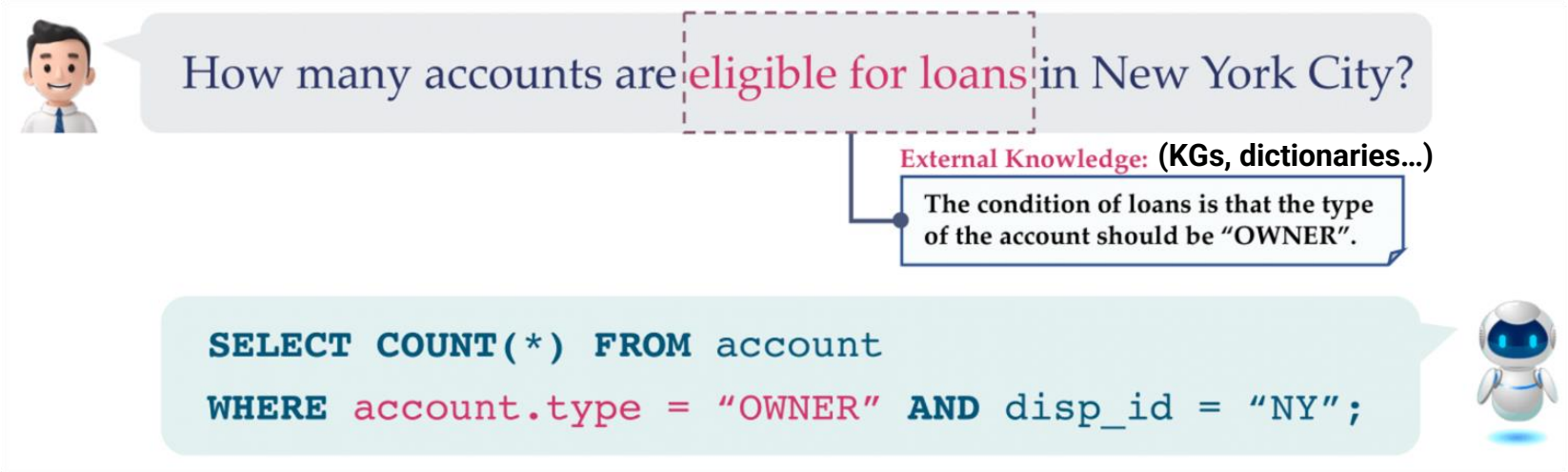
Reinforcement Learning to Enable Reasoning LLMs for Text2SQL

Paolo Papotti

Tabular Data Analysis (TaDA) Workshop, 5 Sept 2025

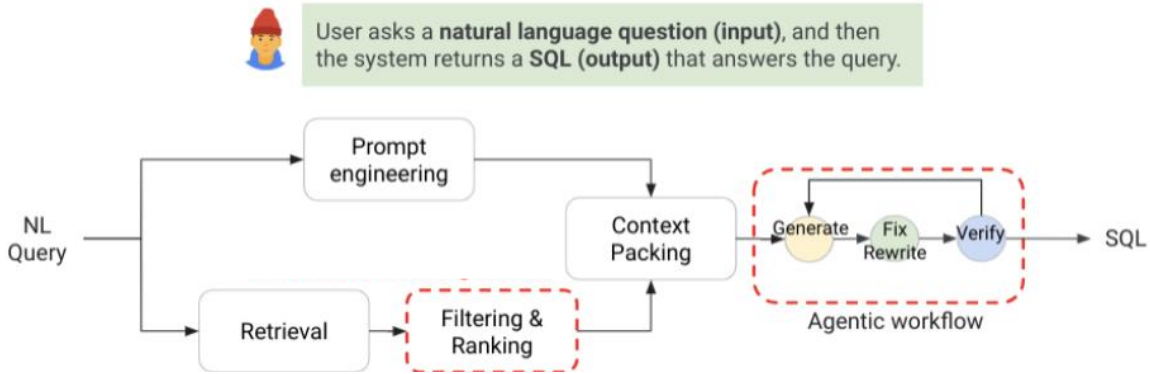
What is Text2SQL?

The task of mapping a **natural language question** on a database to the corresponding executable **SQL query**.



Li, Jinyang, et al. "Can llm already serve as a database interface? a big bench for large-scale database grounded text-to-sqls." Advances in Neural Information Processing Systems 36 (2023): 42330-42357.

