MINDE–c ($\sigma = 2$)	0.2	0.4	0.3	0.4	0.4	0.4	0.4	1.0	1.0	1.0	1.0	0.3	1.0	1.2	1.0	0.4	1.0	0.6	1.6	0.4	0.9	1.0	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.1	0.3	0.2	0.3	0.2	0.4	0.3	0.3	1.6	0.3	0.4
MINDE–c ($\sigma = 3$)	0.2	0.4	0.3	0.4	0.4	0.4	0.4	1.0	1.0	1.0	1.0	0.3	1.0	1.3	1.0	0.4	1.0	0.6	1.6	0.4	0.9	1.0	0.9	0.9	0.9	0.9	0.9	1.0	0.9	0.1	0.3	0.2	0.3	0.2	0.4	0.3	0.3	1.7	0.3	0.4
MINDE–c ($\sigma = 5$)	0.2	0.4	0.3	0.4	0.4	0.4	0.4	1.0	1.0	1.0	1.0	0.3	1.0	1.2	1.0	0.4	1.0	0.6	1.6	0.4	0.9	1.0	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.1	0.3	0.2	0.3	0.2	0.4	0.3	0.3	1.7	0.3	0.4
MINDE–c ($\sigma = 10$)	0.2	0.4	0.3	0.4	0.4	0.4	0.4	1.0	1.0	1.0	1.0	0.3	1.0	1.2	1.0	0.4	1.0	0.6	1.6	0.4	0.9	1.0	0.9	0.9	0.9	0.9	0.9	1.0	0.9	0.1	0.3	0.2	0.3	0.2	0.4	0.3	0.3	1.7	0.3	0.4
MINE	0.2	0.4	0.2	0.4	0.4	0.4	0.4	1.0	1.0	1.0	1.0	0.3	1.0	1.3	1.0	0.4	1.0	0.6	1.6	0.4	0.9	0.9	0.9	0.8	0.7	0.6	0.9	0.9	0.9	0.0	0.0	0.1	0.1	0.1	0.2	0.2	0.4	1.7	0.3	0.4
InfoNCE	0.2	0.4	0.3	0.4	0.4	0.4	0.4	1.0	1.0	1.0	1.0	0.3	1.0	1.3	1.0	0.4	1.0	0.6	1.6	0.4	0.9	1.0	1.0	0.8	0.8	0.8	0.9	1.0	1.0	0.2	0.3	0.2	0.3	0.2	0.4	0.3	0.4	1.7	0.3	0.4
D-V	0.2	0.4	0.3	0.4	0.4	0.4	0.4	1.0	1.0	1.0	1.0	0.3	1.0	1.3	1.0	0.4	1.0	0.6	1.6	0.4	0.9	1.0	1.0	0.8	0.8	0.8	0.9	1.0	1.0	0.0	0.0	0.1	0.1	0.2	0.2	0.2	0.4	1.7	0.3	0.4
NWJ	0.2	0.4	0.3	0.4	0.4	0.4	0.4	1.0	1.0	1.0	1.0	0.3	1.0	1.3	1.0	0.4	1.0	0.6	1.6	0.4	0.9	1.0	1.0	0.8	0.8	0.8	0.9	1.0	1.0	0.0	0.0	0.0	-0.6	0.1	0.1	0.2	0.4	1.7	0.3	0.4
KSG	0.2	0.4	0.2	0.2	0.4	0.4	0.4	0.2	0.9	0.7	1.0	0.3	0.2	1.1	1.0	0.4	0.7	0.6	1.3	0.4	0.2	0.9	0.7	0.2	0.7	0.6	0.2	0.9	0.7	0.2	0.2	0.1	0.1	0.1	0.2	0.2	0.4	1.7	0.3	0.4
LNN	0.2	0.9	2.7	6.6	0.4	0.4	0.4	17.3	2.7	6.4	1.3	0.6	17.3	17.3	3.1	2.5	7.3	6.8	33.5	0.4	17.3	2.5	7.3	17.3	3.1	7.3	17.3	2.4	7.2	-0.2	-0.6	0.5	1.2	2.1	2.7	5.5	0.3	1.5	0.3	0.4
CCA	0.0	0.0	0.0	0.0	0.3	0.4	0.4	1.0	1.0	1.0	1.0	0.3	1.1	1.3	1.0	0.4	1.0	0.6	1.8	0.4	1.0	1.0	1.0	0.9	0.2	0.4	1.0	0.8	0.9	0.3	1.0	0.1	0.1	0.0	0.3	0.0	0.0	1.6	0.2	0.4

Table B.3: MINDE–j and MINDE–c σ ablations study .Mean MI estimates over 10 seeds using N = 10000 samples compared each against the ground-truth. Color indicates relative negative bias (red) and positive bias (blue). Our method and all neural estimators were trained with 100k training samples. List of abbreviations (*Mn*: Multinormal, *St*: Student-t, *Nm*: Normal, *Hc*: Half-cube, *Sp*: Spiral)

B.3.2 Full results with standard deviation

We report in Table B.4 mean results without quantization for the different methods. Figures B.1 and B.2 contains box-plots for all the competitors and all the tasks.

dist	Asinh @ *St* 1 × 1 (dof=1)	Asinh @ *St* 2 × 2 (dof=1)	Asinh @ *St* 3 × 3 (dof=2)	Asinh @ *St* 5 × 5 (dof=2)	Bimodal 1 × 1	Bivariate *Nm* 1 × 1	*Hc* @ Bivariate *Nm* 1 × 1	*Hc* @ *Mn* 25 × 25 (2-pair)	*Hc* @ *Mn* 3 × 3 (2-pair)	*Hc* @ *Mn* 5 × 5 (2-pair)	*Mn* 2 × 2 (2-pair)	*Mn* 2 × 2 (dense)	*Mn* 25 × 25 (2-pair)	*Mn* 25 × 25 (dense)	*Mn* 3 × 3 (2-pair)	*Mn* 3 × 3 (dense)	*Mn* 5 × 5 (2-pair)	*Mn* 5 × 5 (dense)	*Mn* 50 × 50 (dense)	*Nm* CDF @ Bivariate *Nm* 1 × 1	*Nm* CDF @ *Mn* 25 × 25 (2-pair)	*Nm* CDF @ *Mn* 3 × 3 (2-pair)	*Nm* CDF @ *Mn* 5 × 5 (2-pair)	*Sp* @ *Mn* 25 × 25 (2-pair)	*Sp* @ *Mn* 3 × 3 (2-pair)	*Sp* @ *Mn* 5 × 5 (2-pair)	*Sp* @ *Nm* CDF @ *Mn* 25 × 25 (2-pair)	*Sp* @ *Nm* CDF @ *Mn* 3 × 3 (2-pair)	*Sp* @ *Nm* CDF @ *Mn* 5 × 5 (2-pair)	*St* 1 × 1 (dof=1)	*St* 2 × 2 (dof=1)	*St* 2 × 2 (dof=2)	*St* 3 × 3 (dof=2)	*St* 3 × 3 (dof=3)	*St* 5 × 5 (dof=2)	*St* 5 × 5 (dof=3)	Swiss roll 2 × 1	Uniform 1 × 1 (additive noise=.1)	Uniform 1 × 1 (additive noise=.75)	Wiggly @ Bivariate *Nm* 1 × 1
GT	0.22	0.43	0.29	0.45	0.41	0.41	0.41	1.02	1.02	1.02	1.02	0.29	1.02	1.29	1.02	0.41	1.02	0.59	1.62	0.41	1.02	1.02	1.02	1.02	1.02	1.02	1.02	1.02	1.02	0.22	0.43	0.19	0.29	0.18	0.45	0.30	0.41	1.71	0.33	0.41
MINDE–j ($\sigma = 1$)	0.21	0.40	0.26	0.40	0.41	0.41	0.41	1.10	1.01	1.00	1.01	0.29	0.91	1.18	1.00	0.42	1.00	0.59	1.72	0.40	0.96	0.96	0.99	0.87	0.90	0.90	0.95	0.95	0.98	0.17	0.35	0.18	0.25	0.17	0.47	0.29	0.49	1.65	0.31	0.41
MINDE–j	0.22	0.42	0.28	0.42	0.41	0.42	0.42	1.19	1.02	1.02	1.02	0.29	0.99	1.31	1.02	0.41	1.01	0.59	1.73	0.41	1.07	0.99	0.99	0.95	0.92	0.93	1.07	0.99	0.98	0.13	0.24	0.20	0.30	0.18	0.48	0.31	0.40	1.67	0.31	0.42
MINDE–c ($\sigma = 1$)	0.21	0.42	0.27	0.42	0.41	0.41	0.41	0.96	1.00	0.99	1.01	0.29	1.00	1.26	1.01	0.41	1.01	0.59	1.62	0.40	0.94	0.97	0.95	0.92	0.94	0.93	0.93	0.96	0.94	0.13	0.28	0.18	0.29	0.18	0.42	0.30	0.32	1.67	0.31	0.41
MINDE–c	0.21	0.42	0.28	0.42	0.41	0.41	0.41	1.00	1.01	1.01	1.01	0.29	1.00	1.27	1.01	0.41	1.01	0.59	1.60	0.40	0.98	0.99	0.98	0.92	0.94	0.94	0.98	0.98	0.98	0.14	0.26	0.19	0.28	0.17	0.44	0.29	0.40	1.66	0.31	0.41
MINE	0.23	0.38	0.24	0.36	0.40	0.41	0.41	0.96	0.99	0.98	1.01	0.30	0.99	1.28	1.01	0.41	1.00	0.59	1.60	0.39	0.88	0.90	0.90	0.81	0.70	0.65	0.88	0.89	0.87	0.02	0.01	0.12	0.12	0.13	0.16	0.22	0.39	1.66	0.32	0.41
InfoNCE	0.22	0.41	0.27	0.40	0.41	0.41	0.41	0.98	1.01	1.01	1.02	0.29	0.99	1.28	1.01	0.41	1.02	0.59	1.61	0.40	0.92	0.98	0.99	0.83	0.84	0.82	0.92	0.96	0.96	0.15	0.30	0.18	0.27	0.17	0.41	0.28	0.40	1.69	0.32	0.41
D-V	0.22	0.41	0.27	0.40	0.41	0.41	0.41	0.98	1.01	1.01	1.02	0.29	0.99	1.28	1.01	0.41	1.02	0.59	1.61	0.40	0.93	0.98	0.99	0.82	0.82	0.81	0.92	0.96	0.96	0.01	0.05	0.11	0.13	0.15	0.22	0.21	0.40	1.69	0.32	0.41
NWJ	0.22	0.41	0.27	0.40	0.41	0.41	0.41	0.98	1.01	1.01	1.02	0.29	0.99	1.28	1.01	0.41	1.02	0.59	1.60	0.40	0.93	0.98	0.98	0.82	0.82	0.80	0.92	0.95	0.96	0.03	0.02	0.04	-0.65	0.12	0.12	0.21	0.40	1.69	0.32	0.41
KSG	0.22	0.38	0.19	0.24	0.42	0.42	0.42	0.17	0.87	0.66	1.03	0.29	0.20	1.07	0.95	0.41	0.74	0.57	1.28	0.42	0.20	0.92	0.72	0.18	0.71	0.55	0.20	0.90	0.69	0.16	0.22	0.09	0.12	0.07	0.20	0.15	0.42	1.68	0.32	0.42
LNN	0.25	0.89	2.71	6.65	0.41	0.42	0.42		2.68	6.45	1.27	0.65			3.10	2.49	7.31	6.76		0.39		2.49	7.27		3.10	7.31		2.38	7.24			0.53	1.20	2.12	2.74	5.48	0.34	1.48	0.29	0.42
CCA	0.00	0.00	0.00	0.00	0.34	0.41	0.38	0.99	0.95	0.96	1.02	0.29	1.06	1.33	1.02	0.41	1.02	0.60	1.75	0.39	1.00	0.97	0.96	0.86	0.23	0.39	0.99	0.84	0.90	0.33	0.95	0.11	0.13	0.01	0.31	0.02	0.02	1.63	0.19	0.38
DoE(Gaussian)	0.16	0.48	0.27	0.57	0.37	0.37	0.39	0.67	0.97	0.96	0.95	0.35	0.67	7.83	0.95	0.64	0.94	1.27	16.11	0.37	0.70	0.99	0.95	0.48	0.58	0.56	0.57	0.74	0.77	6.74	7.93	1.78	2.54	0.61	4.24	1.17	1.57	0.11	0.38	
DoE(Logistic)	0.13	0.37	0.21	0.43	0.41	0.35	0.36	0.62	0.92	0.92	0.95	0.34	0.69	7.83	0.95	0.63	0.93	1.27	16.15	0.41	0.78	1.08	1.05	0.47	0.60	0.55	0.67	0.79	0.81	-0.32	2.00	0.46	0.82	0.29	1.48	0.59	1.58	0.08	0.35	

Table B.4: Mean estimate over 10 seeds using N = 10000 samples compared each against the ground-truth.
Our method and all neural estimators were trained with 100k training samples.
List of abbreviations (*Mn*: Multinormal, *St*: Student-t, *Nm*: Normal, *Hc*: Half-cube, *Sp*: Spiral)

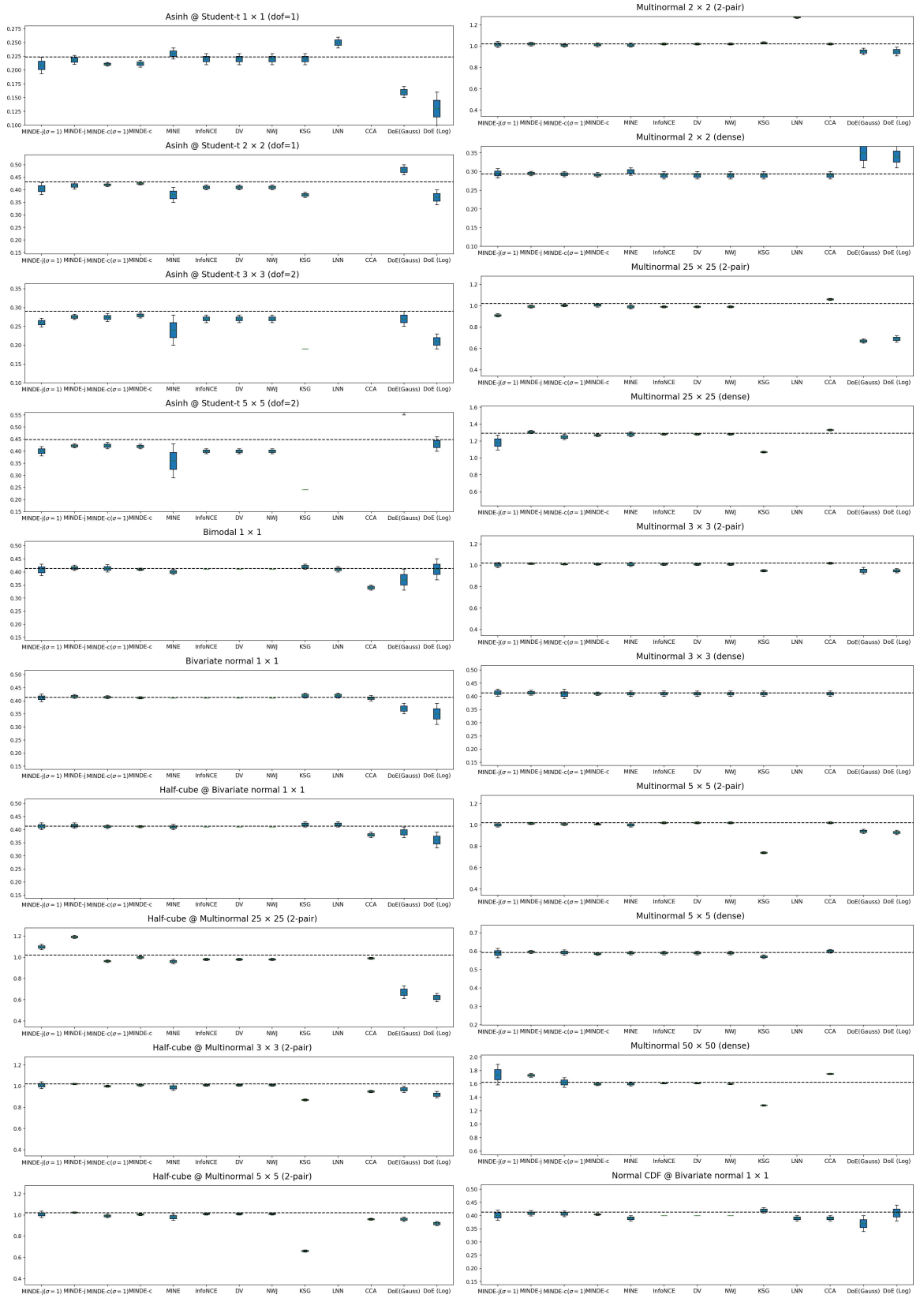


Figure B.1: We report MI estimate results over 10 seeds for N =10000 for our method and competitors for training size 100k sample. A method absent from the depiction implies either non convergence during training or results out of scale

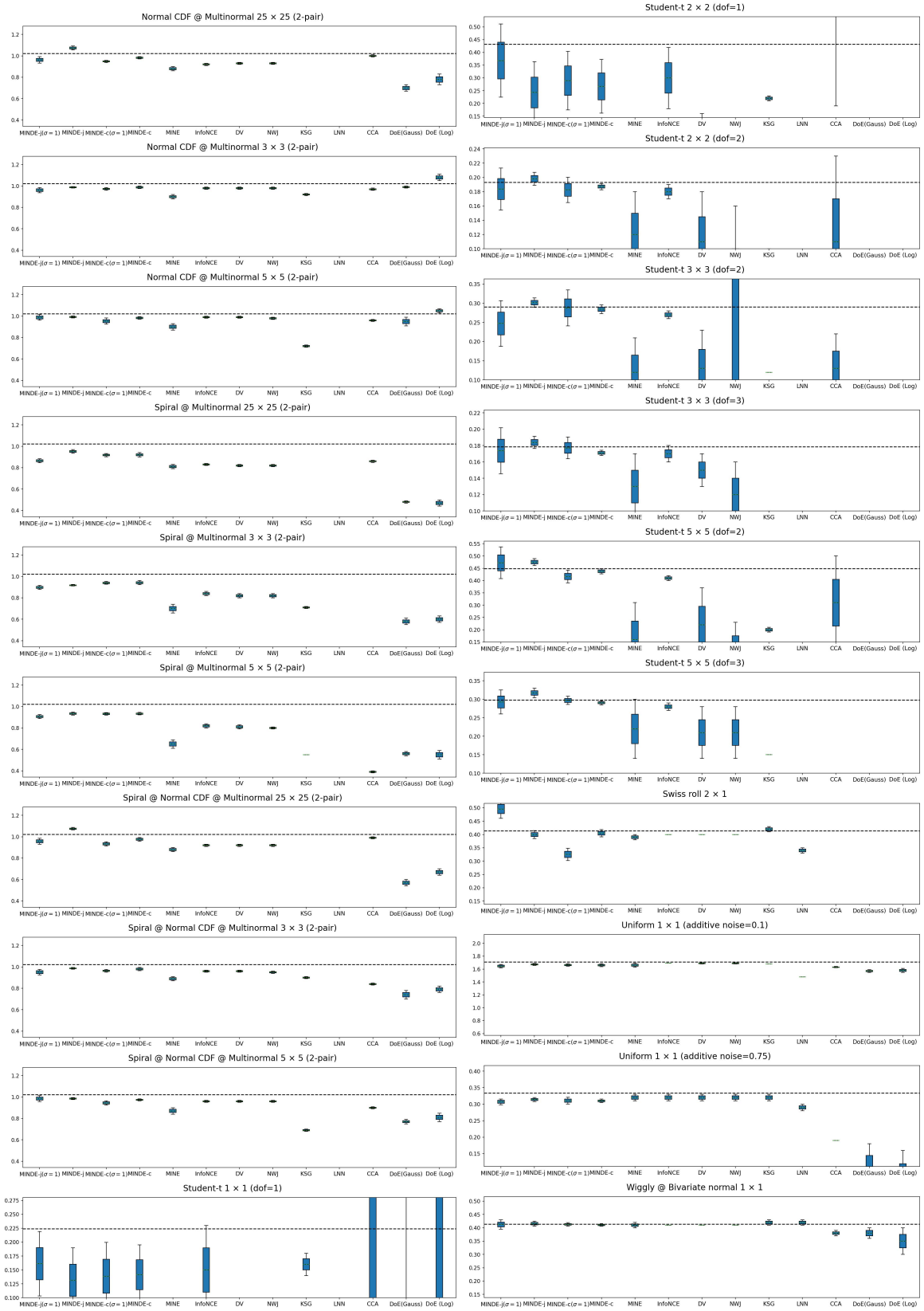


Figure B.2: We report MI estimate results over 10 seeds for N =10000 for our method and competitors for training size 100k sample.

B.3.3 Training size ablation study

We here report, in Figures B.3 to B.6 the results of our ablation study on the training size, varying in the range 5k,10k,50k,100k.