State-Of-Art Text2SQL Pipelines



Long Context NL2SQL	BIRD Dev Ex Acc (%) Simple	Moderate	Challenging	Overall
+ All DB table schema	52.22 (-)	30.82 (-)	32.41 (-)	43.87 (-)
+ Hints	67.35 (↑ **15.13**)	51.08 (↑ **20.26**)	44.14 (↑ **11.73**)	60.23 (↑ **16.36**)
+ Sample column values	68.11 (↑ 0.76)	53.66 (↑ **2.58**)	49.66 (↑ **5.52**)	61.99 (↑ 1.76)
+ Self-correction	71.03 (↑ **2.92**)	56.25 (↑ **2.59**)	52.41 (↑ **2.75**)	64.80 (↑ **2.81**)
+ Disambiguation	71.14 (↑ 0.11)	57.54 (↑ 1.29)	55.17 (↑ **2.76**)	65.51 (↑ 0.71)
+ Synthetic examples	72.32 (↑ 1.18)	59.05 (↑ 1.51)	53.79 (↓ **1.38**)	66.56 (↑ 1.05)
+ Verify & retry[b]	72.65 (↑ 0.33)	59.70 (↑ 0.65)	55.86 (↑ **2.07**)	67.14 (↑ 0.58)

Chung, Yeounoh, et al. "Is Long Context All You Need? Leveraging LLM's Extended Context for NL2SQL." arXiv preprint arXiv:2501.12372 (2025).

nature

View all journals | Search

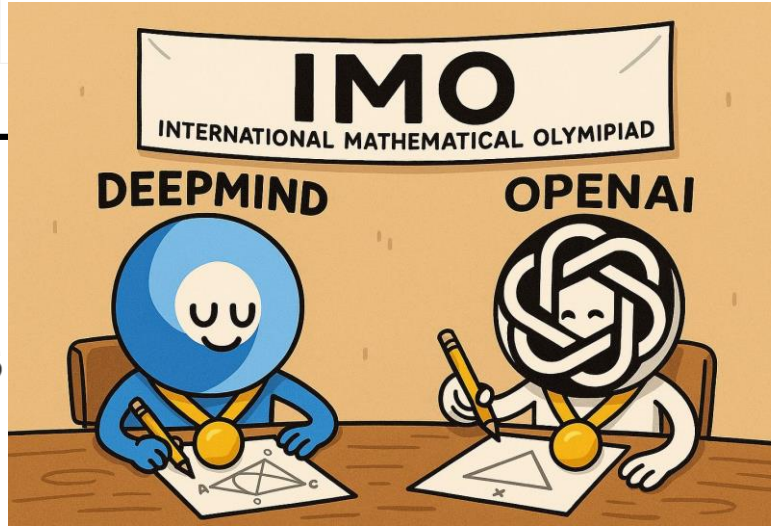
Explore content | About the journal | Publish with us

nature > news > article

NEWS | 24 July 2025

DeepMind and OpenAI models solve maths problems at level of top students

For the first time, large language models performed on a par with gold medallists in the International Mathematical Olympiad.



Google DeepMind | Build with Gemini | Try Gemini

RESEARCH

Advanced version of Gemini with Deep Think officially achieves gold-medal standard at the International Mathematical Olympiad

21 JULY 2025

Thang Luong and Edward Lockhart

OpenAI ✓
@OpenAI

Follow

We achieved gold medal-level performance 🥇 on the 2025 International Mathematical Olympiad with a general-purpose reasoning LLM!

Our model solved world-class math problems—at the level of top human contestants. A major milestone for AI and mathematics.