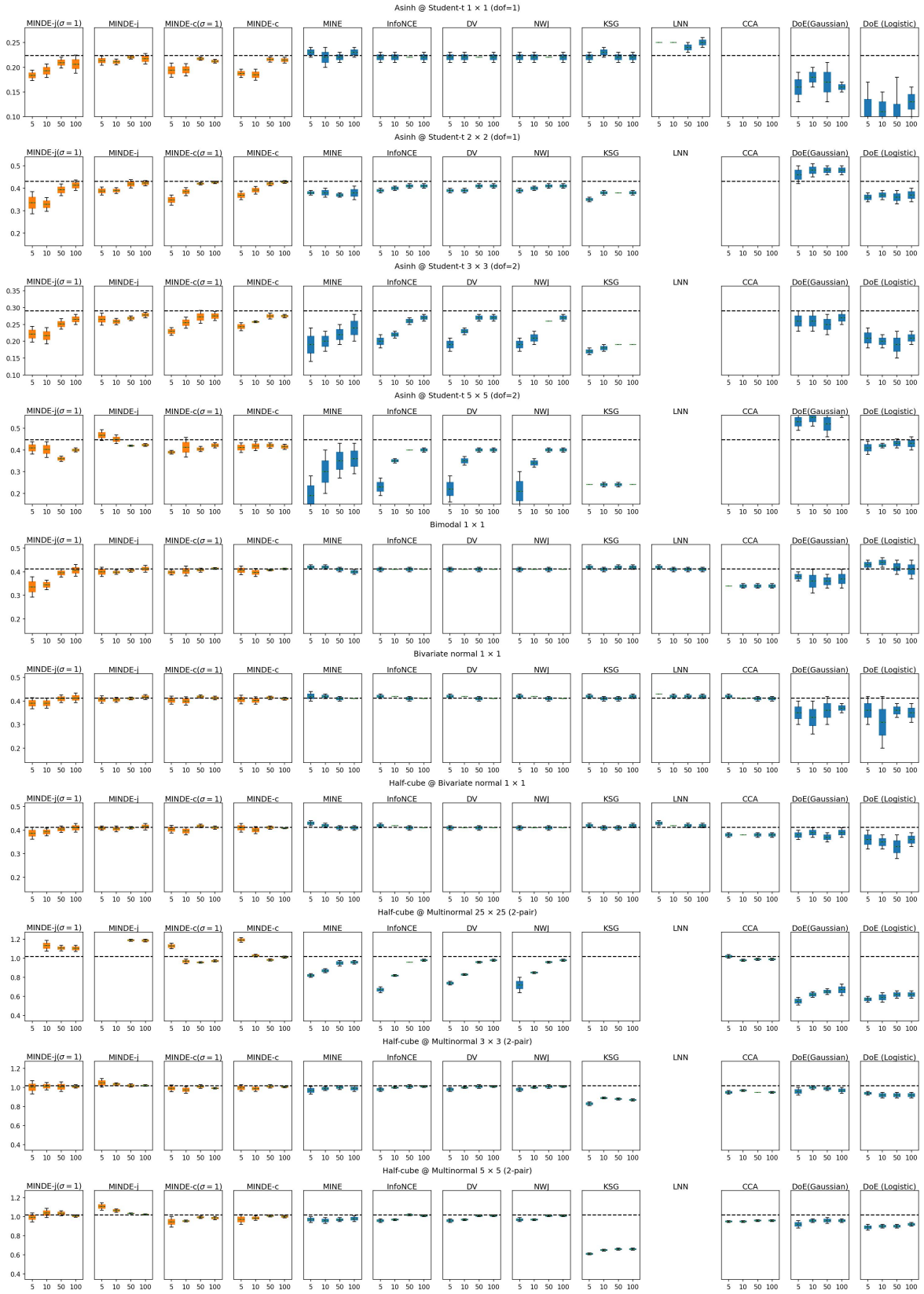


Figure B.3: Training Size ablation study : We report MI estimate results for our method and competitors as a function of the training size used (5k,10k,50k,100k). For readability, we discard the baselines with estimation (error > 2 * GT) or high standard deviation. All results are averaged over 5 seeds. Due the benchmark size, we split the results into 4 figures each containing 10 benchmarks. A method absent from the depiction implies either non convergence during training or results out of scale. In this first plot we report tasks 1-10.

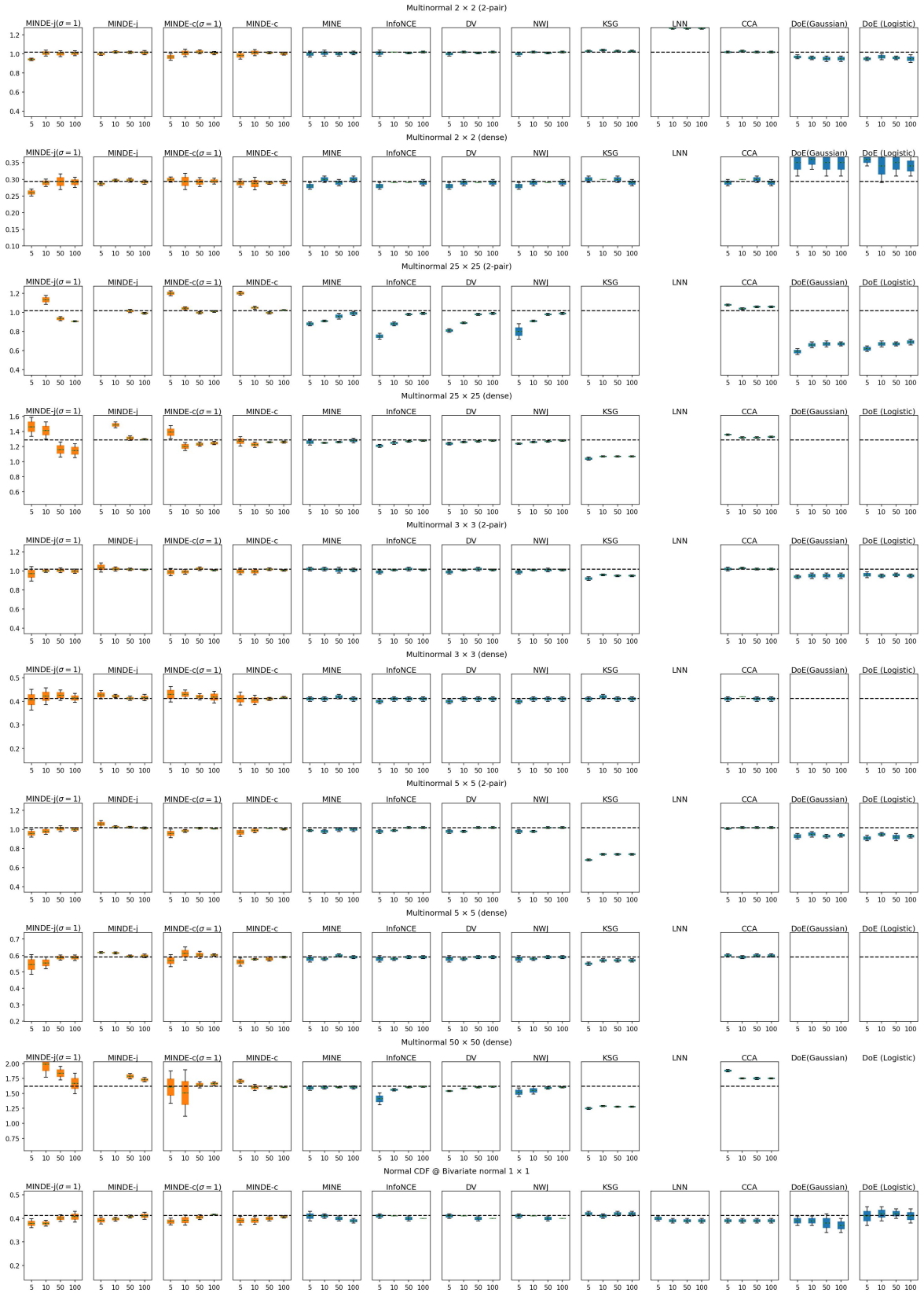


Figure B.4: Part 2 of Figure B.3, tasks 11-20.

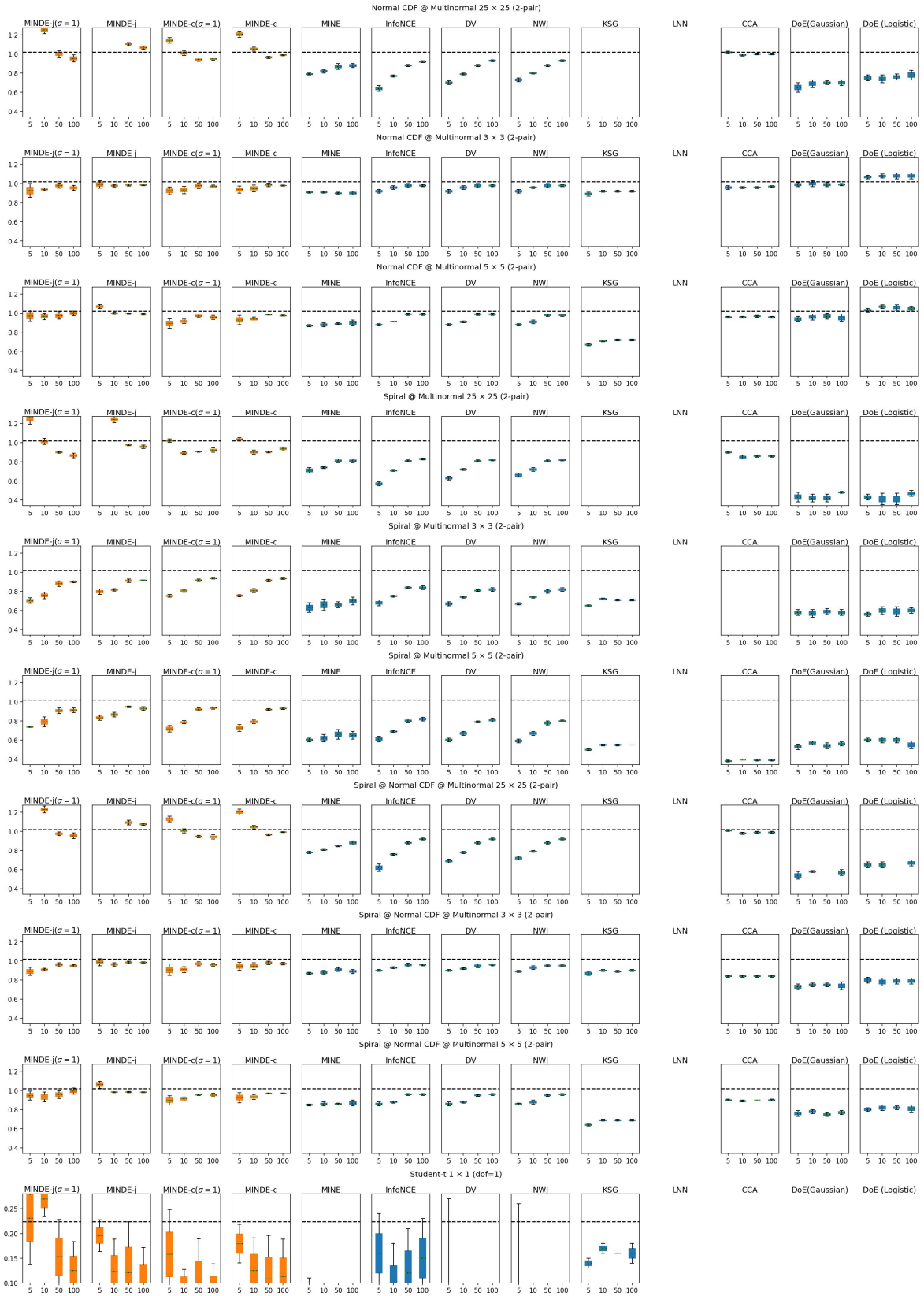


Figure B.5: Part 3 of Figure B.3, tasks 21-30.

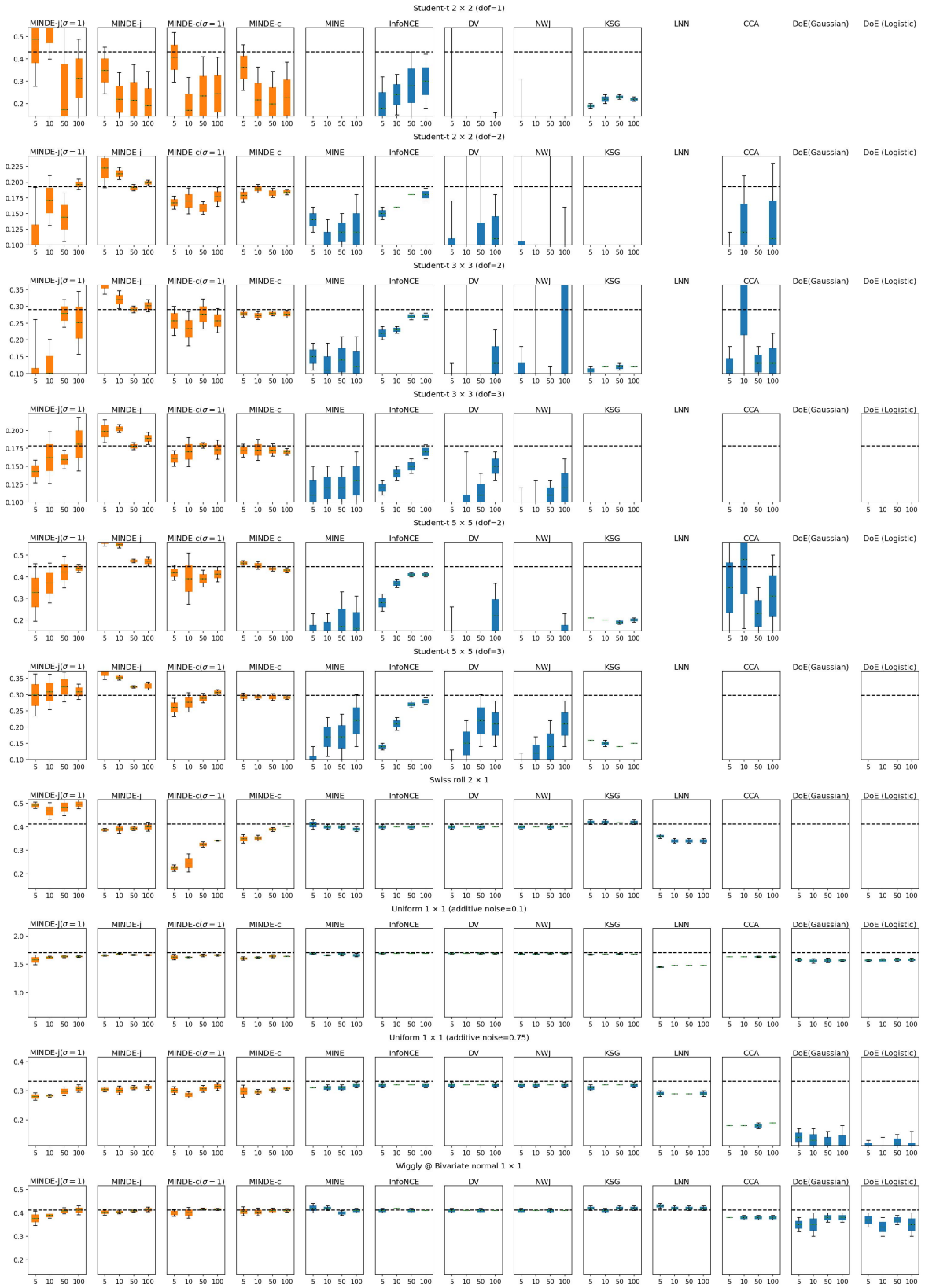


Figure B.6: Part 4 of Figure B.3, tasks 31-40.

Appendix C

Appendix for Chapter 5

C.1 Proofs

C.1.1 Detailed proof of Proposition 2

Here we provide the full proof for Proposition 2 (to avoid unnecessary complications, we assume the 1-d case, the vector proof is identical). Starting from the equation :

$$C = \int \frac{dp_t}{dt} \log\left(\frac{p_t}{q_t}\right) + p_t \frac{d}{dt} \log\left(\frac{p_t}{q_t}\right) dxdt$$

Concerning the first part of the integral:

$$\int \frac{dp_t}{dt} \log\left(\frac{p_t}{q_t}\right) dxdt = \int \Delta(p_t) \log\left(\frac{p_t}{q_t}\right) dxdt = \int p_t \Delta(\log\left(\frac{p_t}{q_t}\right)) dxdt,$$

Where the first equality is simply due to $\frac{dp_t}{dt} = \Delta p_t$, and the second is obtained by properties of the adjoint of the Δ operator. In particular, we need to perform a double application of integration by parts, where we should remember that densities p_t, q_t are equal to zero at infinite values of x and that $\Delta = \nabla\nabla$.

Focusing on the second part of the integral:

$$\int p_t \frac{d}{dt} \log\left(\frac{p_t}{q_t}\right) dxdt = \int p_t \left(\frac{d \log p_t}{dt} - \frac{d \log q_t}{dt}\right) dxdt = \int p_t \left(\frac{\frac{dp_t}{dt}}{p_t} - \frac{\frac{dq_t}{dt}}{q_t}\right) dxdt$$

The first summand $p_t \frac{\frac{dp_t}{dt}}{p_t}$ simplifies to $\frac{dp_t}{dt}$.

Since $\int \frac{dp_t}{dt} dxdt = \int \frac{d}{dt}(\int p_t dx)dt = \int \frac{d}{dt}(1)dt = 0$, this term is cancelled.

The second is transformed as :

$p_t \frac{\frac{dq_t}{dt}}{q_t} = \frac{p_t}{q_t}\frac{dq_t}{dt} = \frac{p_t}{q_t}\Delta q_t$ where again we leveraged $\frac{dq_t}{dt} = \Delta q_t$.

Consequently, we obtain:

$$C = \int p_t \Delta \log\left(\frac{p_t}{q_t}\right) - \frac{p_t}{q_t}\Delta q_t dxdt$$

We apply one step of integration by parts on both Δ operators and obtain :

$$\int -\nabla p_t \nabla \log\left(\frac{p_t}{q_t}\right) + \nabla(\frac{p_t}{q_t})\nabla q_t dxdt$$

The remaining missing clarification in the sketch proof of Proposition 2 is that :

$$\nabla(\frac{p_t}{q_t})\nabla(q_t) = \frac{\nabla(p_t)q_t - \nabla(q_t)p_t}{q_t^2}\nabla(q_t) =$$
$$\frac{\nabla(p_t)}{q_t}\nabla(q_t) - p_t(\frac{\nabla q_t}{q_t})^2 = \nabla p_t \nabla(\log(q_t)) - p_t(\nabla(\log q_t))^2 =$$
$$p_t \nabla(\log p_t)\nabla(\log(q_t)) - p_t(\nabla(\log q_t))^2 = p_t \nabla(\log q_t)(\nabla(\log p_t) - \nabla(\log q_t)) = p_t \nabla(\log q_t)(\nabla(\log \frac{p_t}{q_t}))$$

C.1.2 TC and DTC equivalences

We here prove the equivalences about TC and DTC. Starting from TC :

$$\begin{aligned}\sum_{i=1}^{N} \mathcal{H}(\mathbf{X}^i) - \mathcal{H}(\mathbf{X}) &= \sum_{i=1}^{N} \mathcal{H}(\mathbf{X}^i) - \sum_{i=1}^{N} \mathcal{H}(\mathbf{X}^i \,|\, \mathbf{X}^{>i}) \\ &= \sum_{i=1}^{N-1} \mathcal{I}(\mathbf{X}^i; \mathbf{X}^{>i}) = \mathcal{T}(\mathbf{X})\end{aligned}$$

Concerning DTC:

$$\begin{aligned}
\mathcal{H}(\mathbf{X}) - \sum_{i=1}^{N} \mathcal{H}(\mathbf{X}^i|\mathbf{X}^{\setminus i}) &= \mathcal{H}(\mathbf{X}^1) + \mathcal{H}(\mathbf{X}^{\setminus 1} \,|\, \mathbf{X}^1) - \mathcal{H}(\mathbf{X}^1|\mathbf{X}^{\setminus 1}) - \sum_{i=2}^{N} \mathcal{H}(\mathbf{X}^i|\mathbf{X}^{\setminus i}) \\
&= \mathcal{I}(\mathbf{X}^1; \mathbf{X}^{\setminus 1}) + \mathcal{H}(\mathbf{X}^{\setminus 1} \,|\, \mathbf{X}^1) - \sum_{i=2}^{N} \mathcal{H}(\mathbf{X}^i|\mathbf{X}^{\setminus i}) \\
&= \mathcal{I}(\mathbf{X}^1; \mathbf{X}^{\setminus 1}) + \mathcal{H}(\mathbf{X}^2 \,|\, \mathbf{X}^1) + \mathcal{H}(\mathbf{X}^{\setminus\{1,2\}} \,|\, \mathbf{X}^1, \mathbf{X}^2) - \mathcal{H}(\mathbf{X}^2|\mathbf{X}^{\setminus 2}) \\
&\quad - \sum_{i=3}^{N} \mathcal{H}(\mathbf{X}^i|\mathbf{X}^{\setminus i}) \\
&= \mathcal{I}(\mathbf{X}^1; \mathbf{X}^{\setminus 1}) + \mathcal{I}(\mathbf{X}^2; \mathbf{X}^{>2}|\mathbf{X}^{<2}) + \mathcal{H}(\mathbf{X}^{\setminus\{1,2\}} \,|\, \mathbf{X}^1, \mathbf{X}^2) - \sum_{i=3}^{N} \mathcal{H}(\mathbf{X}^i|\mathbf{X}^{\setminus i}) \\
&= \cdots = \sum_{i=1}^{N-1} \mathcal{I}(\mathbf{X}^i; \mathbf{X}^{>i} \,|\, \mathbf{X}^{<i}) = \mathcal{D}(\mathbf{X})
\end{aligned}$$

Where for the last equality it suffices to consider trivial reordering arguments:

$$\sum_{i=2}^{N} \mathcal{I}(\mathbf{X}^i; \mathbf{X}^{<i} \,|\, \mathbf{X}^{>i}) = \sum_{i=1}^{N-1} \mathcal{I}(\mathbf{X}^i; \mathbf{X}^{>i} \,|\, \mathbf{X}^{<i})$$

C.2 Details of SΩI

In the section we provide additional implementation details about SΩI.