Research Question

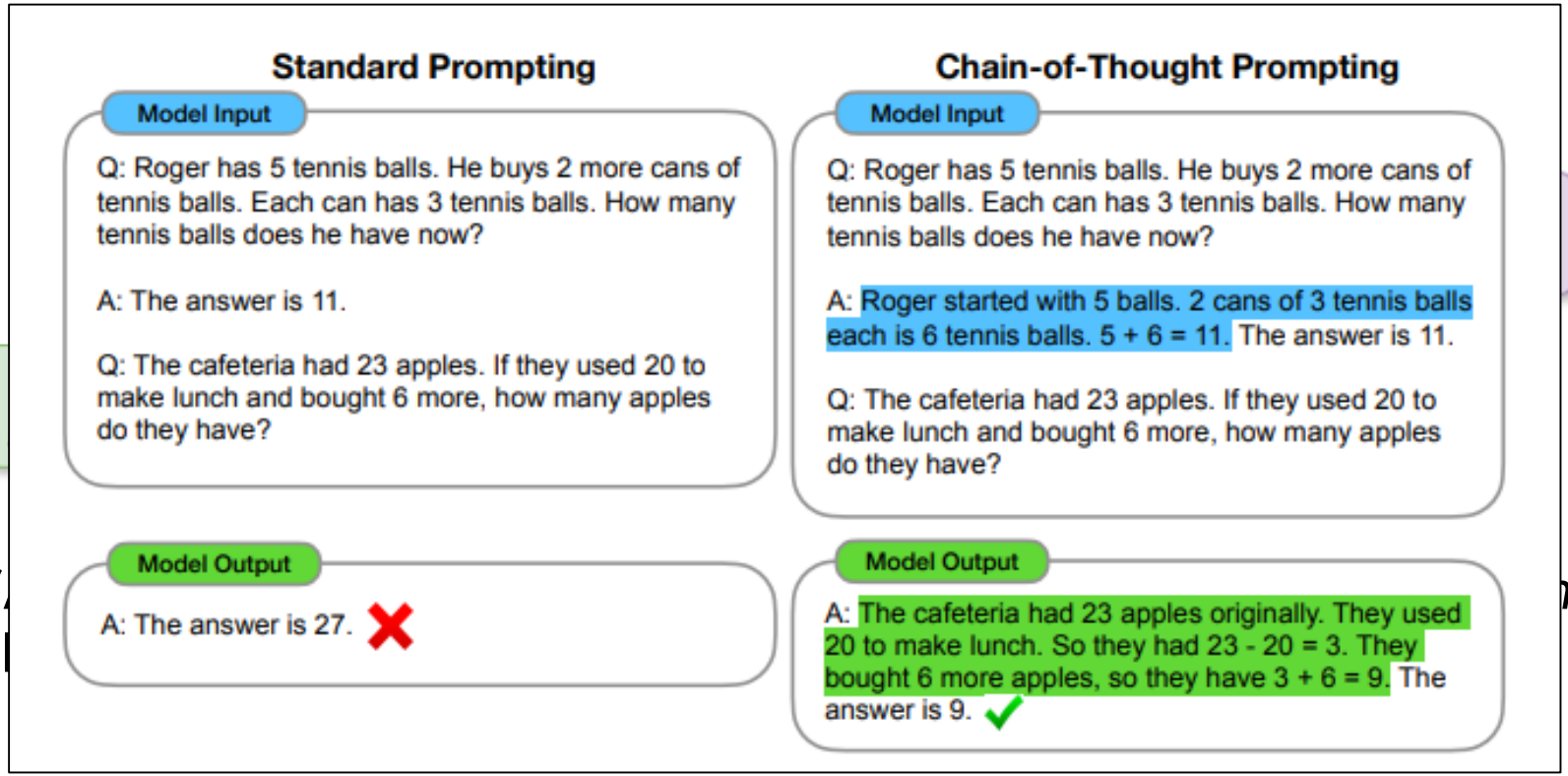
To what extent do **reasoning** capabilities embedded or learned by LLMs **enhance Text2SQL performance** across **different training paradigms**?

Important Remark

We are not trying to achieve SOTA!

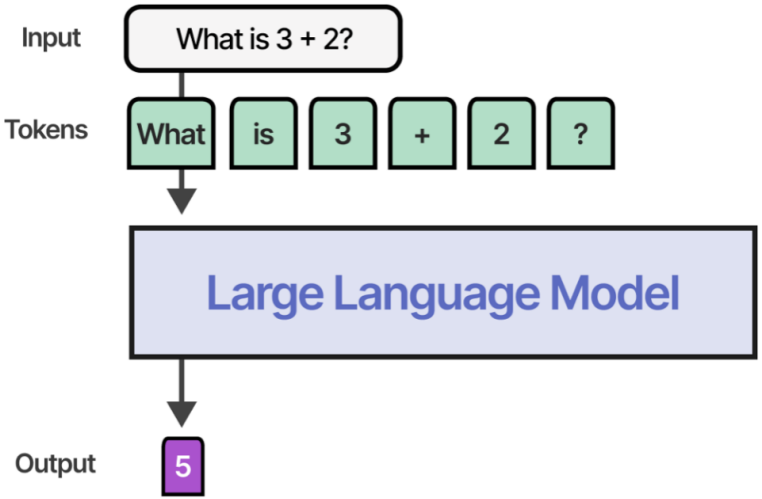
- No prompt engineering, self-reflection, self-consistency, or ensembling
- Tables to answer the questions are provided (Oracle Table retrieval)