C.2.1 Computing O-information

In § 5.3.1, we presented how TC and DTC can be estimated using denoising score functions. Our estimators requires different score functions which can be obtained by learning different denoisers. More particularly, TC requires the joint denoiser $\mathbb{E}[\mathbf{X} \,|\, \mathbf{X}_t]$ and the marginals $\mathbb{E}[\mathbf{X}^i \,|\, \mathbf{X}^i_t]$ for $i \in \{1, \dots, N\}$. DTC estimation is obtained using the joint and the following conditional terms $\mathbb{E}[\mathbf{X}^i \,|\, \mathbf{X}^i_t, \mathbf{X}^{\setminus i}]$ for $i \in \{1, \dots, N\}$. Our formulation in § 5.3.1 is general and can be applied to a wide range of denoising score learning techniques. For the implementation of SΩI, we adopt VP-SDE framework (Song and Ermon, 2019). The latter perturbs the data using an SDE parameterized by a drift $f(t)$ and a diffusion coefficient $g(t)$.

Muti-variate denoising score network. We extend the work from (Bounoua, Franzese, and Michiardi, 2024) to amortize the learning of all the required terms using a **unique** denoising score network. The denoising score network ϵ_θ accepts as input the concatenation of the variables each perturbed at different times. The second input is a vector of size N which describes the state of each variable and allows a parametrization of different denoising score functions.

The joint term corresponds to the case where all the variables are perturbed with the same intensity t and all the elements of the vector $\tau = [t, \dots, t]$ are set equivalently to t. The conditional terms correspond to the case where only the conditioned variable i is perturbed with intensity t whereas the remaining conditioning variables $\setminus i_{\text{th}}$ are kept unperturbed at $t = 0$. Consequently the parameter describing this case is of the form $[0, \dots, t, \dots, 0]$.

While (Bounoua, Franzese, and Michiardi, 2024) framework is not able to learn the marginal denoising score, it's possible via an additional parameterization to include this configuration. This corresponds to the case where the marginal variable i is perturbed with intensity t while all the other variables are made uninformative. The non marginal variables $\setminus i_{\text{th}}$ are replaced with pure noise corresponding to a maximal perturbation at $t = T$. Consequently the parameter describing this case is of the form $[T, \dots, t, \dots, T]$.

Training. The training is carried out through a randomized procedure. At each training step, we select randomly a set of the denoising score functions required for the O-information estimation (joint, conditional or marginals). These denoising scores function are learned by the unique network following Algorithm 10. In total, estimating O-information requires calling $2N + 1$ denoising score functions which we learn using a unique denoising network.

Inference. Once all the denoising score functions are learned, it's possible to estimate TC and DTC via a Monte Carlo estimation of the integral over t in Proposition 3 and Proposition 5 . The outer integration w.r.t. to the time instant is possible by sampling $t \sim \mathcal{U}(0, T)$, and then using the estimation $\int_0^T (\cdot)\mathrm{d}t = T\mathbb{E}_{t\sim\mathcal{U}(0,T)}[(\cdot)]$. In practice we adopt 10 steps for the computation of the expectation. The procedure to estimate O-information is described in algorithm 11. First, samples from $\mathbf{x} \sim p(\mathbf{x})$ are considered, then sampling the time $t \sim \mathcal{U}[0, T]$. A perturbed version of the variables $\mathbf{X}_t$ is computed using the VPSDE. The joint, conditional and marginal denoising scores are computed leveraging the unique denoising score network. This is possible by choosing different perturbation times and manipulating the vector τ as described earlier. Computing the difference of the denoising scores functions (see Proposition 3 and Proposition 3) allows the computation of TC and DTC respectively. Please note that it is possible to implement importance sampling schemes

Algorithm 10: SΩI Training step

Data: $\mathbf{X} = \{\mathbf{X}^i\}_{i=1}^N$
$t \sim \mathcal{U}[0, T]$ // Importance sampling schemes (Huang, Lim, and Courville, 2021; Song et al., 2021) can be adopted to reduce variance
if *Joint* **then**
  $\mathbf{X}_t \sim p_{0t}$ // Obtain noisy version of all the variables using VPSDE (Song and Ermon, 2019) with drift $f(t)$ and diffusion coefficient $g(t)$.
  $s(\mathbf{X}_t) = \epsilon_\theta([\mathbf{X}_t^1, \dots, \mathbf{X}_t^N], \tau = [t, \dots, t, \dots, t])$
  Return $\nabla_\theta \| s_t(\mathbf{X}_t) - \nabla \log p_{0t}(\mathbf{X}_t|\mathbf{X}) \|$ // Denoising score matching of all the variables
if *Conditional* **then**
  $\mathbf{X}_t^i \sim p_{0t}$ // Obtain noisy version of the variable i while the remaining variables are kept unperturbed at $(t = 0)$
  $s(\mathbf{X}_t^i|\mathbf{X}^{\setminus i}) = \epsilon_\theta\left([\mathbf{X}^1, \dots, \mathbf{X}^{i-1}, \mathbf{X}_t^i, \mathbf{X}^{i+1}, \dots, \mathbf{X}^N], \tau = [0, \dots, t, \dots, 0]\right)$
  Return $\nabla_\theta \left\| s(\mathbf{X}_t^i|\mathbf{X}^{\setminus i}) - \nabla \log p_{0t}(\mathbf{X}_t^i|\mathbf{X}^i) \right\|$ // Denoising score matching of the conditioning variable i
if *Marginal* **then**
  $\mathbf{X}_t^i \sim p_{0t}$
  $\mathbf{X}_T^{\setminus i} \leftarrow p_T = \mathcal{N}(0, \mathbb{I})$ // Obtain noisy version of the variable i while the remaining variables are replaced with pure noise $(t = T)$.
  $s(\mathbf{X}_t^i) = \epsilon_\theta([\mathbf{X}_T^1, \dots, \mathbf{X}_T^{i-1}, \mathbf{X}_t^i, \mathbf{X}_T^{i+1}, \dots, \mathbf{X}_T^N], \tau = [T, \dots, t, \dots, T])$
  Return $\nabla_\theta \| s(\mathbf{X}_t^i) - \nabla \log p_{0t}(\mathbf{X}_t^i|\mathbf{X}^i) \|$ // Denoising score matching of the marginal variable i

to reduce the variance, along the lines of what described by Huang, Lim, and Courville (2021).

C.2.2 Computing gradient of O-information

To compute the gradient of O-information recall that $\partial_i \Omega(\mathbf{X}) = \Omega(\mathbf{X}) - \Omega(\mathbf{X}^{\setminus i})$. The first order gradient of O-information requires the estimation of O-information of all the subsystems of size $N - 1$.

Algorithm 11: SΩI inference time

Data: $\mathbf{X} = \{\mathbf{X}^i\}_{i=1}^N$
$t \sim \mathcal{U}[0, T]$ // Importance sampling scheme can also be adopted
$\mathbf{X}_t \sim p_{0t}$ // Obtain the noisy version of all the variables using VPSDE (Song and Ermon, 2019) with a diffusion coefficient $g(t)$.

$s(\mathbf{X}_t) \leftarrow \epsilon_\theta([\mathbf{X}_t^1, \dots, \mathbf{X}_t^N], \tau = [t, \dots, t, \dots, t])$ // Compute the joint score

for *i = 1* ***to*** N // Compute the conditional and marginal terms
do
  $s(\mathbf{X}_t^i|\mathbf{X}^{\setminus i}) \leftarrow \epsilon_\theta\left([\mathbf{X}^1, \dots, \mathbf{X}^{i-1}, \mathbf{X}_t^i, \mathbf{X}^{i+1}, \dots, \mathbf{X}^N], \tau = [0, \dots, t, \dots, 0]\right)$
  $s(\mathbf{X}_t^i) \leftarrow \epsilon_\theta([\mathbf{X}_T^1, \dots, \mathbf{X}_T^{i-1}, \mathbf{X}_t^i, \mathbf{X}_T^{i+1}, \dots, \mathbf{X}_T^N], \tau = [T, \dots, t, \dots, T])$
  // Similarly to Algorithm 10 the non marginal variables are replaced with pure noise $\mathbf{X}_T^{\setminus i} \sim \mathcal{N}(0, \mathbb{I})$
end

$\hat{\mathcal{T}}(\mathbf{X}) \leftarrow \frac{g^2(t)}{2}\left\| s(\mathbf{X}_t) - [s(\mathbf{X}_t^i)]_{i=1}^N \right\|^2$ // See Proposition 3
$\hat{\mathcal{D}}(\mathbf{X}) \leftarrow \frac{g^2(t)}{2}\left\| s(\mathbf{X}_t) - \left[s_t(\mathbf{X}_t^i|\mathbf{X}^{\setminus i})\right]_{i=1}^N \right\|^2$ // See Proposition 5
$\hat{\Omega}(\mathbf{X}) \leftarrow \hat{\mathcal{T}}(\mathbf{X}) - \hat{\mathcal{D}}(\mathbf{X})$
Return $\hat{\Omega}(\mathbf{X})$

$$\Omega(\mathbf{X}^{\setminus i}) = \mathcal{T}(\mathbf{X}^{\setminus i}) - \mathcal{D}(\mathbf{X}^{\setminus i}) \tag{C.1}$$

$$= \sum_{j=1, j\neq i}^{N} \mathcal{H}(\mathbf{X}^j) - \mathcal{H}(X^{\setminus i}) \tag{C.2}$$

$$- (\mathcal{H}(\mathbf{X}^{\setminus i}) - \sum_{j=1, j\neq i}^{N} \mathcal{H}(\mathbf{X}^j|\mathbf{X}^{\setminus\{i,j\}})) \tag{C.3}$$

It's possible to use an alternative formulation to estimate the gradient of O-information based on MI terms:

$$\partial_i \Omega(\mathbf{X}) = (2-N)\mathcal{I}(\mathbf{X}^i, \mathbf{X}^{\setminus i}) + \sum_{j=1, j\neq i}^{N} \mathcal{I}(\mathbf{X}^i, \mathbf{X}^{\setminus\{i,j\}}) \tag{C.4}$$

$$= (2-N)\left[\mathcal{H}(\mathbf{X}^i) - \mathcal{H}(\mathbf{X}^i|\mathbf{X}^{\setminus i})\right] + \sum_{j=1, j\neq i}^{N} \mathcal{H}(\mathbf{X}^i) - \mathcal{H}(\mathbf{X}^i|\mathbf{X}^{\setminus\{i,j\}}) \tag{C.5}$$

Many denoising score functions in Eq. (C.3) were also used to estimate the global O-information. To learn the additional necessary terms to compute $\Omega(X^{\setminus i})$, the randomized set of scores adopted during the training step (see Appendix C.2.1) is extended to account for the new requirements. Please note that we still use a unique denoising network that considers all the terms necessary to compute O-information and its gradient. A large number of learned denoising score functions is a potential reason for the bias observed in our experiment Figure 5.4. A highly flexible architecture capable of fitting large number of scores may be needed to infer gradient of O-information.

C.3 Experimental settings

C.3.1 Canonical multivariate Gaussian system

In this section we provide additional details about the construction of the synthetic benchmark § 5.4.1.

Redundancy benchmark. All the variable of the system are composed of a redundant component and unique information specific to each variable.

We modulate the redundant inter-dependency strength by setting different values for σ. We consider a standardized system where all the variables mean is 0 and standard deviation equal to $\mathbb{I}$. This results in the following covariance matrix:

$$\begin{bmatrix} \mathbb{I} & \rho\mathbb{I} & \vdots & \rho\mathbb{I} \\ \rho\mathbb{I} & \mathbb{I} & \dots & \rho\mathbb{I} \\ \vdots & \vdots & \ddots & \rho\mathbb{I} \\ \rho\mathbb{I} & \rho\mathbb{I} & \dots & \mathbb{I} \end{bmatrix} \tag{C.6}$$

With $\rho = \frac{1}{1+\sigma^2}$ which modulates the interactions strength in the system.

Synergy benchmark. We consider a standardized system where all the variables mean is 0 and standard deviation equal to $\mathbb{I}$. This results in the following covariance matrix :

$$\begin{bmatrix} \mathbb{I} & \frac{1}{\sqrt{N-1}}\mathbb{I} & 0 & \dots & 0 \\ \frac{1}{\sqrt{N-1}}\mathbb{I} & \mathbb{I} & \frac{\rho}{\sqrt{N-1}}\mathbb{I} & \dots & \frac{\rho}{\sqrt{N-1}}\mathbb{I} \\ 0 & \frac{\rho}{\sqrt{N-1}}\mathbb{I} & \mathbb{I} & \dots & 0 \\ 0 & \vdots & 0 & \ddots & 0 \\ 0 & \frac{\rho}{\sqrt{N-1}} & 0 & \dots & \mathbb{I} \end{bmatrix} \tag{C.7}$$

Where $\rho = \frac{1}{\sqrt{1+\sigma^2}}$ modulates the interactions strength in the system.

Mixed benchmark. The covariance matrix is easy to obtain as the mixed benchmark is made of independent subsystems.

Ground Truth. Having access to the covariance matrix of the system, computing entropy in close form for Gaussian distribution is possible. For $\mathbf{X} \sim \mathcal{N}(\mu, \sigma)$:

$$\mathcal{H}(\mathbf{X}) = \frac{1}{2}\log\left(2\pi\sigma^2\right) + \frac{1}{2} \tag{C.8}$$

For a multivariate Gaussian distribution $\mathbf{X}^d \sim \mathcal{N}_d(\mu, \Sigma)$:

$$\mathcal{H}(\mathbf{X}) = \frac{D}{2}(1 + \log(2\pi)) + \frac{1}{2}\log det(\Sigma) \tag{C.9}$$

C.3.2 SΩI implementation details

We provide code-base for SΩI implementation at [1]. The training of SΩI is carried out using *Adam optimizer* (Kingma and Ba, 2014). We use Exponential moving average (EMA) with a momentum parameter $m = 0.999$. Importance sampling (Huang, Lim, and Courville, 2021) [2] at train and test-time. The hyper-parameters are presented in Table C.1. To estimate the gradient of O-information (Figure 5.4) the model width is double the one presented in Table C.1 to account for the additional necessary terms to learn. Concerning the experiments in Figure 5.5 , we use the same architecture used for the canonical examples and follow the same procedure to choose the model capacity(see Table C.1 for the hyper-parameters details).

[1] https://github.com/MustaphaBounoua/soi
[2] https://github.com/CW-Huang/sdeflow-light

Table C.1: SΩI network training details. Dim of the task correspond the sum of the dimensions of all variables of the system. For the neural data application we report the number of training iterations (.,.) corresponding the "change" case and "No change" case. The number of iteration used for the "No change" is higher since the dataset contains more "no change" flashes compared to "change" flashes.

	Width	Time embed	Batch size	Lr	Iterations	Number of params
$(Dim \leq 50)$	128	128	256	1e-2	195k	320k
$(Dim \leq 100)$	192	192	256	1e-2	195k	747k
$(Dim \geq 100)$	256	256	256	1e-2	195k	1003k
Neural application						
$(Dim \leq 30)$	128	128	256	1e-2	(100k,160k)	320k
$(Dim \leq 75)$	192	192	256	1e-2	(100k,160k)	737k
$(Dim \leq 150)$	256	256	256	1e-2	(100k,160k)	1300k
$(Dim \geq 150)$	384	384	256	1e-2	(100k,160k)	3000k

C.3.3 Baselines

(Bai et al., 2023) decomposes TC into $N-1$ MI terms which are estimated using pairwise neural MI estimator. Similarly by leveraging Eq. (C.12) DTC can also be retrieved by estimating $N-1$ additional MI terms.

$$\mathcal{T}(\mathbf{X}) = \sum_{i=1}^{N-1} \mathcal{I}(\mathbf{X}^i; X^{>i}) \tag{C.10}$$

$$\mathcal{D}(\mathbf{X}) = \mathcal{S}(\mathbf{X}) - \mathcal{T}(\mathbf{X}) = \sum_{i=1}^{N} \mathcal{I}(\mathbf{X}^i; \mathbf{X}^{\setminus i}) - \mathcal{T}(\mathbf{X}) \tag{C.11}$$

$$\mathcal{D}(\mathbf{X}) = \sum_{i=2}^{N} \mathcal{I}(\mathbf{X}^i; \mathbf{X}^{\setminus i}) - \sum_{i=2}^{N-1} \mathcal{I}(\mathbf{X}^i; \mathbf{X}^{>i}) \tag{C.12}$$

Our implementation in based on the the official codebase [3] of (Bai et al., 2023). We use the same architecture and hyper parameters from (Bai et al., 2023): LR $= 1e-3$, Batch size $= 64$. We use an MLP architecture for all the variants of the baseline with 3 linear layers with varying width. For each MI term, the capacity of the neural network is aligned to the input dimension. *Adam optimizer* (Kingma and Ba, 2014) is used for training. We increase the width of the hidden layer to accommodate the data dimension. For the variant of the baseline implemented with MINE, we used smaller layer size as large capacity led to divergence during training. To ensure the best performance, we train each MI estimator model for $80k$ steps for a number of variables $N = 10$ and $40k$ for number of variables $N = 6$. In

[3] https://github.com/Linear95/TC-estimation

the different experiments, we reported the performance results averaged over 5 seeds and dropped the baseline in case of divergence during training.

Limitations of the baseline in computing gradients of O-information It's possible to leverage the decomposition of (Bai et al., 2023), using the compact gradient of O-information formulation Figure C.1.

This will require N MI term for each $\partial_i \Omega(\mathbf{X})$. Consequently to compute all the terms, it's required to train $N * N$ pairwise MI models. While it's possible to leverage some MI terms, if already estimated for the computation of O-information, the overall complexity remains of order $\mathcal{O}(N^2)$.

This naturally raises a scalability problem in training a large number of neural estimator models. Moreover, as the number of MI terms increases, this approach is likely to suffer from cumulative errors observed when estimating O-information.

To compute the gradient of O-information with SΩI, we are instead required to approximate an additional number of denoising score functions. However, our method SΩI amortizes the training costs : we use a unique score network to approximate all the required score functions.

C.3.4 The Visual Behavior Neuropixels

Hereafter we describe the different pre-processing steps applied on the Visual Behavior Neuropixels in § 5.4.2. We follow the same procedure described by (Venkatesh et al., 2023). The selected mice are the ones with both familiar and novel sessions and a minimum number of 20 units in each of the six brain regions: VISp, VISl, VISal ,VISrl, VISam and VISpm. Only the units of good quality are kept. The selection criteria was based on an SNR at least 1, and with fewer than 1 inter-spike interval violations. The non-change flashes correspond to the ones where the image does not change and happen between 4 and 10 flashes after the trial start. Trials corresponding to a change are naturally the ones when the image has changed. Only flashes that occurred while the animal was engaged (based on the reward information) is kept, while the ones corresponding to an omission, or after an omission, and flashes during which the animal licked, were all removed.

The trials were aligned to the start of each stimulus flash, and the 250ms recordings were divided into 5 bins of 50 ms duration averaged over the units of the same region. We use different step sizes to count the spikes which resulted in different dimensional representation but resulted in the same intuition (See Figure C.17,Figure C.16 and Figure C.15). Please note

that unlike (Venkatesh et al., 2023), we don't use PCA to reduce the dimension of the data, and count the number of spikes per unit by averaging the activity over the units of the same region indexed by time.

C.4 A transformer based SΩI

Throughout our experimental campaign as referenced in § 5.4, we employed an MLP structure enhanced with skip connections. While this setup reliably estimated O-information, it produced perfectible gradient of O-information estimation. We address this shortcoming by integrating a more robust architecture capable of scaling with an increased number of denoising score functions. Our approach is based on the latest developments in denoising score matching, incorporating a transformer-based model.

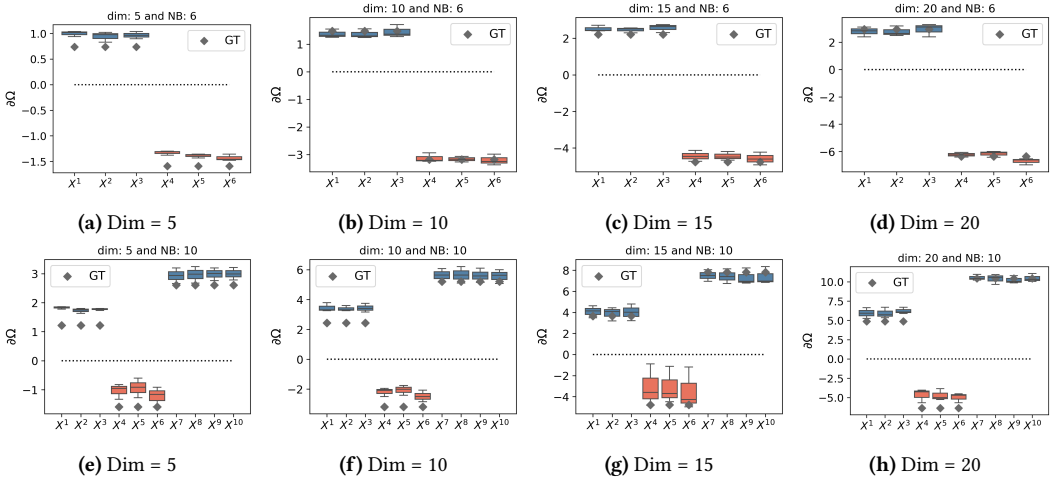
Our method is simple: we adopt the architecture from (Peebles and Xie, 2023) to learn the denoising score functions, treating each modality as a distinct token, while substituting any non-marginal modality with a NULL token (a token with zero value). A transformer block is employed to learn the conditional signal, which is subsequently merged with the temporal signal. This conditioning employs the adaLN-Zero configuration. Our model consists of 4 Blocks, each with 6 attention heads, and the width of the transformer's linear layers is scaled according to the dimension size of the benchmark.The training follows a randomized approach akin to that detailed in Appendix C.2 eliminating the need for a multi-time vector. To compute gradient of O-information, we utilize the formulation presented in Eq. (C.5).

The results presented in Figure C.1 demonstrate the ability of SΩI to accurately estimate the gradients of O-information, provided that the denoising network has sufficient capacity to approximate all the denoising score functions.

C.5 Beyond Normal Benchmarks

In this section, we evaluate SΩI and alternatives across more challenging distributions. To construct such settings we apply MI-invariant transformations to the benchmarks established in Section § 5.4. Since TC and DTC can be written in terms of MI terms, the in-variance of O-information to MI invariant transformations is self-evident.

Half-cube $x \to x\sqrt{|x|}$ is recognized as an MI invariant transformation, which serves to lengthen the tail of the distribution. Addressing the long tail distribution poses a significant

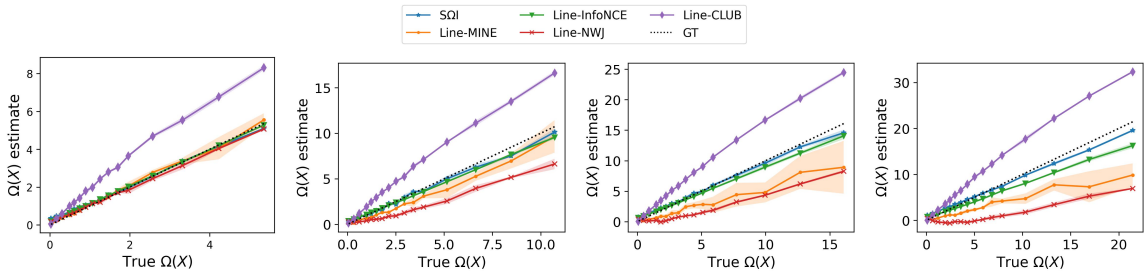


(a) Dim = 5 **(b)** Dim = 10 **(c)** Dim = 15 **(d)** Dim = 20

(e) Dim = 5 **(f)** Dim = 10 **(g)** Dim = 15 **(h)** Dim = 20

Figure C.1: Gradient of O-information using a **transformer based architecture** for the mixed benchmark, for a system of 6 variables, and a system of 10 variables, and different dimension of variables.

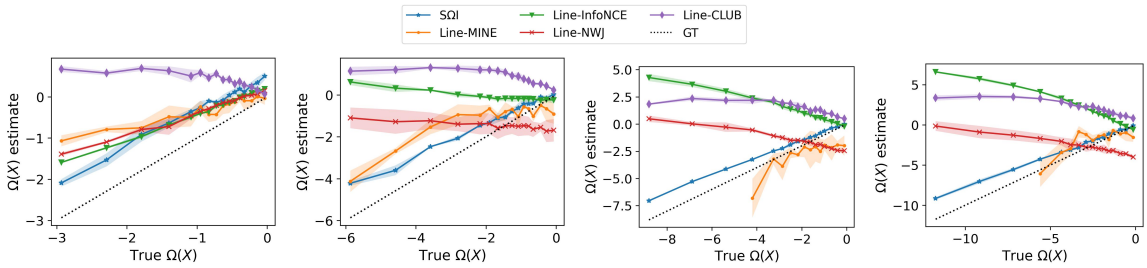
challenge for neural MI estimators, as highlighted in recent studies by(Franzese, Bounoua, and Michiardi, 2024; Czyż et al., 2023). InFigure C.2,Figure C.3 and Figure C.4, we showcase the performance outcomes of SΩI and other baselines on half-Cube transformed benchmarks that exhibit similar interactions as detailed in § 5.4. Our approach stands out by delivering superior performance. Notably, the synergistic transformed benchmark emerges as the most demanding scenario: competitors suffer particularly with high-dimensional variables, while SΩI shows bias, especially in cases of high synergistic interactions, indicated by very low O-information values.



(a) Dim=5 **(b)** Dim=10 **(c)** Dim=15 **(d)** Dim=20

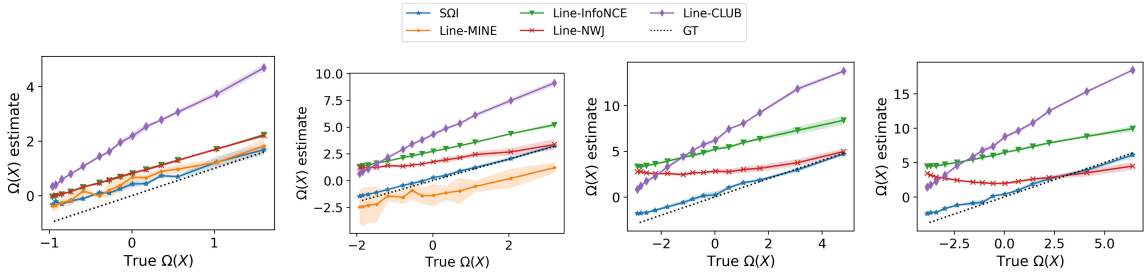
Figure C.2: Redundant system with 10 variables, organized into subsets of sizes $\{3, 3, 4\}$ and increasing interaction strength. A **half-cube** transformation is applied on-top of the multi-normal distribution

CDF The second transformation we consider is the application of a normal cumulative distribution function (CDF), which uniformizes the distribution margins (See (Czyż et al., 2023)). InFigure C.5,Figure C.6 and Figure C.7, we present the performance results of SΩI and alternatives on CDF-transformed benchmarks with a similar configuration used in § 5.4. Our method outperforms competitors, especially for high-dimensional variables. On



(a) Dim=5 **(b)** Dim=10 **(c)** Dim=15 **(d)** Dim=20

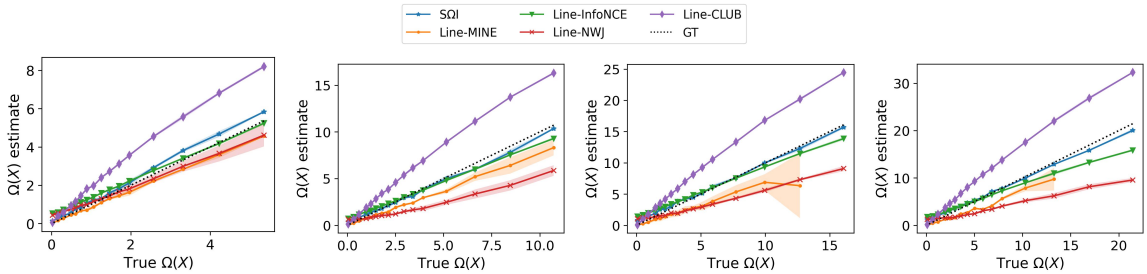
Figure C.3: Synergistic system with 10 variables, organized into subsets of sizes $\{3, 3, 4\}$ and increasing interaction strength. A **half-cube** transformation is applied on-top of the multi-normal distribution.



(a) Dim=5 **(b)** Dim=10 **(c)** Dim=15 **(d)** Dim=20

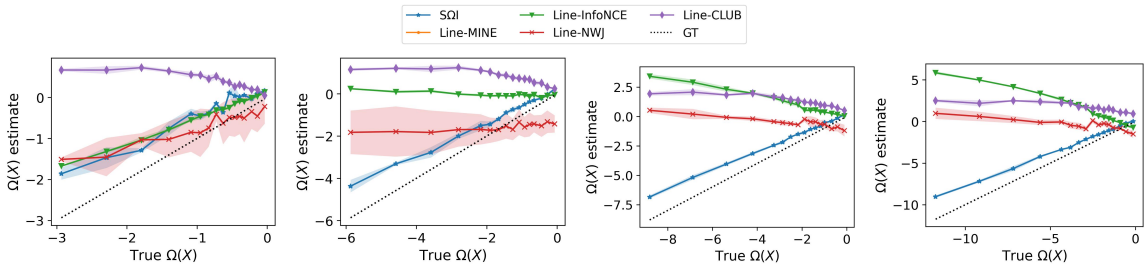
Figure C.4: Mixed-interaction system with 10 variables, organized into 2 redundancy-dominant subsets of size $\{3, 4\}$ variables and one synergy-dominant subset with 3 variables. O-information is modulated by fixing the synergy inter-dependency and increasing the redundancy. A **half-cube** transformation is applied on-top of the multivariate-normal distribution.

the challenging synergistic benchmark, SΩI shows perfectible performance for very low O-information, while competitors fail completely in this setting.



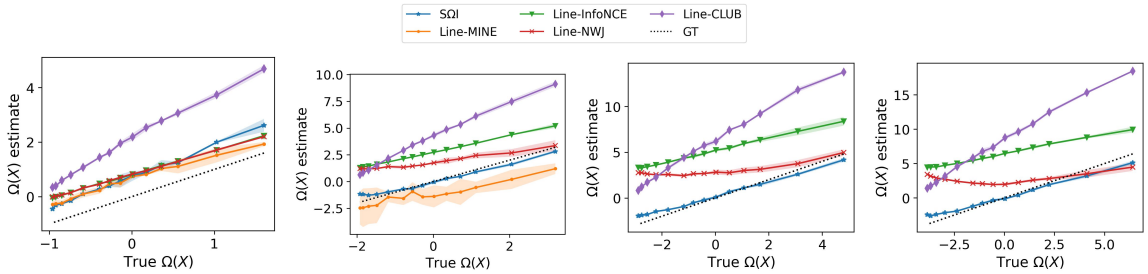
(a) Dim=5 **(b)** Dim=10 **(c)** Dim=15 **(d)** Dim=20

Figure C.5: Redundant system with 10 variables, organized into subsets of sizes $\{3, 3, 4\}$ and increasing interaction strength. A **CDF** transformation is applied on-top of the multi-normal distribution



(a) Dim=5 **(b)** Dim=10 **(c)** Dim=15 **(d)** Dim=20

Figure C.6: Synergistic system with 10 variables, organized into subsets of sizes $\{3, 3, 4\}$ and increasing interaction strength. A **CDF** transformation is applied on-top of the multi-normal distribution



(a) Dim=5 **(b)** Dim=10 **(c)** Dim=15 **(d)** Dim=20

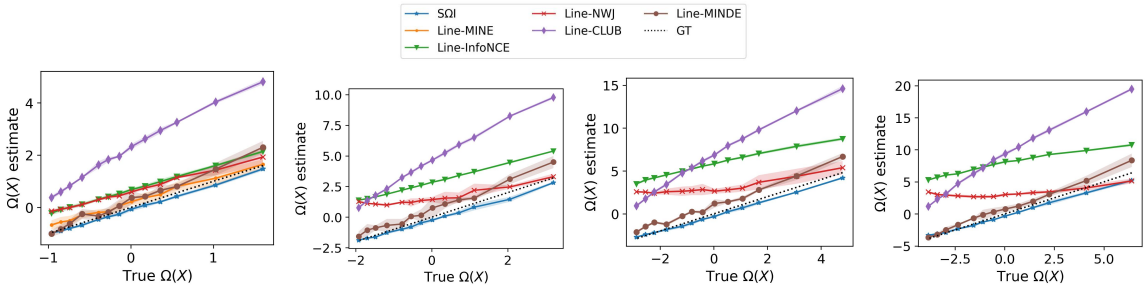
Figure C.7: Mixed-interaction system with 10 variables, organized into 2 redundancy-dominant subsets of size $\{3, 4\}$ variables and one synergy-dominant subset with 3 variables. O-information is modulated by fixing the synergy inter-dependency and increasing the redundancy. A **CDF** transformation is applied on-top of the multivariate-normal distribution.

C.6 Additional results

C.6.1 Additional baseline

(Franzese, Bounoua, and Michiardi, 2024) have shown that the KL divergence between two distributions can be computed using the denoising score function enabling the proposition of an MI estimator. In Figure C.8, we present results on the mixed benchmark (redundancy and synergy) extended with the new baseline called Line-MINDE, that computes O-information using the MI estimator from (Franzese, Bounoua, and Michiardi, 2024). Note that this approach requires learning a set of independent score models, one for each MI term: this increases the total number of parameters to learn, resulting in a more computationally heavy training process compared to our proposed method. In these new experiments, we follow the authors hyper-parameters and score network architecture. We observe that while Line-MINDE outperforms other pairwise MI based estimators, SΩI stands out with the best performance. Our findings indicate that the superiority of SΩI is due to efficiency of score based models in estimating information theoretic measures, which explains the superiority of SΩI and Line-MINDE against other neural estimators. Secondly, the direct estimation of

TC and DTC and the amortized training using a unique network is more efficient which explains why SΩI outperforms Line-MINDE.



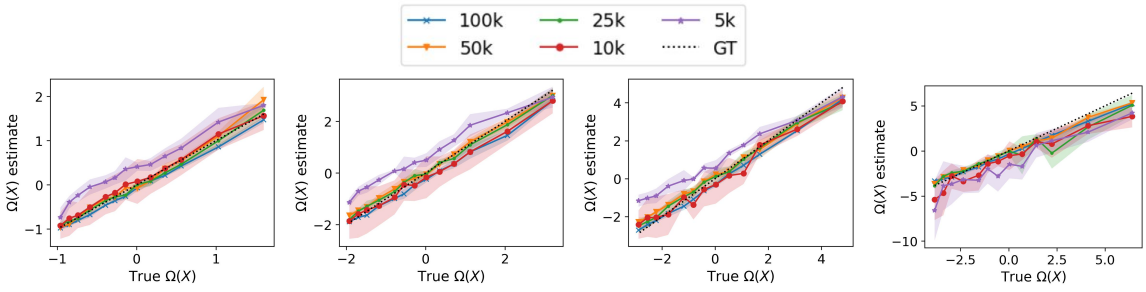
(a) Dim=5 **(b)** Dim=10 **(c)** Dim=15 **(d)** Dim=20

Figure C.8: Additional Line-MINDE (Franzese, Bounoua, and Michiardi, 2024) baseline. Mixed-interaction system with 10 variables, organized into a redundancy-dominant subsets of size $3, 4$ variables and one synergy-dominant subset with 3 variables. O-information is modulated by fixing the synergy inter-dependency and increasing the redundancy.

C.6.2 Ablation study

Data size

In Figure C.9, we present a training size ablation study on the mixed benchmark. The considered number of training samples are of 5k,10k,25k,50k,100k samples. We fix the testset to 10k samples, except when the training size is 5k, for which we use 5k test samples. We observe that for data size superior to 10k, SΩI obtains very good estimates in terms of bias and variance; when the training size has 10k samples, SΩI estimates have increased variance; when we use only 5k training samples, SΩI have increased bias. These results are to be expected, since neural estimators, in general, require sufficient training data to shine.

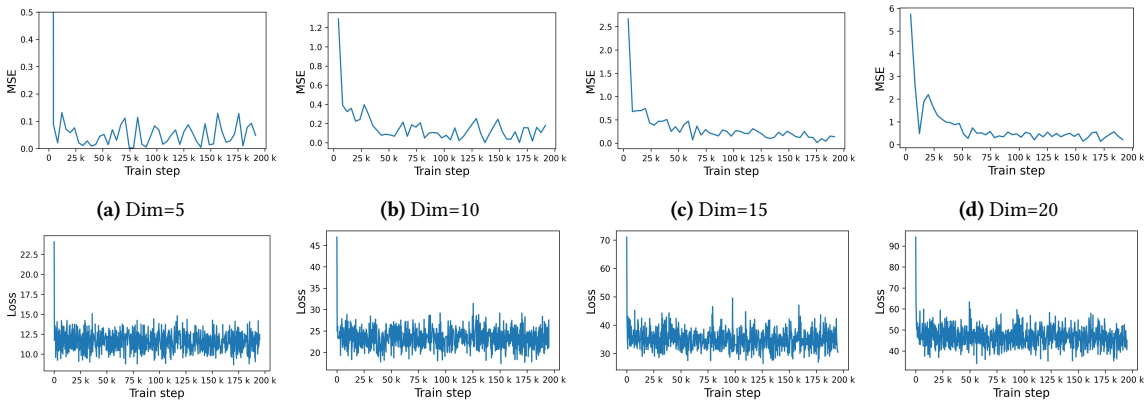


(a) Dim=5 **(b)** Dim=10 **(c)** Dim=15 **(d)** Dim=20

Figure C.9: SΩI training size ablation study : 100k,50k,25k,10k,5k. We use a test size of 10k for all the settings except when the train set size is equal to 5k where we use test size of similar size. The considered benchmark is a mixed-interaction system with 10 variables, organized into a redundancy-dominant subsets of size $3, 4$ variables and one synergy-dominant subset with 3 variables. O-information is modulated by fixing the synergy inter-dependency and increasing the redundancy.

Number of training iterations

In Figure C.10, we present the training curves contrasted with MI estimate mean squared error. Clearly, the number of iterations required to achieve satisfactory results depends on the dataset complexity.



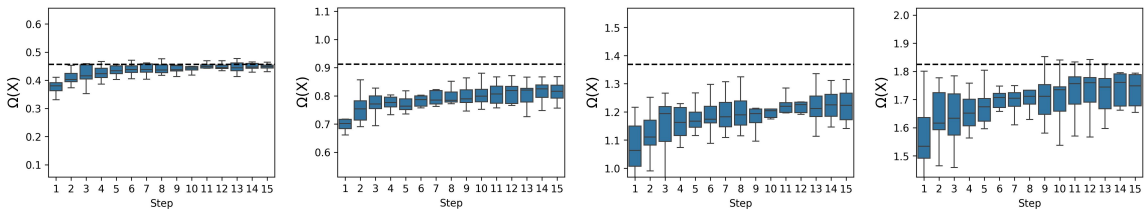
(a) Dim=5 **(b)** Dim=10 **(c)** Dim=15 **(d)** Dim=20

(e) Dim=5 **(f)** Dim=10 **(g)** Dim=15 **(h)** Dim=20

Figure C.10: Training Loss curve Vs Estimation of O-information MSE. Mixed-interaction system with 10 variables, organized into a redundancy-dominant subsets of size $3, 4$ variables and one synergy-dominant subset with 3 variables. For different benchmark dimensions, we report: **Top:** O-information estimation mean square error as a function of the training iterations. **Bottom:** Training loss curve.

Monte Carlo integration steps

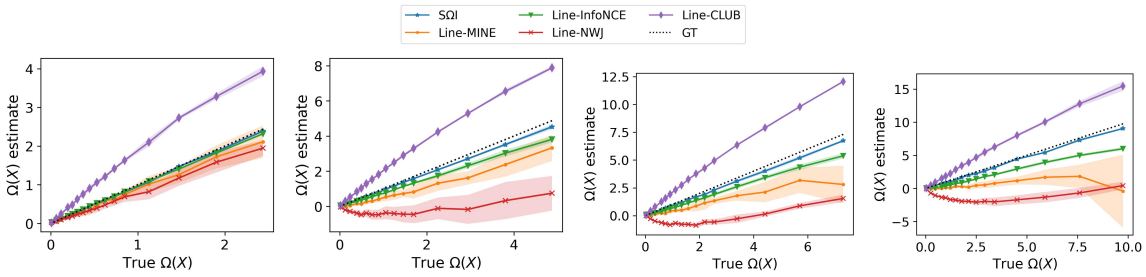
In Figure C.11, we present an ablation on the number of Monte Carlo steps, for the case of a mixed (redundancy and synergy) benchmark with $N = 10$ random variables. We notice that an increased number of steps improves the estimation variance and bias. Naturally, this depends on the data dimension and complexity.



(a) Dim=5 **(b)** Dim=10 **(c)** Dim=15 **(d)** Dim=20

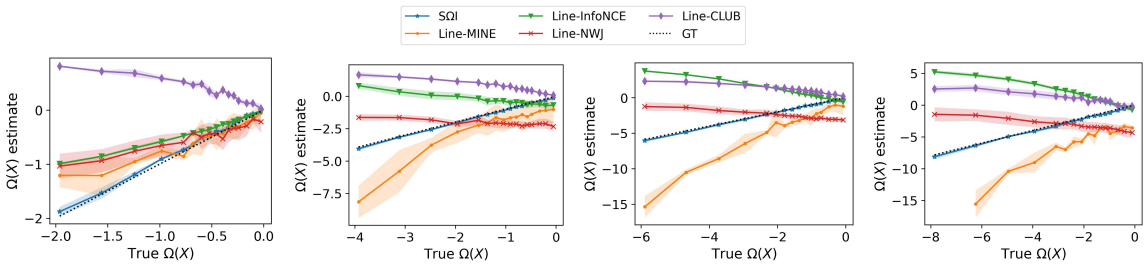
Figure C.11: Estimation of O-information as a function of Monte Carlo Averaging steps run over 10 seeds. Mixed-interaction system with 10 variables, organized into a redundancy-dominant subsets of size $3, 4$ variables and one synergy-dominant subset with 3 variables. Dashed line represents ground truth O-information.