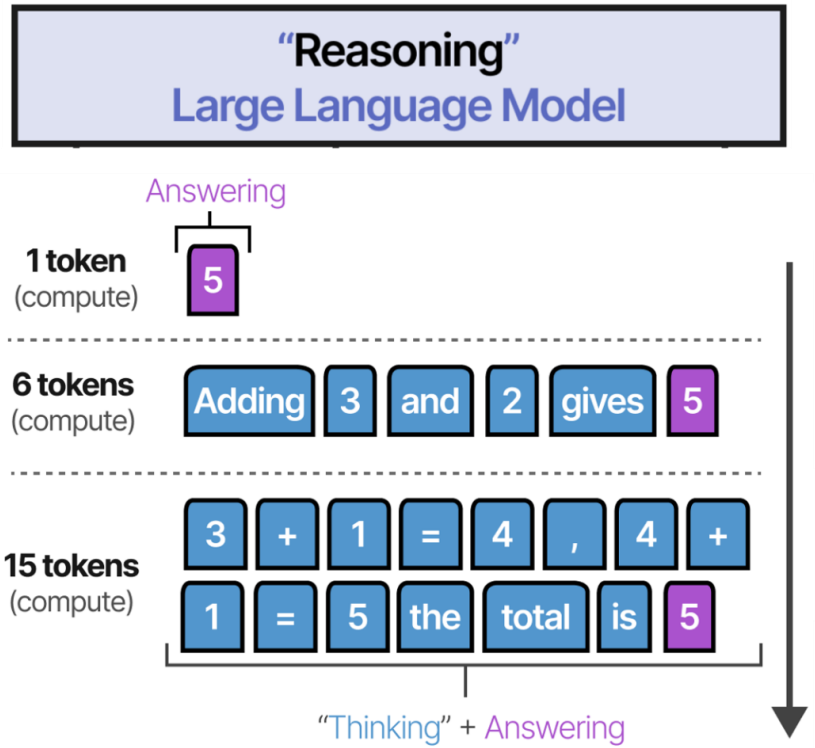
The key assumption



Kambhampati et al. *Stop Anthropomorphizing Intermediate Tokens as Reasoning/Thinking Traces!* Arxiv 2025