C.6.3 Additional synthetic experiments



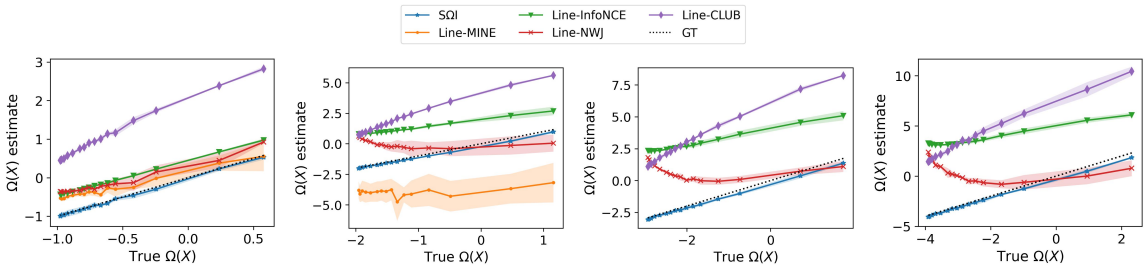
(a) Dim=5 **(b)** Dim=10 **(c)** Dim=15 **(d)** Dim=20

Figure C.12: Redundant system with 6 variables, organized into subsets of sizes $\{3, 3\}$ and increasing interaction strength.



(a) Dim=5 **(b)** Dim=10 **(c)** Dim=15 **(d)** Dim=20

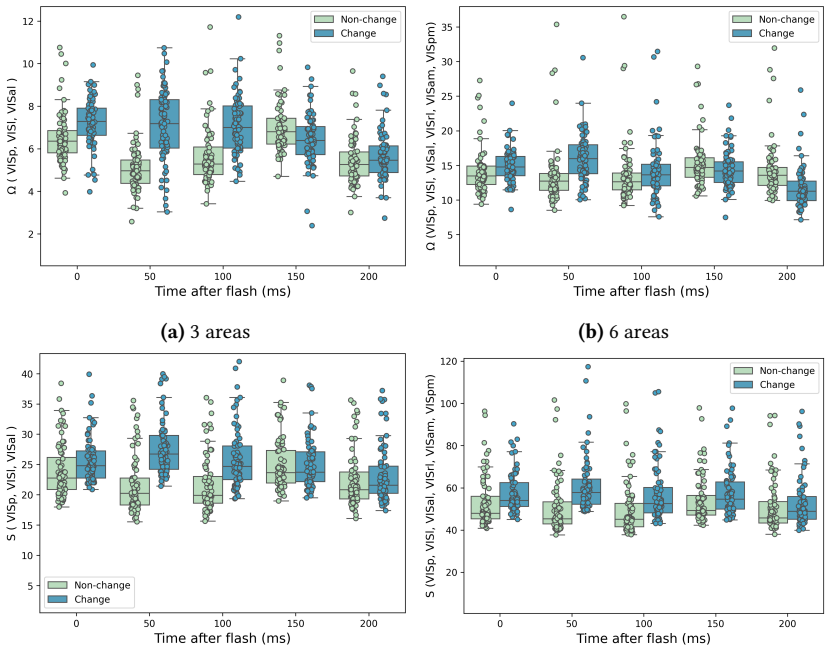
Figure C.13: Synergistic system with 6 variables, organized into subsets of sizes $\{3, 3\}$ and increasing interaction strength.



(a) Dim=5 **(b)** Dim=10 **(c)** Dim=15 **(d)** Dim=20

Figure C.14: Mixed-interaction system with 6 variables, organized into a redundancy-dominant subsets of size 3 variables and one synergy-dominant subset with 3 variables. O-information is modulated by fixing the synergy inter-dependency and increasing the redundancy.

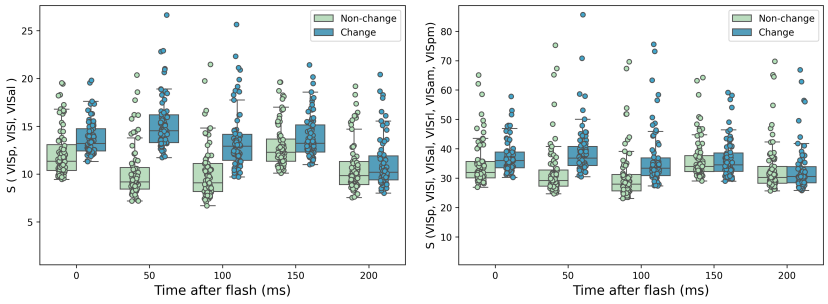
C.6.4 The neural application additional experiments



(a) 3 areas **(b)** 6 areas

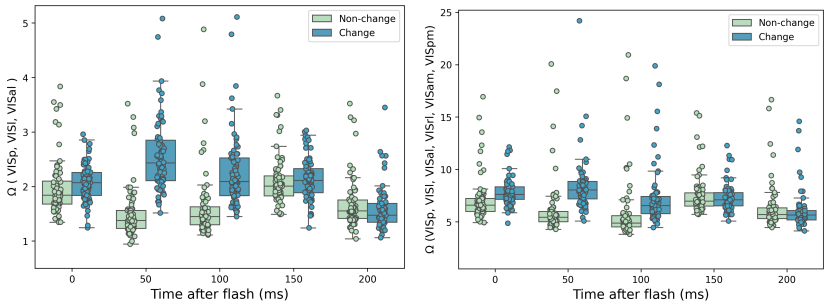
(c) 3 areas **(d)** 6 areas

Figure C.15: O-information and S-information estimate in the visual cortex region activity after two types of stimulus flash across 72 trial sessions. **Left**: Analysis using three brain region areas, **Right**: Extended analysis using six brain region areas. The step size is set to $1ms$ which results in **50** dimensional data for each bin per area.



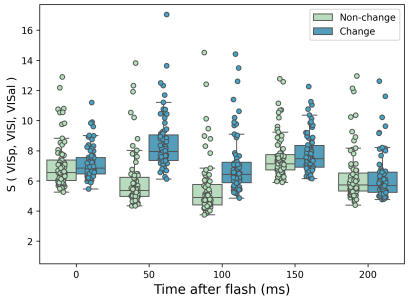
(a) 3 areas **(b)** 6 areas

Figure C.16: S-information estimate in the visual cortex region activity after two types of stimulus flash across 72 trial sessions. **Left**: Analysis using three brain region areas, **Right**: Extended analysis using six brain region areas. The step size is set to $2ms$ which results in **25** dimensional data for each bin per area.

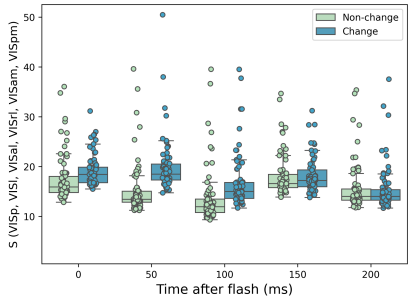


(a) 3 areas

(b) 6 areas



(c) 3 areas



(d) 6 areas

Figure C.17: O-information and S-information estimate in the visual cortex region activity after two types of stimulus flash across 72 trial sessions. **Left**: Analysis using three brain region areas, **Right**: Extended analysis using six brain region areas. The step size is set to $5ms$ which results in **10** dimensional data for each bin per area.

Appendix D

Appendix for Chapter 6

D.1 Implementation Details

D.1.1 Diffusion Models Training with Reinforcement Learning

Our methodology begins by training unconditional diffusion models for both data modalities. We then use a reinforcement learning technique to train two conditional models (initialized from the first step) in a cooperative manner, allowing them to learn from each other to optimize the joint coupling under MEC constraints and objectives. We formulate this second phase as training diffusion models with reinforcement learning and $\mathbb{KL}$-regularization. We follow the training scheme presented by Fan et al. (2023), where samples are generated conditionally using classifier guidance (Ho and Salimans, 2022) with a DDIM sampler (Song, Meng, and Ermon, 2020). The generated trajectories are then used to update the diffusion model, which is framed as a Markov Decision Process (MDP), using a policy gradient RL algorithm.

Reward Estimation In DDMEC, the reward signals are log-likelihood values mutually generated by the two conditional models. Accurately estimating this signal is crucial for steering training towards the optimal MEC solution. To achieve this, we use multiple Monte Carlo steps to estimate Equation 6.9.

Policy Gradient Training We follow the training procedure of Fan et al. (2023), where, at each step, a batch of samples is generated using DDIM (Song, Meng, and Ermon, 2020). These generated trajectories are then used to perform multiple gradient updates. Additionally, we apply importance sampling and ratio clipping (Schulman et al., 2017) to improve training stability.

Classifier-Free Guidance We employ classifier-free guidance (Ho and Salimans, 2022) in all experiments. This technique enables conditional sampling in Line 2. The denoising loss in 4 is optimized to account for the guidance mechanism by randomly dropping 10% of the conditional signal, thereby stabilizing the unconditional model.

D.1.2 Technical Details and Hyperparameters

Single-Cell Alignment We use the SNARE-SEQ dataset available in the official code repository of (Demetci et al., 2022) `https://github.com/rsinghlab/SCOT`,

along with the provided preprocessing steps and evaluation procedure. We find it beneficial to normalize the data by subtracting the mean and standardizing it to unit variance. In this experiment, we employ a simple MLP network as described in Table D.1. The model is initially trained unconditionally for 2000 steps using DDPM with $T = 50$. Subsequently, we train the two conditional models for 350 steps, where each step corresponds to a single iteration of Algorithm 1, involving one policy update and one application of the denoising loss. For importance sampling, we set the clipping hyperparameter to 0.01. We use a batch size of 64, a learning rate of 5×10^{-3}, and the Adam optimizer (Kingma and Ba, 2014).

Layer	Details
Input Embedding	Linear (dim, 128)
Condition Embedding	Linear (input_dim, 128)
Time Embedding	Positional Encoding (128)
Fully Connected 1	Linear (256) + ReLU
Fully Connected 2	Linear (256) + ReLU
Fully Connected 3	Linear (256) + ReLU
Fully Connected 4	Linear (256) + ReLU
Output Layer	Linear (input_dim)

Table D.1: Architecture of the denoising network used in § 6.4.1.

Unpaired Image Translation - Cat→Dog and Wild→Dog Tasks: We utilize the pre-trained model from the official implementation of Choi et al. (2021) (https://github.com/jychoi118/ilvr_adm) to initialize the dog modality conditional model. For the other domains (Cat, Wild): We train a diffusion model from scratch using the same architecture and hyper-parameters as done in the target domain.

To introduce additional conditioning into the pre-trained diffusion model, we follow the work in (Zhang, Rao, and Agrawala, 2023), where the encoder part of the U-Net is duplicated and used as a conditional encoder. The various hyperparameters are summarized in Table D.2. We follow the evaluation protocol described in (Zhao et al., 2022).

General Settings	Dataset
	AFHQ
Batch Size	16
Learning Rate:	$2e-5$
Optimizer	ADAM
Training Steps	2000
Weight Decay	0.0
Diffusion Model	
Noise Scheduler	Linear
Number of Diffusion Timesteps (T)	1000
Sampler	DDIM
Guidance Scale (training)	7.0
Sampling steps	50
Exponential moving average	Yes
Reinforcement Learning	
Reward (Monte Carlo steps)	3
Policy Update Steps	4
Importance Sampling Clipping	$1e-4$
λ_1, λ_2	$1e-3$
Gradient Accumulation	12
Gradient Clipping	1.0

Table D.2: Hyperparameters used for training.

D.2 Additional Results

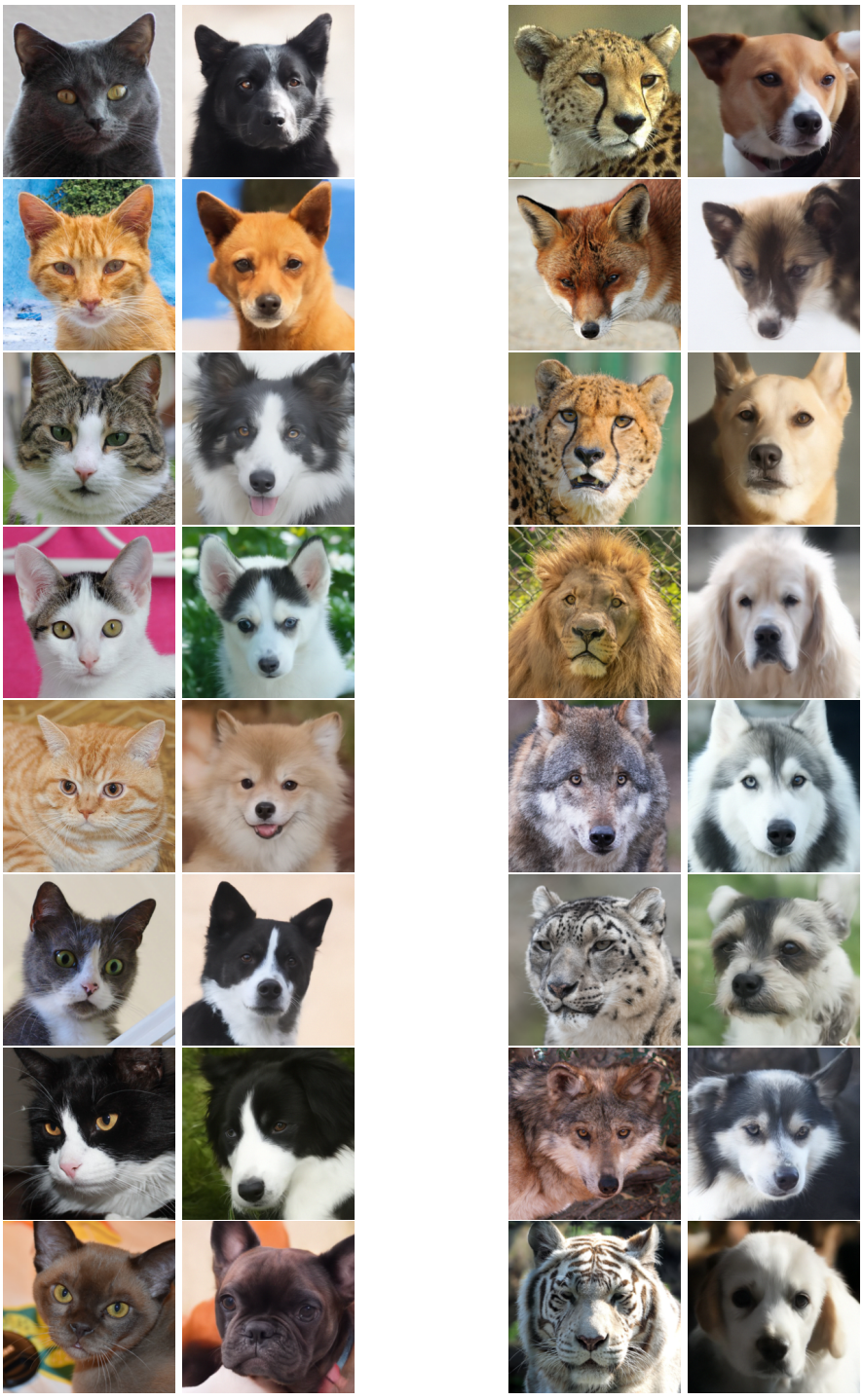


Figure D.1: DDMEC (guidance=7) CAT→DOG (*Left*) and WILD→DOG image (*right*) translation examples. Source domain image is used to generate the target dog image.

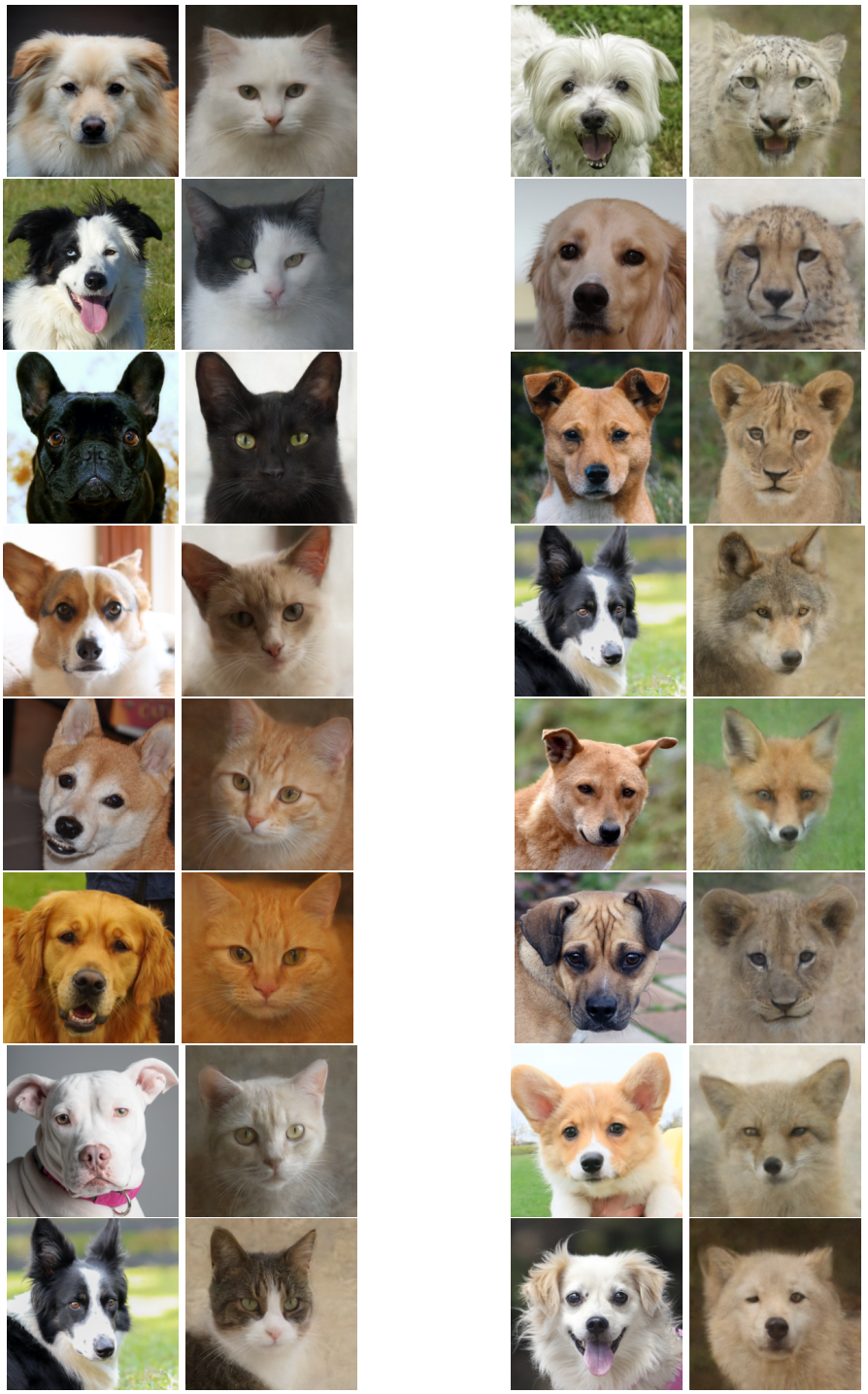


Figure D.2: DDMEC (guidance=7) Dog→Cat (*Left*) and Dog→Wild image (*right*) translation examples. Source domain image is used to generate the target Cat/Wild image.

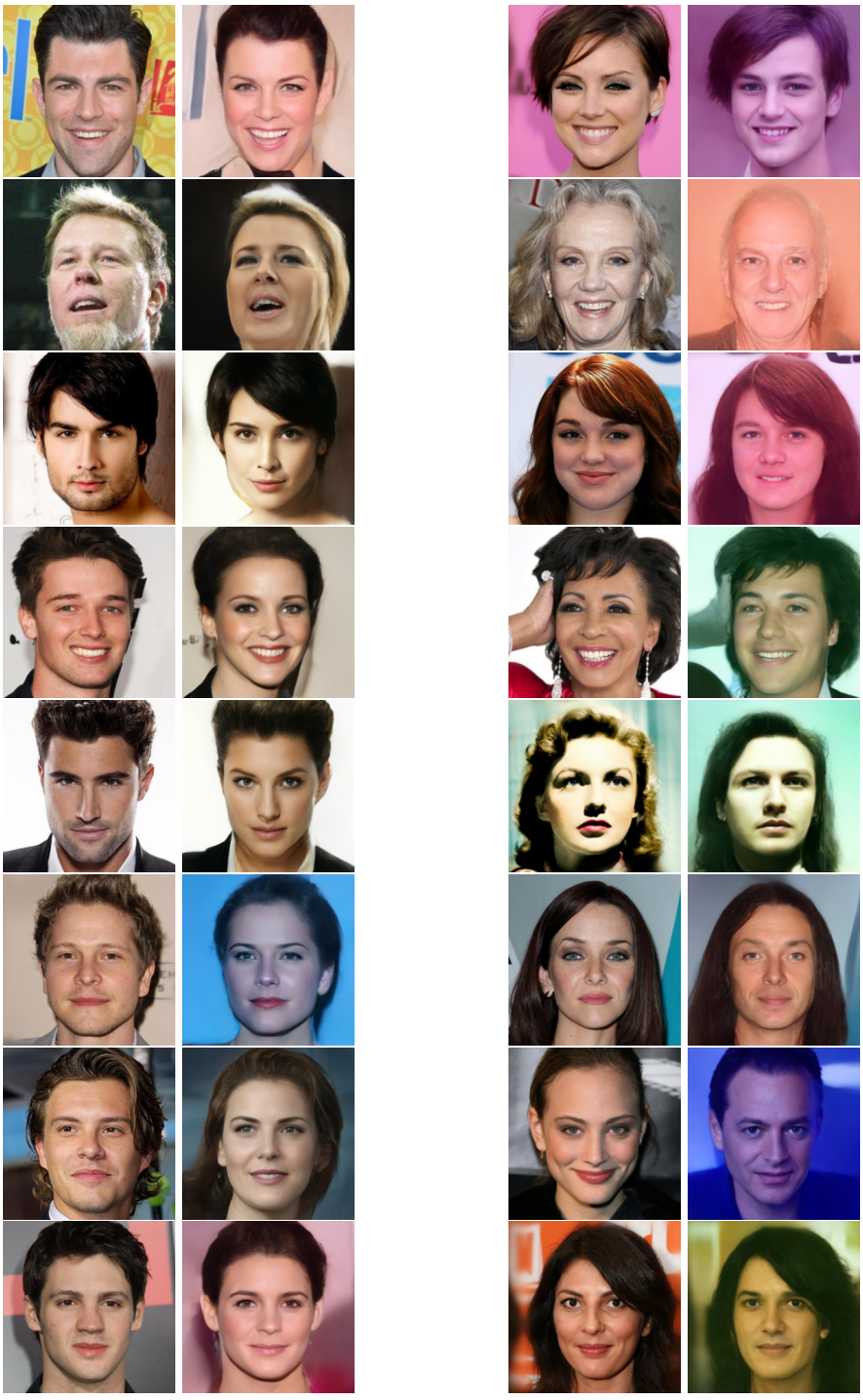


Figure D.3: DDMEC MALE→FEMALE (*Left*) and FEMALE→MALE image (*right*) translation examples. Source domain image is used to generate the target female image.