Background 1: **What are reasoning models?**

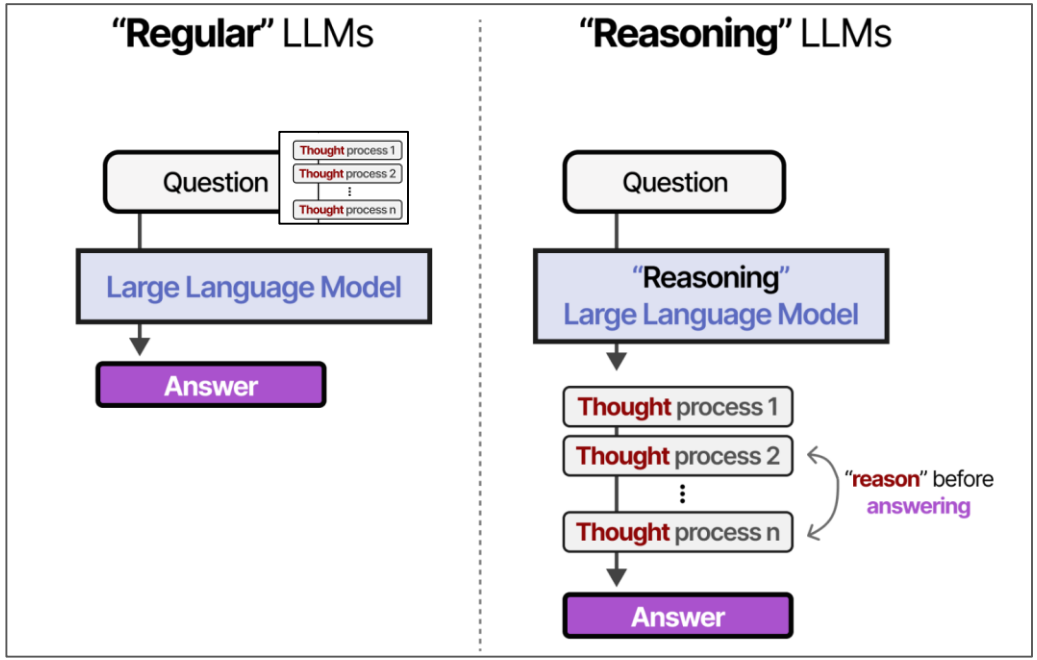


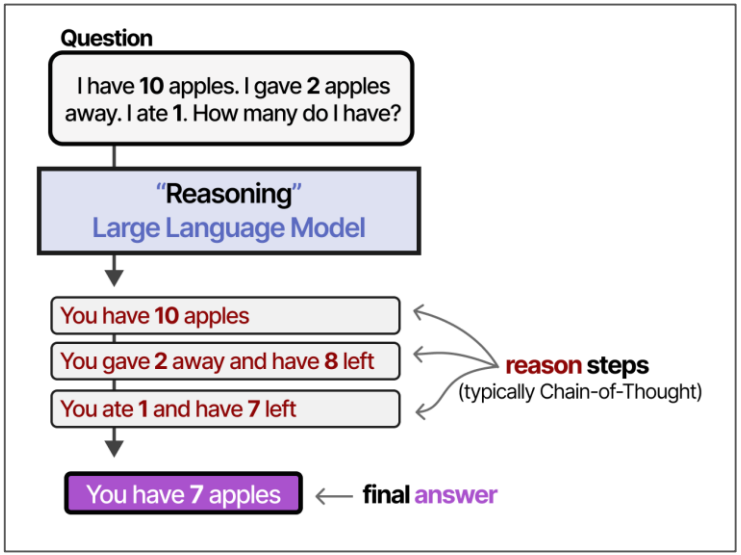


More tokens → more compute
→ better performance

https://newsletter.maartengrootendorst.com/p/a-visual-guide-to-reasoning-llms

Background 2: **Difference with Chain-of-Thought?**





LLMs: *Manually* specify in the prompt the step-by-step procedure for the problem (CoT)
Reasoning LLMs: *Automatically* generates the "**best**" step-by-step formulation → **reasoning trace**

https://platform.openai.com/docs/guides/reasoning-best-practices

Background 3: **Supervised-Fine-Tuning (SFT)**

Thinking traces are “distilled” from DeepSeek-R1 *(+ Gemini Pro)*

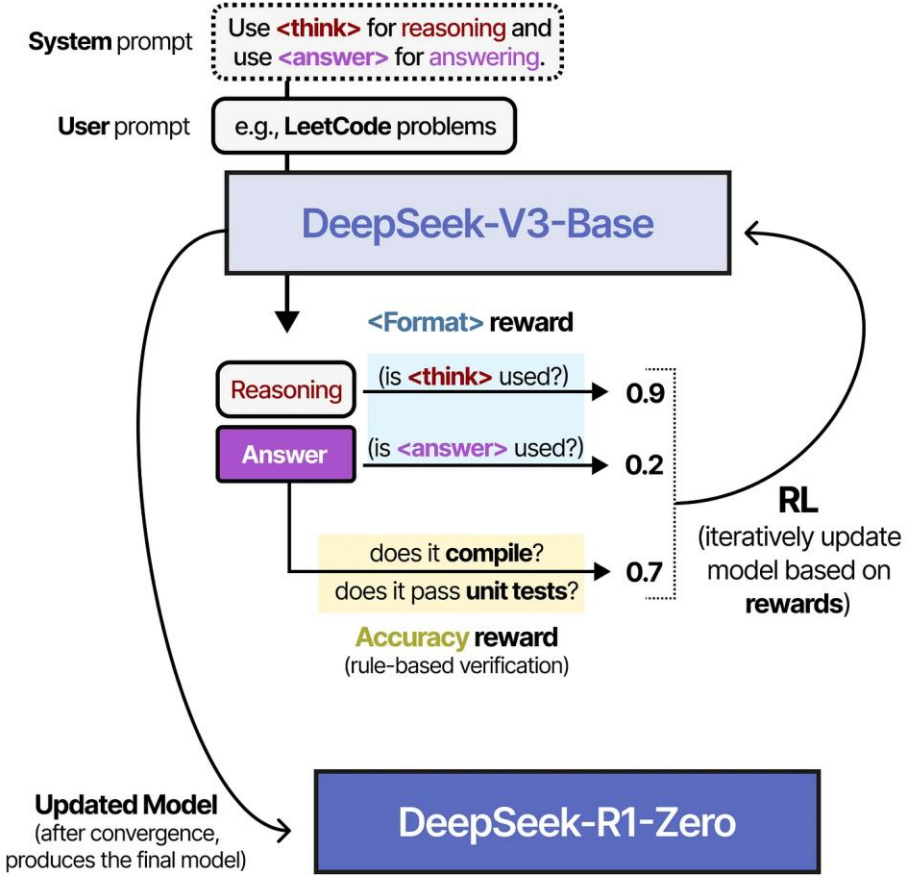
Complete training/inference prompt.