References

Abu Tami, Mohammad, Huthaifa I. Ashqar, Mohammed Elhenawy, Sebastien Glaser, and Andry Rakotonirainy (2024). "Using Multimodal Large Language Models (MLLMs) for Automated Detection of Traffic Safety-Critical Events". In: *Vehicles* 6.3, pp. 1571–1590. ISSN: 2624-8921.

Achiam, Josh, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. (2023). "Gpt-4 technical report". In: *arXiv preprint arXiv:2303.08774*.

Ackley, David H, Geoffrey E Hinton, and Terrence J Sejnowski (1985). "A learning algorithm for Boltzmann machines". In: *Cognitive science* 9.1, pp. 147–169.

Acosta, Julián N., Guido J. Falcone, Pranav Rajpurkar, and Eric J. Topol (2022). "Multimodal biomedical AI". In: *Nature Medicine* 28, pp. 1773–1784.

Akkaya, Ilge, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. (2019). "Solving rubik's cube with a robot hand". In: *arXiv preprint arXiv:1910.07113*.

Alemi, Alexander, Ben Poole, Ian Fischer, Joshua Dillon, Rif A Saurous, and Kevin Murphy (2018). "Fixing a broken ELBO". In: *International conference on machine learning*. PMLR, pp. 159–168.

Alemi, Alexander A and Ian Fischer (2018). "GILBO: One metric to measure them all". In: *Advances in Neural Information Processing Systems* 31.

Alemi, Alexander A, Ian Fischer, Joshua V Dillon, and Kevin Murphy (2016). "Deep Variational Information Bottleneck". In: *International Conference on Learning Representations*.

Allen-Institute (2022). "Visual behavior neuropixels dataset overview". In.

Amodio, Matthew and Smita Krishnaswamy (2018). "MAGAN: Aligning biological manifolds". In: *International conference on machine learning*. PMLR, pp. 215–223.

Anderson, Brian D. O. (1982). "Reverse-time diffusion equation models". In: *Stochastic Processes and their Applications* 12.3, pp. 313–326.

Anderson, Peter, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang (2018). "Bottom-up and top-down attention for image captioning and visual question answering". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6077–6086.

Antelmi, Luigi, Nicholas Ayache, Philippe Robert, and Marco Lorenzi (June 2019). "Sparse Multi-Channel Variational Autoencoder for the Joint Analysis of Heterogeneous Data".

In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, pp. 302–311.

Arjovsky, Martin, Soumith Chintala, and Léon Bottou (2017). "Wasserstein generative adversarial networks". In: *International conference on machine learning*. PMLR, pp. 214–223.

Asperti, Andrea and Matteo Trentin (2020). "Balancing Reconstruction Error and Kullback-Leibler Divergence in Variational Autoencoders". In: *IEEE Access* 8, pp. 199440–199448.

Ay, Nihat, Daniel Polani, and Nathaniel Virgo (2019). "Information Decomposition based on Cooperative Game Theory". In: *ArXiv* abs/1910.05979.

Azizi, Shekoofeh, Simon Kornblith, Chitwan Saharia, Mohammad Norouzi, and David J. Fleet (2023). *Synthetic Data from Diffusion Models Improves ImageNet Classification*. arXiv: 2304.08466 [cs.CV].

Bai, Ke, Pengyu Cheng, Weituo Hao, Ricardo Henao, and Larry Carin (2023). "Estimating Total Correlation with Mutual Information Estimators". In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 2147–2164.

Balaji, Yogesh, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, et al. (2022). "ediffi: Text-to-image diffusion models with an ensemble of expert denoisers". In: *arXiv preprint arXiv:2211.01324*.

Baltrušaitis, Tadas, Chaitanya Ahuja, and Louis-Philippe Morency (2018). "Multimodal machine learning: A survey and taxonomy". In: *IEEE transactions on pattern analysis and machine intelligence* 41.2, pp. 423–443.

Bao, Fan, Shen Nie, Kaiwen Xue, Chongxuan Li, Shi Pu, Yaole Wang, Gang Yue, Yue Cao, Hang Su, and Jun Zhu (2023). *One Transformer Fits All Distributions in Multi-Modal Diffusion at Scale*. arXiv: 2303.06555 [cs.LG].

Barber, David and Felix Agakov (2004). "The im algorithm: a variational approach to information maximization". In: *Advances in neural information processing systems* 16.320, p. 201.

Barrett, Adam B. (2014). "An exploration of synergistic and redundant information sharing in static and dynamical Gaussian systems". In: *CoRR* abs/1411.2832. arXiv: 1411.2832.

Belghazi, Mohamed Ishmael, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm (2018). "Mutual Information Neural Estimation". In: *Proceedings of the 35th International Conference on Machine Learning*.

Bell, Anthony J and Terrence J Sejnowski (1995). "An information-maximization approach to blind separation and blind deconvolution". In: *Neural computation* 7.6, pp. 1129–1159.

Benaim, Sagie and Lior Wolf (2017). "One-sided unsupervised domain mapping". In: *Advances in neural information processing systems* 30.

Benes, Viktor and Josef Stepán (2012). *Distributions with Given Marginals and Moment Problems*. Springer Science & Business Media.

Bengio, Yoshua and Samy Bengio (1999). "Modeling high-dimensional discrete data with multi-layer neural networks". In: *Advances in neural information processing systems* 12.

Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Black, Kevin, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine (2024). "Training Diffusion Models with Reinforcement Learning". In: *The Twelfth International Conference on Learning Representations*.

Blattmann, Andreas, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis (2023). *Align your Latents: High-Resolution Video Synthesis with Latent Diffusion Models*. arXiv: 2304.08818 [cs.CV].

Bogdoll, Daniel, Yitian Yang, and J Marius Zöllner (2023). "Muvo: A multimodal generative world model for autonomous driving with geometric representations". In: *arXiv preprint arXiv:2311.11762*.

Bounoua, Mustapha, Christophe Beaugeant, Giulio Franzese, and Pietro Michiardi (2024). "Enhancing Sensor Robustness in Automotive Systems: A Multimodal Generative Approach". In: *SIA-Vision 2024*. Paris, France.

Bounoua, Mustapha, Giulio Franzese, and Pietro Michiardi (2023). "Masked Multi-time Diffusion for Multi-modal Generative Modeling". In: *Neural Information Processing Systems (NeurIPS) 2023 Workshop on Diffusion Models*. New Orleans, US.

Bounoua, Mustapha, Giulio Franzese, and Pietro Michiardi (2024a). "Multi-modal latent diffusion". In: *Entropy* 26.4, p. 320.

Bounoua, Mustapha, Giulio Franzese, and Pietro Michiardi (2024b). "SΩI: Score-based O-INFORMATION Estimation". In: *ICML 2024, 41st International Conference on Machine Learning*. IEEE. Vienna, Austria.

Bounoua, Mustapha, Giulio Franzese, and Pietro Michiardi (2025). "Learning to Match Unpaired Data with Minimum Entropy Coupling". In: *ICML 2025, 42nd International Conference on Machine Learning*. Vancouver, Canada.

Brekelmans, Rob, Sicong Huang, Marzyeh Ghassemi, Greg Ver Steeg, Roger Baker Grosse, and Alireza Makhzani (2022). "Improving Mutual Information Estimation with Annealed and Energy-Based Bounds". In: *International Conference on Learning Representations*.

Brown, Tom, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. (2020). "Language models are few-shot learners". In: *Advances in neural information processing systems* 33, pp. 1877–1901.

Caesar, Holger, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom (2020). "nuscenes: A multimodal dataset for autonomous driving". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11621–11631.

Callen, Herbert B (1991). *Thermodynamics and an Introduction to Thermostatistics*. John wiley & sons.

Cao, Kai, Xiangqi Bai, Yiguang Hong, and Lin Wan (2020). "Unsupervised topological alignment for single-cell multi-omics integration". In: *Bioinformatics* 36.Supplement_1, pp. i48–i56.

Chang, Huiwen, Han Zhang, Jarred Barber, AJ Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy, William T. Freeman, Michael Rubinstein, Yuanzhen Li, and Dilip

Krishnan (2023). *Muse: Text-To-Image Generation via Masked Generative Transformers*. arXiv: 2301.00704 [cs.CV].

Chen, Mengxi, Linyu Xing, Yu Wang, and Ya Zhang (2023). "Enhanced Multimodal Representation Learning with Cross-Modal KD". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11766–11775.

Chen, Shan, Brook B. Lake, and Kun Zhang (2019). "High-throughput sequencing of the transcriptome and chromatin accessibility in the same cell". In: *Nature Biotechnology* 37.12, pp. 1452–1457.

Chen, Wei, Xixuan Hao, Yuankai Wu, and Yuxuan Liang (2024). "Terra: A Multimodal Spatio-Temporal Dataset Spanning the Earth". In: *Advances in Neural Information Processing Systems*.

Chen, Xi, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel (2016). "Infogan: Interpretable representation learning by information maximizing generative adversarial nets". In: *Advances in neural information processing systems* 29.

Cheng, Pengyu, Weituo Hao, Shuyang Dai, Jiachang Liu, Zhe Gan, and Lawrence Carin (2020). "Club: A contrastive log-ratio upper bound of mutual information". In: *International conference on machine learning*. PMLR, pp. 1779–1788.

Chiarion, Giovanni, Laura Sparacino, Yuri Antonacci, Luca Faes, and Luca Mesin (2023). "Connectivity Analysis in EEG Data: A Tutorial Review of the State of the Art and Emerging Trends". In: *Bioengineering* 10.3. ISSN: 2306-5354.

Choi, Jooyoung, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon (2021). "Ilvr: Conditioning method for denoising diffusion probabilistic models". In: *arXiv preprint arXiv:2108.02938*.

Choi, Yunjey, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha (June 2020). "StarGAN v2: Diverse Image Synthesis for Multiple Domains". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chuang, Ching-Yao, Stefanie Jegelka, and David Alvarez-Melis (2023). "InfoOT: Information Maximizing Optimal Transport". In: *Proceedings of the 40th International Conference on Machine Learning*.

Cicalese, Ferdinando, Luisa Gargano, and Ugo Vaccaro (2016). "Approximating probability distributions with short vectors, via information theoretic distance measures". In: *2016 IEEE International Symposium on Information Theory (ISIT)*. IEEE, pp. 1138–1142.

Cicalese, Ferdinando, Luisa Gargano, and Ugo Vaccaro (2017). "How to find a joint probability distribution of minimum entropy (almost) given the marginals". In: *2017 IEEE International Symposium on Information Theory (ISIT)*. IEEE, pp. 2173–2177.

Cicalese, Ferdinando, Luisa Gargano, and Ugo Vaccaro (2019). "Minimum-Entropy Couplings and Their Applications". In: *IEEE Transactions on Information Theory* 65.6, pp. 3436–3451.

Clark, Kevin, Paul Vicol, Kevin Swersky, and David J Fleet (2023). "Directly fine-tuning diffusion models on differentiable rewards". In: *arXiv preprint arXiv:2309.17400*.

Collet, Jean-François and Florent Malrieu (2008). "Logarithmic Sobolev inequalities for inhomogeneous Markov semigroups". In: *ESAIM: Probability and Statistics* 12, pp. 492–504.

Compton, Spencer (2022). "A tighter approximation guarantee for greedy minimum entropy coupling". In: *2022 IEEE International Symposium on Information Theory (ISIT)*. IEEE, pp. 168–173.

Cover, Thomas M, Joy A Thomas, et al. (1991). "Entropy, relative entropy and mutual information". In: *Elements of information theory* 2.1, pp. 12–13.

Crooks, Gavin E (1999). "Entropy production fluctuation theorem and the nonequilibrium work relation for free energy differences". In: *Physical Review E* 60.3, p. 2721.

Csiszár, Imre (1967). "On information-type measure of difference of probability distributions and indirect observations". In: *Studia Sci. Math. Hungar.* 2, pp. 299–318.

Czyż, Paweł, Frederic Grabowski, Julia E Vogt, Niko Beerenwinkel, and Alexander Marx (2023). "Beyond Normal: On the Evaluation of Mutual Information Estimators". In: *Advances in Neural Information Processing Systems*.

Da Silva–Filarder, Matthieu, Andrea Ancora, Maurizio Filippone, and Pietro Michiardi (2021). "Multimodal Variational Autoencoders for Sensor Fusion and Cross Generation". In: *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 1069–1076.

Dai, Bin and David Wipf (2019). "Diagnosing and enhancing VAE models". In: *arXiv preprint arXiv:1903.05789*.

Daunhawer, Imant, Thomas M. Sutter, Kieran Chin-Cheong, Emanuele Palumbo, and Julia E Vogt (2022). "On the Limitations of Multimodal VAEs". In: *International Conference on Learning Representations*.

Demetci, Pinar, Ryan Santorella, Björn Sandstede, William Stafford Noble, and Ritambhara Singh (2022). "SCOT: Single-Cell Multi-Omics Alignment with Optimal Transport". In: *Journal of Computational Biology* 29.1, pp. 3–18.

Den Hollander, Frank (2012). *Probability Theory: The Coupling Method*. Lecture notes available online.

Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei (2009). "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee, pp. 248–255.

Dewan, Shaurya Rajat, Rushikesh Zawar, Prakanshul Saxena, Yingshan Chang, Andrew Luo, and Yonatan Bisk (2024). "Diffusion PID: Interpreting Diffusion via Partial Information Decomposition". In: *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Dhariwal, Prafulla and Alexander Nichol (2021). "Diffusion Models Beat GANs on Image Synthesis". In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan. Vol. 34. Curran Associates, Inc., pp. 8780–8794.

Di Giacomo, G, G Franzese, T Cerquitelli, CF Chiasserini, P Michiardi, et al. (2024). "DIMVIDA: Diffusion-based Multi-View Data Augmentation". In: *IEEE CAMAD 2024*. IEEE.

Di Giacomo, Giuseppe, Giulio Franzese, Tania Cerquitelli, Carla-Fabiana Chiasserini, and Pietro Michiardi (2023). "Multi-View Latent Diffusion". In: *2023 IEEE International Conference on Big Data (BigData)*. IEEE, pp. 6152–6154.

Dieng, Adji B, Yoon Kim, Alexander M Rush, and David M Blei (2019). "Avoiding latent variable collapse with generative skip models". In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 2397–2405.

Dinh, Laurent, David Krueger, and Yoshua Bengio (2014). "Nice: Non-linear independent components estimation". In: *arXiv preprint arXiv:1410.8516*.

Dinh, Laurent, Jascha Sohl-Dickstein, and Samy Bengio (2016). "Density estimation using real nvp". In: *arXiv preprint arXiv:1605.08803*.

Dockhorn, Tim, Arash Vahdat, and Karsten Kreis (2022). "Score-Based Generative Modeling with Critically-Damped Langevin Diffusion". In: *International Conference on Learning Representations*.

Dong, Runpei, Chunrui Han, Yuang Peng, Zekun Qi, Zheng Ge, Jinrong Yang, Liang Zhao, Jianjian Sun, Hongyu Zhou, Haoran Wei, Xiangwen Kong, Xiangyu Zhang, Kaisheng Ma, and Li Yi (2024). "DreamLLM: Synergistic Multimodal Comprehension and Creation". In: *International Conference on Learning Representations (ICLR)*.

Dosi, Giovanni and Andrea Roventini (2019). "More is different... and complex! the case for agent-based macroeconomics". In: *Journal of Evolutionary Economics* 29, pp. 1–37.

Dupont, Emilien, Hyunjik Kim, S. M. Ali Eslami, Danilo Jimenez Rezende, and Dan Rosenbaum (July 2022). "From data to functa: Your data point is a function and you can treat it like one". In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato. Vol. 162. Proceedings of Machine Learning Research. PMLR, pp. 5694–5725.

Ebrahimi, M Reza, Jun Chen, and Ashish Khisti (2024). "Minimum Entropy Coupling with Bottleneck". In: *arXiv preprint arXiv:2410.21666*.

Ehrlich, David A, Kyle Schick-Poland, Abdullah Makkeh, Felix Lanfermann, Patricia Wollstadt, and Michael Wibral (2023). "Partial Information Decomposition for Continuous Variables based on Shared Exclusions: Analytical Formulation and Estimation". In: *arXiv preprint arXiv:2311.06373*.

Enk, Steven J. van (2023). "Pooling probability distributions and partial information decomposition." In: *Physical review. E* 107 5-1, p. 054133.

Esser, Patrick, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach (2024). "Scaling Rectified Flow Transformers for High-Resolution Image Synthesis". In: *Proceedings of the 41st International Conference on Machine Learning (ICML)*.

Fan, Ying, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee (2023). "DPOK: Reinforcement Learning for Fine-tuning Text-to-Image Diffusion Models". In: *Advances in Neural Information Processing Systems*. Ed. by A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine. Vol. 36. Curran Associates, Inc., pp. 79858–79885.

Federici, Marco, David Ruhe, and Patrick Forré (2023). "On the Effectiveness of Hybrid Mutual Information Estimation". In: *arXiv preprint arXiv:2306.00608*.

Finn, Conor and Joseph T. Lizier (2020). "Generalised Measures of Multivariate Information Content". In: *Entropy* 22.2. ISSN: 1099-4300.

Foresti, Alberto, Giulio Franzese, and Pietro Michiardi (2025). "Info-SEDD: continuous time markov chains as scalable information metrics estimators". In: *ICLR 2025, DeLTa Workshop, Deep Generative Model in Machine Learning:
 Theory, Principle and Efficacy, 28 April 2025, Singapore, Singapore*. Ed. by EURECOM. Singapore.

Franchi, Gianni, Marwane Hariat, Xuanlong Yu, Nacim Belkhir, Antoine Manzanera, and David Filliat (2024). "InfraParis: A Multi-Modal and Multi-Task Autonomous Driving Dataset". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE, pp. 2973–2983.

Franzese, Giulio, Mustapha Bounoua, and Pietro Michiardi (2024). "MINDE: Mutual Information Neural Diffusion Estimation". In: *ICLR 2024, The Twelfth International Conference on Learning Representations*. Vienna, Austria.

Franzese, Giulio, Mattia Martini, Giulio Corallo, Paolo Papotti, and Pietro Michiardi (2025). "Latent Abstractions in Generative Diffusion Models". In: *Entropy* 27.4. ISSN: 1099-4300.

Franzese, Giulio, Simone Rossi, Lixuan Yang, Alessandro Finamore, Dario Rossi, Maurizio Filippone, and Pietro Michiardi (2023). "How much is enough? a study on diffusion times in score-based generative models". In: *Entropy*.

Fréchet, Maurice (1951). "Sur les tableaux de corrélation dont les marges sont données". In: *Annales de l'Université de Lyon. 3ê série, Sciences. Section A* 14, pp. 53–77.

Fu, Huan, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, Kun Zhang, and Dacheng Tao (2019). "Geometry-consistent generative adversarial networks for one-sided unsupervised domain mapping". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2427–2436.

Ganmor, Elad, Ronen Segev, and Elad Schneidman (2011). "Sparse low-order interaction network underlies a highly correlated and learnable neural population code". In: *Proceedings of the National Academy of sciences* 108.23, pp. 9679–9684.

Gao, Shuyang, Greg Ver Steeg, and Aram Galstyan (2015). "Efficient estimation of mutual information for strongly dependent variables". In: *Artificial intelligence and statistics*. PMLR, pp. 277–286.

Gat, Itay and Naftali Tishby (1998). "Synergy and redundancy among brain cells of behaving monkeys". In: *Advances in neural information processing systems* 11.

Ge, Yuying, Yizhuo Li, Yixiao Ge, and Ying Shan (2024). "Divot: Diffusion Powers Video Tokenizer for Comprehension and Generation". In: *arXiv preprint arXiv:2412.04432*.

Giacomo, Giuseppe Di, Giulio Franzese, Tania Cerquitelli, Carla Fabiana Chiasserini, and Pietro Michiardi (2024). "DiMViS: Diffusion-based Multi-View Synthesis". In: *ICML 2024 Workshop on Structured Probabilistic Inference & Generative Modeling*.

Gonzalez, Rafael C (2009). *Digital image processing*. Pearson education india.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep Learning*. http://www.deeplearningbook.org. MIT Press.

Goodfellow, Ian J., Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio (2014). *Generative Adversarial Networks*. arXiv: 1406.2661 [stat.ML].

Gu, Junyi, Artjom Lind, Tek Raj Chhetri, Mauro Bellone, and Raivo Sell (2023). "End-to-End Multimodal Sensor Dataset Collection Framework for Autonomous Vehicles". In: *Sensors* 23.15, p. 6783.

Guo, Wenzhong, Jianwen Wang, and Shiping Wang (2019). "Deep multimodal representation learning: A survey". In: *Ieee Access* 7, pp. 63373–63394.

Gutknecht, Aaron J., Abdullah Makkeh, and Michael Wibral (2023). "From Babel to Boole: The Logical Organization of Information Decompositions". In: *ArXiv* abs/2306.00734.

Hanchate, Abhishek, Himanshu Balhara, Vishal S. Chindepalli, and Satish T. S. Bukkapatnam (2024). "Process signature-driven high spatio-temporal resolution alignment of multimodal data". In: *arXiv preprint arXiv:2403.06888*.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

He, Ruifei, Shuyang Sun, Xin Yu, Chuhui Xue, Wenqing Zhang, Philip Torr, Song Bai, and XIAOJUAN QI (2023). "IS SYNTHETIC DATA FROM GENERATIVE MODELS READY FOR IMAGE RECOGNITION?" In: *The Eleventh International Conference on Learning Representations*.

Hessel, Jack, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi (2021). "Clipscore: A reference-free evaluation metric for image captioning". In: *arXiv preprint arXiv:2104.08718*.

Heusel, Martin, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter (2017). "Gans trained by a two time-scale update rule converge to a local nash equilibrium". In: *Advances in neural information processing systems* 30.

Hjelm, R Devon, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio (2019). "Learning deep representations by mutual information estimation and maximization". In: *International Conference on Learning Representations*.

Ho, Jonathan, Ajay Jain, and Pieter Abbeel (2020). "Denoising diffusion probabilistic models". In: *Advances in Neural Information Processing Systems* 33, pp. 6840–6851.

Ho, Jonathan and Tim Salimans (2022). "Classifier-free diffusion guidance". In: *arXiv preprint arXiv:2207.12598*.

Hong, Wenyi, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang (2023). "CogVideo: Large-scale Pretraining for Text-to-Video Generation via Transformers". In: *The Eleventh International Conference on Learning Representations*.

Hu, Anthony, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado (2023a). "Gaia-1: A generative world model for autonomous driving". In: *arXiv preprint arXiv:2309.17080*.

Hu, Minghui, Chuanxia Zheng, Zuopeng Yang, Tat-Jen Cham, Heliang Zheng, Chaoyue Wang, Dacheng Tao, and Ponnuthurai N. Suganthan (2023b). "Unified Discrete Dif-

fusion for Simultaneous Vision-Language Generation". In: *The Eleventh International Conference on Learning Representations*.

Huang, Chin-Wei, Jae Hyun Lim, and Aaron C Courville (2021). "A variational perspective on diffusion-based generative models and score matching". In: *Advances in Neural Information Processing Systems* 34, pp. 22863–22876.

Huang, Sicong, Alireza Makhzani, Yanshuai Cao, and Roger Grosse (2020a). "Evaluating lossy compression rates of deep generative models". In: *International Conference on Machine Learning*. PMLR.

Huang, Xun, Ming-Yu Liu, Serge Belongie, and Jan Kautz (2018). "Multimodal unsupervised image-to-image translation". In: *NIPS*.

Huang, Xun, Arun Mallya, Ting-Chun Wang, and Ming-Yu Liu (2022). "Multimodal Conditional Image Synthesis With Product-of-Experts GANs". In: *Computer Vision – ECCV 2022: 17th European Conference*. Berlin, Heidelberg: Springer-Verlag, 91–109. ISBN: 978-3-031-19786-4.

Huang, Zhiyu, Chen Lv, Yang Xing, and Jingda Wu (2020b). "Multi-modal sensor fusion-based deep neural network for end-to-end autonomous driving with scene understanding". In: *IEEE Sensors Journal* 21.10, pp. 11781–11790.

Hwang, HyeongJoo, Geon-Hyeong Kim, Seunghoon Hong, and Kee-Eung Kim (2021). "Multi-View Representation Learning via Total Correlation Objective". In: *Advances in Neural Information Processing Systems* 34, pp. 12194–12207.

Ibrahimi, Sarah, Mina Ghadimi Atigh, Nanne van Noord, Pascal Mettes, and Marcel Worring (2024). "Intriguing Properties of Hyperbolic Embeddings in Vision-Language Models". In: *Transactions on Machine Learning Research*.

Ivanovic, Boris, Karen Leung, Edward Schmerling, and Marco Pavone (2020). "Multimodal deep generative models for trajectory prediction: A conditional variational autoencoder approach". In: *IEEE Robotics and Automation Letters* 6.2, pp. 295–302.

Jaegle, Andrew, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira (2021). "Perceiver: General perception with iterative attention". In: *International conference on machine learning*. PMLR, pp. 4651–4664.

Javaloy, Adrian, Maryam Meghdadi, and Isabel Valera (July 2022). "Mitigating Modality Collapse in Multimodal VAEs via Impartial Optimization". In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato. Vol. 162. Proceedings of Machine Learning Research. PMLR, pp. 9938–9964.

Javidian, Mohammad Ali, Vaneet Aggarwal, Fanglin Bao, and Zubin Jacob (2021). "Quantum entropic causal inference". In: *Quantum Information and Measurement*. Optica Publishing Group, F2C–3.

Jumper, John, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. (2021). "Highly accurate protein structure prediction with AlphaFold". In: *nature* 596.7873, pp. 583–589.

Kantorovich, Leonid (1942). "On the translocation of masses". In: *C.R. (Doklady) Acad. Sci. URSS (N.S.)* 37, pp. 199–201.

Kaplanis, Christos, Pedro Mediano, and Fernando Rosas (2023). "Learning causally emergent representations". In: *NeurIPS 2023 workshop: Information-Theoretic Principles in Cognitive Systems*.

Karras, Tero (2017). "Progressive Growing of GANs for Improved Quality, Stability, and Variation". In: *arXiv preprint arXiv:1710.10196*.

Karras, Tero, Samuli Laine, and Timo Aila (2019). "A style-based generator architecture for generative adversarial networks". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410.

Kester, Lennart and Alexander van Oudenaarden (2018). "Single-cell transcriptomics meets lineage tracing". In: *Cell Stem Cell* 23, pp. 166–179.

Kilgour, Kevin, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi (2019). "Fréchet Audio Distance: A Reference-Free Metric for Evaluating Music Enhancement Algorithms". In: *Proc. Interspeech 2019*, pp. 2350–2354.

Kim, J (2019). "U-gat-it: unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation". In: *arXiv preprint arXiv:1907.10830*.

Kim, Jin-Hwa, Yunji Kim, Jiyoung Lee, Kang Min Yoo, and Sang-Woo Lee (2022). "Mutual Information Divergence: A Unified Metric for Multimodal Generative Models". In: *arXiv preprint arXiv:2205.13445*.

Kingma, Diederik P and Jimmy Ba (2014). "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980*.

Kingma, Diederik P, Tim Salimans, Ben Poole, and Jonathan Ho (2021). "Variational Diffusion Models". In: *Advances in Neural Information Processing Systems*. Ed. by A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan.

Kingma, Diederik P and Max Welling (2013). "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114*.

Kingma, Durk P and Prafulla Dhariwal (2018). "Glow: Generative flow with invertible 1x1 convolutions". In: *Advances in neural information processing systems* 31.

Klindžić, Rajna Šošić, Bartul Šiljeg, and Hrvoje Kalafatić (2024). "Multiscale and Multitemporal Remote Sensing for Neolithic Settlement Detection and Protection—The Case of Gorjani, Croatia". In: *Remote Sensing* 16.5, p. 736.

Kline, Adrienne, Hanyin Wang, Yikuan Li, Saya Dennis, Meghan Hutch, Zhenxing Xu, Fei Wang, Feixiong Cheng, and Yuan Luo (2022). "Multimodal machine learning in precision health: A scoping review". In: *npj Digital Medicine* 5, p. 171.

Kocaoglu, Murat, Alexandros Dimakis, Sriram Vishwanath, and Babak Hassibi (2017). "Entropic causal inference". In: *Proceedings of the AAAI Conference on Artificial Intelligence*.

Kolchinsky, Artemy (2019). "A novel approach to multivariate redundancy and synergy". In: *CoRR* abs/1908.08642. arXiv: 1908.08642.

Kong, Xianghao, Rob Brekelmans, and Greg Ver Steeg (2022). "Information-Theoretic Diffusion". In: *International Conference on Learning Representations*.

Kong, Xianghao, Ollie Liu, Han Li, Dani Yogatama, and Greg Ver Steeg (2024). "Interpretable Diffusion via Information Decomposition". In: *The Twelfth International Conference on Learning Representations*.

Kovačević, Mladen, Ivan Stanojević, and Vojin Šenk (2012). "On the hardness of entropy minimization and related problems". In: *2012 IEEE Information Theory Workshop*. IEEE, pp. 512–516.

Kovacevic, Mladen, Ivan Stanojevic, and Vojin Senk (2013). "On the Entropy of Couplings". In: *CoRR* abs/1303.3235. arXiv: 1303.3235.

Kraskov, Alexander, Harald Stögbauer, and Peter Grassberger (2004). "Estimating mutual information". In: *Physical review E* 69.6, p. 066138.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger. Vol. 25. Curran Associates, Inc.

Latham, Peter E and Sheila Nirenberg (2005). "Synergy, redundancy, and independence in population codes, revisited". In: *Journal of Neuroscience* 25.21, pp. 5195–5206.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton (2015). "Deep learning". In: *nature* 521.7553, pp. 436–444.

Lee, Cheng-Han, Ziwei Liu, Lingyun Wu, and Ping Luo (2019). "MaskGAN: Towards Diverse and Interactive Facial Image Manipulation". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5548–5557.

Lee, Kimin, Hao Liu, Moonkyung Ryu, Olivia Watkins, Yuqing Du, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, and Shixiang Shane Gu (2023). "Aligning text-to-image models using human feedback". In: *arXiv preprint arXiv:2302.12192*.

Lee, Soochan, Junsoo Ha, and Gunhee Kim (2019). "Harmonizing Maximum Likelihood with GANs for Multimodal Conditional Generation". In: *International Conference on Learning Representations*.

Letizia, Nunzio A, Nicola Novello, and Andrea M Tonello (2023). "Variational f-Divergence and Derangements for Discriminative Mutual Information Estimation". In: *arXiv preprint arXiv:2305.20025*.

Letizia, Nunzio A and Andrea M Tonello (2022). "Copula Density Neural Estimation". In: *arXiv preprint arXiv:2211.15353*.

Li, Cheuk Ting (2021). "Efficient Approximate Minimum Entropy Coupling of Multiple Probability Distributions". In: *IEEE Transactions on Information Theory* 67.8, pp. 5259–5268.

Li, Jinlong, Baolu Li, Zhengzhong Tu, Xinyu Liu, Qing Guo, Felix Juefei-Xu, Runsheng Xu, and Hongkai Yu (2024). "Light the Night: A Multi-Condition Diffusion Framework for Unpaired Low-Light Enhancement in Autonomous Driving". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15205–15215.

Li, Junnan, Dongxu Li, Silvio Savarese, and Steven Hoi (2023). "BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models". In: *Proceedings of the 40th International Conference on Machine Learning*.

Liang, Paul Pu, Yun Cheng, Xiang Fan, Chun Kai Ling, Suzanne Nie, Richard Chen, Zihao Deng, Nicholas Allen, Randy Auerbach, Faisal Mahmood, Ruslan Salakhutdinov, and Louis-Philippe Morency (2023). "Quantifying & Modeling Multimodal Interactions: An Information Decomposition Framework". In: *Advances in Neural Information Processing Systems*.

Liang, Paul Pu, Chun Kai Ling, Yun Cheng, Alexander Obolenskiy, Yudong Liu, Rohan Pandey, Alex Wilf, Louis-Philippe Morency, and Russ Salakhutdinov (2024). "Multimodal Learning Without Labeled Multimodal Data: Guarantees and Applications". In: *The Twelfth International Conference on Learning Representations*.

Liang, Paul Pu, Amir Zadeh, and Louis-Philippe Morency (2022). "Foundations and Trends in Multimodal Machine Learning: Principles, Challenges, and Open Questions". In: *arXiv preprint arXiv:2209.03430*.

Liang, Weixin, Yuhui Zhang, Yongchan Kwon, Serena Yeung, and James Zou (2022). "Mind the Gap: Understanding the Modality Gap in Multi-modal Contrastive Representation Learning". In: *Advances in Neural Information Processing Systems*. Vol. 35, pp. 7024–7037.

Lin, Gwo Dong, Xiaoling Dou, Satoshi Kuriki, and Jin-Sheng Huang (2014). "Recent developments on the construction of bivariate distributions with fixed marginals". In: *Journal of Statistical Distributions and Applications* 1, pp. 1–23.

Lipman, Yaron, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le (2022). "Flow matching for generative modeling". In: *arXiv preprint arXiv:2210.02747*.

Liu, Haotian, Chunyuan Li, Qingyang Wu, and Yong Jae Lee (2023). "Visual Instruction Tuning". In: *Advances in Neural Information Processing Systems*. Vol. 36.

Liu, Jing, Yujian Huang, Ritambhara Singh, Jean-Philippe Vert, and William Stafford Noble (2019). "Jointly Embedding Multiple Single-Cell Omics Measurements". In: *Proceedings of the 19th Workshop on Algorithms in Bioinformatics (WABI)*. Vol. 143. Leibniz International Proceedings in Informatics (LIPIcs). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 10:1–10:12.

Loaiza-Ganem, Gabriel, Brendan Leigh Ross, Jesse C Cresswell, and Anthony L. Caterini (2022). "Diagnosing and Fixing Manifold Overfitting in Deep Generative Models". In: *Transactions on Machine Learning Research*. ISSN: 2835-8856.

Lu, Zhou (2023). "A Theory of Multimodal Learning". In: *Advances in Neural Information Processing Systems*. Vol. 36.

Lugmayr, Andreas, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool (2022). "Repaint: Inpainting using denoising diffusion probabilistic models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11461–11471.

MacKay, David JC (2003). *Information theory, inference and learning algorithms*. Cambridge university press.

Makkeh, Abdullah, Aaron J Gutknecht, and Michael Wibral (2021). "Introducing a differentiable measure of pointwise shared information". In: *Physical Review E* 103.3, p. 032149.

Manzoor, Muhammad Arslan, Sarah Albarri, Ziting Xian, Zaiqiao Meng, Preslav Nakov, and Shangsong Liang (2023). "Multimodality Representation Learning: A Survey on

Evolution, Pretraining and Its Applications". In: *ACM Transactions on Multimedia Computing, Communications, and Applications* 20.3, 74:1–74:34.

Mao, Xudong, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley (2017). "Least squares generative adversarial networks". In: *Proceedings of the IEEE international conference on computer vision*, pp. 2794–2802.

Martin-Turrero, Carmen, Maxence Bouvier, Manuel Breitenstein, Pietro Zanuttigh, and Vincent Parret (2024). "ALERT-Transformer: Bridging Asynchronous and Synchronous Machine Learning for Real-Time Event-based Spatio-Temporal Data". In: *Proceedings of the 41st International Conference on Machine Learning*. Vol. 235.

Martinez Mediano, Pedro Antonio (2022). "Integrated information theory in complex neural systems". In.

McAllester, David and Karl Stratos (2020). "Formal limitations on the measurement of mutual information". In: *International Conference on Artificial Intelligence and Statistics*.

McCulloch, Warren S and Walter Pitts (1943). "A logical calculus of the ideas immanent in nervous activity". In: *The bulletin of mathematical biophysics* 5, pp. 115–133.

Meng, Chenlin, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon (2021). "Sdedit: Guided image synthesis and editing with stochastic differential equations". In: *arXiv preprint arXiv:2108.01073*.

Min, Sewon, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer (2022). *Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?* arXiv: 2202.12837 [cs.CL].

Monge, Gaspard (1781). "Mémoire sur la théorie des déblais et des remblais". In: *Histoire de l'Académie Royale des Sciences de Paris, avec les Mémoires de Mathématique et de Physique pour la même année*, pp. 666–704.

Moon, Young-Il, Balaji Rajagopalan, and Upmanu Lall (1995). "Estimation of mutual information using kernel density estimators". In: *Physical Review E* 52.3, p. 2318.

Murphy, Kevin P (2012). *Machine learning: a probabilistic perspective*. MIT press.

Nguyen, XuanLong, Martin J Wainwright, and Michael Jordan (2007). "Estimating divergence functionals and the likelihood ratio by penalized convex risk minimization". In: *Advances in Neural Information Processing Systems*.

Nichol, Alex, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen (2022). *GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models*. arXiv: 2112.10741 [cs.CV].

Nichol, Alexander Quinn and Prafulla Dhariwal (2021). "Improved denoising diffusion probabilistic models". In: *International conference on machine learning*. PMLR, pp. 8162–8171.

Nowozin, Sebastian, Botond Cseke, and Ryota Tomioka (2016). "F-GAN: Training Generative Neural Samplers Using Variational Divergence Minimization". In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. NIPS'16. Red Hook, NY, USA: Curran Associates Inc., 271–279. ISBN: 9781510838819.

Øksendal, Bernt (2003). *Stochastic differential equations*. Springer.

Oksendal, Bernt (2013). *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media.

Oord, Aaron van den, Yazhe Li, and Oriol Vinyals (2018). "Representation learning with contrastive predictive coding". In: *Advances in neural information processing systems*.

Painsky, Amichai, Saharon Rosset, and Meir Feder (2013). "Memoryless representation of Markov processes". In: *2013 IEEE International Symposium on Information Theory*. IEEE, pp. 2294–2298.

Palumbo, Emanuele, Imant Daunhawer, and Julia E Vogt (2023). "MMVAE+: Enhancing the Generative Quality of Multimodal VAEs without Compromises". In: *The Eleventh International Conference on Learning Representations*.

Pang, Yingxue, Jianxin Lin, Tao Qin, and Zhibo Chen (2021). "Image-to-image translation: Methods and applications". In: *IEEE Transactions on Multimedia* 24, pp. 3859–3881.

Paninski, Liam (2003). "Estimation of entropy and mutual information". In: *Neural computation* 15.6, pp. 1191–1253.

Papamakarios, George, Theo Pavlakou, and Iain Murray (2017). "Masked autoregressive flow for density estimation". In: *Advances in neural information processing systems* 30.

Parcalabescu, Letitia, Nils Trost, and Anette Frank (2021). "What is multimodality?" In: *arXiv preprint arXiv:2103.06304*.

Park, Taesung, Alexei A Efros, Richard Zhang, and Jun-Yan Zhu (2020). "Contrastive learning for unpaired image-to-image translation". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16*. Springer, pp. 319–345.

Peebles, William and Saining Xie (2023). "Scalable diffusion models with transformers". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205.

Peyré, Gabriel and Marco Cuturi (2019). "Computational Optimal Transport: With Applications to Data Science". In: *Foundations and Trends in Machine Learning* 11.5-6, pp. 355–607.

Pizer, Stephen M, E Philip Amburn, John D Austin, Robert Cromartie, Ari Geselowitz, Trey Greer, Bart ter Haar Romeny, John B Zimmerman, and Karel Zuiderveld (1987). "Adaptive histogram equalization and its variations". In: *Computer vision, graphics, and image processing* 39.3, pp. 355–368.

Poole, Ben, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker (2019). "On variational bounds of mutual information". In: *International Conference on Machine Learning*.

Radford, Alec, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever (July 2021). "Learning Transferable Visual Models From Natural Language Supervision". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, pp. 8748–8763.

Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. (2019). "Language models are unsupervised multitask learners". In: *OpenAI blog* 1.8, p. 9.

Rainforth, Tom, Rob Cornish, Hongseok Yang, Andrew Warrington, and Frank Wood (2018). "On nesting monte carlo estimators". In: *International Conference on Machine Learning*. PMLR, pp. 4267–4276.

Ramesh, Aditya, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen (2022). *Hierarchical Text-Conditional Image Generation with CLIP Latents*.

Rezende, Danilo and Shakir Mohamed (2015). "Variational inference with normalizing flows". In: *International conference on machine learning*. PMLR, pp. 1530–1538.

Rhodes, Benjamin, Kai Xu, and Michael U Gutmann (2020). "Telescoping density-ratio estimation". In: *Advances in neural information processing systems*.

Risken, Hannes (1996). "Fokker-Planck Equation". In: *The Fokker-Planck Equation: Methods of Solution and Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 63–95. ISBN: 978-3-642-61544-3.

Rombach, Robin, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer (June 2022). "High-Resolution Image Synthesis With Latent Diffusion Models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695.

Rosas, Fernando E., Pedro A. M. Mediano, Michael Gastpar, and Henrik Jeldtoft Jensen (2019). "Quantifying High-order Interdependencies via Multivariate Extensions of the Mutual Information". In: *Physical review. E* 100 3-1, p. 032305.

Rosas, Fernando E., Pedro A. M. Mediano, Borzoo Rassouli, and Adam Barrett (2020). "An operational information decomposition via synergistic disclosure". In: *Journal of Physics A: Mathematical and Theoretical* 53.

Roy, Debashri, Yuanyuan Li, Tong Jian, Peng Tian, Kaushik Chowdhury, and Stratis Ioannidis (2023). "Multi-Modality Sensing and Data Fusion for Multi-Vehicle Detection". In: vol. 25, pp. 2280–2295.

Ruan, Ludan, Yiyang Ma, Huan Yang, Huiguo He, Bei Liu, Jianlong Fu, Nicholas Jing Yuan, Qin Jin, and Baining Guo (2023). *MM-Diffusion: Learning Multi-Modal Diffusion Models for Joint Audio and Video Generation*. arXiv: 2212.09478 [cs.CV].

Runge, Jakob, Sebastian Bathiany, Erik Bollt, Gustau Camps-Valls, Dim Coumou, Ethan Deyle, Clark Glymour, Marlene Kretschmer, Miguel D Mahecha, Jordi Muñoz-Marí, et al. (2019). "Inferring causation from time series in Earth system sciences". In: *Nature communications* 10.1, p. 2553.

Saharia, Chitwan, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Raphael Gontijo-Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi (2022a). "Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding". In: *Advances in Neural Information Processing Systems*. Ed. by Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho.

Saharia, Chitwan, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Raphael Gontijo-Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi (2022b). "Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding". In: *Advances in Neural Information Processing Systems*. Ed. by Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho.

Sariyildiz, Mert Bulent, Karteek Alahari, Diane Larlus, and Yannis Kalantidis (2023). *Fake it till you make it: Learning transferable representations from synthetic ImageNet clones*. arXiv: 2212.08420 [cs.CV].

Sasaki, Hiroshi, Chris G Willcocks, and Toby P Breckon (2021). "Unit-ddpm: Unpaired image translation with denoising diffusion probabilistic models". In: *arXiv preprint arXiv:2104.05358*.

Scagliarini, Tomas, Daniele Marinazzo, Yike Guo, Sebastiano Stramaglia, and Fernando E. Rosas (2021). "Quantifying high-order interdependencies on individual patterns via the local O-information: Theory and applications to music analysis". In: *Physical Review Research*.

Scagliarini, Tomas, D. Nuzzi, Yuri Antonacci, Luca Faes, Fernando Rosas, Daniele Marinazzo, and Sebastiano Stramaglia (Jan. 2023). "Gradients of O-information: Low-order descriptors of high-order dependencies". In: *Physical Review Research* 5.

Schiebinger, Geoffrey, Jun Shu, Maciej Tabaka, Brian Cleary, V. Subramanian, A. Solomon, S. Liu, S. Lin, P. Berube, L. Lee, J. Chen, J. Brumbaugh, P. Rigollet, K. Hochedlinger, R. Jaenisch, A. Regev, and E. S. Lander (2019). "Optimal-Transport Analysis of Single-Cell Gene Expression Identifies Developmental Trajectories in Reprogramming". In: *Cell* 176.4, 928–943.e22.

Schuhmann, Christoph, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev (2022). *LAION-5B: An open large-scale dataset for training next generation image-text models*. arXiv: 2210.08402 [cs.CV].

Schulman, John, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov (2017). "Proximal policy optimization algorithms". In: *arXiv preprint arXiv:1707.06347*.

Scirè, Alessandro (2024). "Emergence and Criticality in Spatiotemporal Synchronization: The Complementarity Model". In: *Artificial Life* 30.4, pp. 508–522.

Shannon, Claude Elwood (1948). "A mathematical theory of communication". In: *The Bell system technical journal* 27.3, pp. 379–423.

Shi, Yuge, Siddharth N, Brooks Paige, and Philip Torr (2019). "Variational Mixture-of-Experts Autoencoders for Multi-Modal Deep Generative Models". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc.

Shi, Yuge, Brooks Paige, Philip Torr, and Siddharth N (2021). "Relating by Contrasting: A Data-efficient Framework for Multimodal Generative Models". In: *International Conference on Learning Representations*.

Shukor, Mustafa, Corentin Dancette, Alexandre Ramé, and Matthieu Cord (2023). "UnIVAL: Unified Model for Image, Video, Audio and Language Tasks". In: *Transactions on Machine Learning Research*.

Singer, Uriel, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman (2022). *Make-A-Video: Text-to-Video Generation without Text-Video Data*. arXiv: 2209.14792 [cs.CV].

Sohl-Dickstein, Jascha, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli (July 2015). "Deep Unsupervised Learning using Nonequilibrium Thermodynamics". In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, pp. 2256–2265.

Sokota, Samuel, Christian A Schroeder De Witt, Maximilian Igl, Luisa M Zintgraf, Philip Torr, Martin Strohmeier, Zico Kolter, Shimon Whiteson, and Jakob Foerster (2022). "Communicating via markov decision processes". In: *International Conference on Machine Learning*. PMLR, pp. 20314–20328.

Song, Jiaming and Stefano Ermon (2019a). "Understanding the Limitations of Variational Mutual Information Estimators". In: *International Conference on Learning Representations*.

Song, Jiaming, Chenlin Meng, and Stefano Ermon (2020). "Denoising diffusion implicit models". In: *arXiv preprint arXiv:2010.02502*.

Song, Yang, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever (2023). "Consistency models". In.

Song, Yang, Conor Durkan, Iain Murray, and Stefano Ermon (2021a). "Maximum likelihood training of score-based diffusion models". In: *Advances in Neural Information Processing Systems* 34, pp. 1415–1428.

Song, Yang and Stefano Ermon (2019b). "Generative Modeling by Estimating Gradients of the Data Distribution". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc.

Song, Yang and Stefano Ermon (2020). "Improved Techniques for Training Score-Based Generative Models". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., pp. 12438–12448.

Song, Yang, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole (2021b). "Score-Based Generative Modeling through Stochastic Differential Equations". In: *International Conference on Learning Representations*.

Sparacino, Laura, Luca Faes, Gorana Mijatović, Giuseppe Parla, Vincenzina Lo Re, Roberto Miraglia, Jean de Ville de Goyet, and Gianvincenzo Sparacia (2023). "Statistical Approaches to Identify Pairwise and High-Order Brain Functional Connectivity Signatures on a Single-Subject Basis". In: *Life* 13.

Srivastava, Sameer, Juan E. Vargas, and Devis Tuia (2019). "Understanding urban landuse from the above and ground perspectives: a deep learning, multimodal solution". In: *Remote Sensing of Environment* 228, pp. 129–143.

Stein, Barry E and M Alex Meredith (1993). *The merging of the senses*. MIT press.

Stramaglia, Sebastiano, Tomas Scagliarini, Bryan C. Daniels, and Daniele Marinazzo (2021). "Quantifying Dynamical High-Order Interdependencies From the O-Information: An Application to Neural Spiking Dynamics". In: *Frontiers in Physiology* 11. ISSN: 1664-042X.

Stratos, Karl (2019). "Mutual Information Maximization for Simple and Accurate Part-Of-Speech Induction". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*.

Stuart, Tim, Andrew Butler, Paul Hoffman, Christoph Hafemeister, Efthymia Papalexi, William M Mauck, Yuhan Hao, Marlon Stoeckius, Peter Smibert, and Rahul Satija (2019). "Comprehensive integration of single-cell data". In: *Cell* 177.7, 1888–1902.e21.

Sun, Shikun, Longhui Wei, Junliang Xing, Jia Jia, and Qi Tian (2023). "SDDM: score-decomposed diffusion models on manifolds for unpaired image-to-image translation". In: *International Conference on Machine Learning*. PMLR, pp. 33115–33134.

Sun, TH (1975). "Linear dependence structure of the entropy space". In: *Inf Control* 29.4, pp. 337–68.

Sun Han, Te (1980). "Multiple mutual informations and multiple interactions in frequency data". In: *Information and Control* 46.1, pp. 26–45. ISSN: 0019-9958.

Sutter, Thomas M., Imant Daunhawer, and Julia E. Vogt (2020). "Multimodal Generative Learning Utilizing Jensen-Shannon-Divergence". In: *CoRR* abs/2006.08242.

Sutter, Thomas M., Imant Daunhawer, and Julia E Vogt (2021). "Generalized Multimodal ELBO". In: *International Conference on Learning Representations*.

Tang, Fuchou, Catalin Barbacioru, Yang Wang, Eric Nordman, Chao Lee, Na Xu, Xian Wang, John Bodeau, Brian B Tuch, Ahmed Siddiqui, Kaifu Lao, and M Azim Surani (2009). "mRNA-Seq whole-transcriptome analysis of a single cell". In: *Nature Methods* 6.5, pp. 377–382.

Tang, Shengkun, Yaqing Wang, Caiwen Ding, Yi Liang, Yao Li, and Dongkuan Xu (2024). "Adadiff: Accelerating diffusion models through step-wise adaptive computation". In: *European Conference on Computer Vision*. Springer, pp. 73–90.

Tang, Zineng, Ziyi Yang, Mahmoud Khademi, Yang Liu, Chenguang Zhu, and Mohit Bansal (2023a). "CoDi-2: In-Context, Interleaved, and Interactive Any-to-Any Generation". In: *arXiv preprint arXiv:2311.18775*.

Tang, Zineng, Ziyi Yang, Chenguang Zhu, Michael Zeng, and Mohit Bansal (2023b). *Any-to-Any Generation via Composable Diffusion*. arXiv: 2305.11846 [cs.CV].

Tao, Ming, Hao Tang, Fei Wu, Xiao-Yuan Jing, Bing-Kun Bao, and Changsheng Xu (2022). *DF-GAN: A Simple and Effective Baseline for Text-to-Image Synthesis*. arXiv: 2008.05865 [cs.CV].

Tax, Tycho M.S., Pedro A.M. Mediano, and Murray Shanahan (2017). "The Partial Information Decomposition of Generative Neural Network Models". In: *Entropy* 19.9. ISSN: 1099-4300.

Touvron, Hugo, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. (2023). "Llama 2: Open foundation and fine-tuned chat models". In: *arXiv preprint arXiv:2307.09288*.

Tran, Ba-Hien, Simone Rossi, Dimitrios Milios, Pietro Michiardi, Edwin V Bonilla, and Maurizio Filippone (2021). "Model selection for bayesian autoencoders". In: *Advances in Neural Information Processing Systems* 34, pp. 19730–19742.

Tran, Luan, Xiaoming Liu, Jiayu Zhou, and Rong Jin (July 2017). "Missing Modalities Imputation via Cascaded Residual Autoencoder". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Turing, A. M. (Oct. 1950). "I.—COMPUTING MACHINERY AND INTELLIGENCE". In: *Mind* LIX.236, pp. 433–460. ISSN: 0026-4423. eprint: https://academic.oup.com/mind/article-pdf/LIX/236/433/61209000/mind_lix_236_433.pdf.

Uehara, Masatoshi, Yulai Zhao, Kevin Black, Ehsan Hajiramezanali, Gabriele Scalia, Nathaniel Lee Diamant, Alex M Tseng, Tommaso Biancalani, and Sergey Levine (2024). "Fine-tuning of continuous-time diffusion models as entropy-regularized control". In: *arXiv preprint arXiv:2402.15194*.

Vahdat, Arash, Karsten Kreis, and Jan Kautz (2021). "Score-based Generative Modeling in Latent Space". In: *Advances in Neural Information Processing Systems*. Ed. by A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan.

Van Gansbeke, Wouter, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool (2020). "Scan: Learning to classify images without labels". In: *European conference on computer vision*. Springer, pp. 268–285.

Varley, Thomas F., Maria Pope, Joshua Faskowitz, and Olaf Sporns (2022). "Multivariate information theory uncovers synergistic subsystems of the human cerebral cortex". In: *Communications Biology* 6.

Varley, Thomas F., Maria Pope, Maria Grazia Puxeddu, Joshua Faskowitz, and Olaf Sporns (2023). "Partial entropy decomposition reveals higher-order information structures in human brain activity". In: *Proceedings of the National Academy of Sciences of the United States of America* 120.

Vasco, Miguel, Hang Yin, Francisco S. Melo, and Ana Paiva (Feb. 2022). "Leveraging hierarchy in multimodal generative models for effective cross-modality inference". In: *Neural Networks* 146, pp. 238–255. ISSN: 18792782.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin (2017). "Attention is all you need". In: *Advances in neural information processing systems* 30.

Venkatesh, Praveen, Corbett Bennett, Sam Gale, Tamina K. Ramirez, Greggory Heller, Severine Durand, Shawn R Olsen, and Stefan Mihalas (2023). "Gaussian Partial Information

Decomposition: Bias Correction and Application to High-dimensional Data". In: *Thirty-seventh Conference on Neural Information Processing Systems*.

Vidyasagar, Mathukumalli (2011). "Metrics Between Probability Distributions on Finite Sets of Different Cardinalities by Maximizing Mutual Information (MMI)". In: *CoRR* abs/1104.4521. arXiv: 1104.4521.

Vidyasagar, Mathukumalli (2012). "A metric between probability distributions on finite sets of different cardinalities and applications to order reduction". In: *IEEE Transactions on Automatic Control* 57.10.

Villani, Cédric (2009). *Optimal transport: old and new*. Vol. 338. Springer.

Vinay, Ashvala and Alexander Lerch (2022). "Evaluating generative audio systems and their metrics". In: *arXiv preprint arXiv:2209.00130*.

Vincent, Pascal (2011). "A Connection Between Score Matching and Denoising Autoencoders". In: *Neural Computation* 23.7, pp. 1661–1674.

Voleti, Vikram, Alexia Jolicoeur-Martineau, and Chris Pal (2022). "Mcvd-masked conditional video diffusion for prediction, generation, and interpolation". In: *Advances in neural information processing systems* 35, pp. 23371–23385.

Wah, Catherine, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie (2011). "The caltech-ucsd birds-200-2011 dataset". In.

Wang, Chao, Giulio Franzese, Alessandro Finamore, Massimo Gallo, and Pietro Michiardi (2025a). "Information Theoretic Text-to-Image Alignment". In: *The Thirteenth International Conference on Learning Representations*.

Wang, Chao, Giulio Franzese, Alessandro Finamore, and Pietro Michiardi (2025b). "RFMI: Estimating mutual information on rectified flow for text-to-image alignment". In: *ICLR 2025, DeLTa Workshop, Deep Generative Model in Machine Learning: Theory, Principle and Efficacy*. Ed. by EURECOM. Singapore.

Wang, Zhou, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli (2004). "Image quality assessment: from error visibility to structural similarity". In: *IEEE transactions on image processing* 13.4, pp. 600–612.

Watson, Daniel, Jonathan Ho, Mohammad Norouzi, and William Chan (2021). "Learning to efficiently sample from diffusion probabilistic models". In: *arXiv preprint arXiv:2106.03802*.

Welch, Joshua D, Alexander J Hartemink, and Jan F Prins (2017). "Matcher: manifold alignment reveals correspondence between single cell transcriptome and epigenome dynamics". In: *Genome Biology* 18.1, p. 138.

Welch, Joshua D, Velina Kozareva, Alexandre Ferreira, Charles Vanderburg, Cynthia Martin, and Evan Z Macosko (2019). "Single-cell multi-omic integration compares and contrasts features of brain cell identity". In: *Cell* 177.7, 1873–1887.e17.

Wesego, Daniel and Amirmohammad Rooshenas (2023). *Score-Based Multimodal Autoencoders*. arXiv: 2305.15708 [cs.LG].

Williams, Paul L. and Randall D. Beer (2010). *Nonnegative Decomposition of Multivariate Information*. arXiv: 1004.2515 [cs.IT].

Witt, Christian Schroeder de, Samuel Sokota, J Zico Kolter, Jakob Foerster, and Martin Strohmeier (2022). "Perfectly secure steganography using minimum entropy coupling". In: *arXiv preprint arXiv:2210.14889*.

Wu, Fuxiang, Liu Liu, Fusheng Hao, Fengxiang He, and Jun Cheng (June 2022). "Text-to-Image Synthesis Based on Object-Guided Joint-Decoding Transformer". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18113–18122.

Wu, Mike and Noah Goodman (2018). "Multimodal Generative Models for Scalable Weakly-Supervised Learning". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Vol. 31. Curran Associates, Inc.

Wu, Qitian, Rui Gao, and Hongyuan Zha (2021). "Bridging explicit and implicit deep generative models via neural stein estimators". In: *Advances in Neural Information Processing Systems* 34, pp. 11274–11286.

Wu, Shengqiong, Hao Fei, Leigang Qu, Wei Ji, and Tat-Seng Chua (2024). "NExT-GPT: Any-to-Any Multimodal LLM". In: *Proceedings of the 41st International Conference on Machine Learning*.

Wu, Xiaoshi, Keqiang Sun, Feng Zhu, Rui Zhao, and Hongsheng Li (Oct. 2023). "Human Preference Score: Better Aligning Text-to-Image Models with Human Preference". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2096–2105.

Wunder, Gerhard, Benedikt Groß, Rick Fritschek, and Rafael F Schaefer (2021). "A reverse Jensen inequality result with application to mutual information estimation". In: *2021 IEEE Information Theory Workshop (ITW)*.

Xi, Johnny, Jana Osea, Zuheng Xu, and Jason Hartford (2024). "Propensity Score Alignment of Unpaired Multimodal Data". In: *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Xia, Yan, Hai Huang, Jieming Zhu, and Zhou Zhao (2023). "Achieving Cross Modal Generalization with Multimodal Unified Representation". In: *Advances in Neural Information Processing Systems*. Vol. 36.

Xie, Sang Michael, Aditi Raghunathan, Percy Liang, and Tengyu Ma (2022). "An Explanation of In-context Learning as Implicit Bayesian Inference". In: *International Conference on Learning Representations*.

Xie, Shaoan, Yanwu Xu, Mingming Gong, and Kun Zhang (June 2023). "Unpaired Image-to-Image Translation With Shortest Path Regularization". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10177–10187.

Xu, Jiazheng, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong (2024). "Imagereward: Learning and evaluating human preferences for text-to-image generation". In: *Advances in Neural Information Processing Systems* 36.

Xue, Le, Manli Shu, Anas Awadalla, Jun Wang, An Yan, Senthil Purushwalkam, Honglu Zhou, Viraj Prabhu, Yutong Dai, Michael S Ryoo, Shrikant Kendre, Jieyu Zhang, Can Qin, Shu Zhang, Chia-Chih Chen, Ning Yu, Juntao Tan, Tulika Manoj Awalgaonkar, Shelby

Heinecke, Huan Wang, Yejin Choi, Ludwig Schmidt, Zeyuan Chen, Silvio Savarese, Juan Carlos Niebles, Caiming Xiong, and Ran Xu (2024). *xGen-MM (BLIP-3): A Family of Open Large Multimodal Models*. arXiv: 2408.08872 [cs.CV].

Yang, Ling, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang (2023a). "Diffusion models: A comprehensive survey of methods and applications". In: *ACM Computing Surveys* 56.4, pp. 1–39.

Yang, Shuai, Liming Jiang, Ziwei Liu, and Chen Change Loy (2023b). "Gp-unit: Generative prior for versatile unsupervised image-to-image translation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.10, pp. 11869–11883.

Yi, Zili, Hao Zhang, Ping Tan, and Minglun Gong (2017). "Dualgan: Unsupervised dual learning for image-to-image translation". In: *Proceedings of the IEEE international conference on computer vision*, pp. 2849–2857.

Yu, Lei and Vincent YF Tan (2018). "Asymptotic coupling and its applications in information theory". In: *IEEE Transactions on Information Theory* 65.3, pp. 1321–1344.

Zhang, Hao, Jin-Jian Xu, Hong-Wei Cui, Lin Li, Yaowen Yang, Chao-Sheng Tang, and Niklas Boers (2024). "When Geoscience Meets Foundation Models: Toward a general geoscience artificial intelligence system". In: *IEEE Geoscience and Remote Sensing Magazine*.

Zhang, Lvmin, Anyi Rao, and Maneesh Agrawala (2023). "Adding Conditional Control to Text-to-Image Diffusion Models". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3836–3847.

Zhang, Richard, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang (2018). "The unreasonable effectiveness of deep features as a perceptual metric". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595.

Zhang, Yue, Chengtao Peng, Qiuli Wang, Dan Song, Kaiyan Li, and S. Kevin Zhou (2023). *Unified Multi-Modal Image Synthesis for Missing Modality Imputation*. arXiv: 2304.05340 [cs.CV].

Zhang, Yuhui, Elaine Sui, and Serena Yeung-Levy (2024). "Connect, Collapse, Corrupt: Learning Cross-Modal Tasks with Uni-Modal Data". In: *International Conference on Learning Representations (ICLR)*.

Zhao, Min, Fan Bao, Chongxuan Li, and Jun Zhu (2022). "Egsde: Unpaired image-to-image translation via energy-guided stochastic differential equations". In: *Advances in Neural Information Processing Systems* 35, pp. 3609–3623.

Zhao, Shengjia, Jiaming Song, and Stefano Ermon (2018). "A Lagrangian Perspective on Latent Variable Generative Models". In: *Proc. 34th Conference on Uncertainty in Artificial Intelligence*.

Zheng, Chuanxia, Tat-Jen Cham, and Jianfei Cai (2021). "The spatially-correlative loss for various image translation tasks". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16407–16417.

Zheng, Wenzhao, Ruiqi Song, Xianda Guo, Chenming Zhang, and Long Chen (2024). "Genad: Generative end-to-end autonomous driving". In: *European Conference on Computer Vision*. Springer, pp. 87–104.

Zhu, Jun-Yan, Taesung Park, Phillip Isola, and Alexei A. Efros (2017). "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2223–2232.