<|im_start|>system
You are a helpful AI Assistant that provides well-reasoned and detailed responses. You first think about the reasoning process as an internal monologue and then provide the user with the answer. Respond in the following format: <think>
...
</think>
<answer>
...
</answer><|im_end|>
<|im_start|>user
Answer the following question with the SQL code. Use the piece of evidence and base your answer on the database schema. Given the question, the evidence and the database schema, return in the <answer> tags only the SQL script that addresses the question.
Question:
Calculate the average of 2020’s population in each zip code.
Evidence:
average of 2020 population in each zip code refers to Divide (Sum(population_2020), Count(zip_code))
Database Schema:
CREATE TABLE zip_data
(
zip_code INTEGER
primary key,
city TEXT,
state TEXT,
multi_county TEXT,
”1st_quarter_payroll” INTEGER,
...
foreign key (state) references state(abbreviation),
foreign key (CBSA) references CBSA(CBSA)
)
Return only the SQL script enclosed in <answer> tags.<|im_end|>
<|im_start|>assistant

Thinking generated trace

```
<think>
To solve this problem, we need to follow these steps:
1. Identify the patients who have a body mass index (BMI) recorded
in the observations table.
2. Filter the records to find the lowest BMI value.
3. Retrieve the full name (first and last) of the patient with the
lowest BMI.

The evidence provided indicates that the description we are looking for
is "Body Mass Index" and the units should be 'kg/m2'.
We also need to find the minimum value of the BMI.

We can use a subquery to find the minimum value of BMI and then
join this with the patients table to get the full name of the patient.
</think>
<answer>
SELECT p.first, p.last FROM patients p
JOIN (
    SELECT PATIENT, MIN(VALUE) AS min_bmi
    FROM observations
    WHERE DESCRIPTION = 'Body Mass Index' AND UNITS = 'kg/m2'
    GROUP BY PATIENT
) o ON p.patient = o.PATIENT
WHERE o.min_bmi IN (
    SELECT MIN(VALUE)
    FROM observations
    WHERE DESCRIPTION = 'Body Mass Index' AND UNITS = 'kg/m2'
)
</answer>
```

Background 4: **RL with Verifiable Rewards (RLVR) – GRPO**



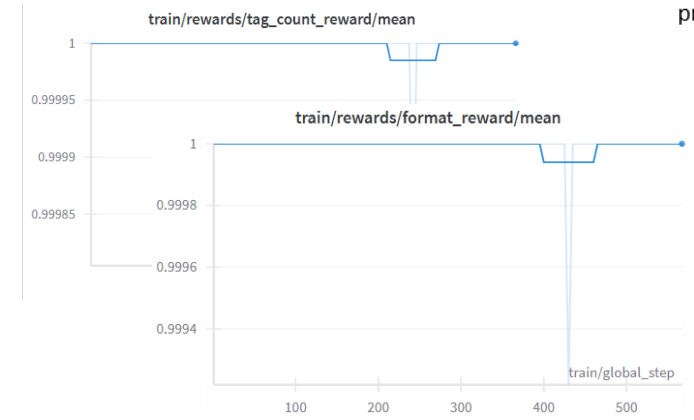
Training procedure

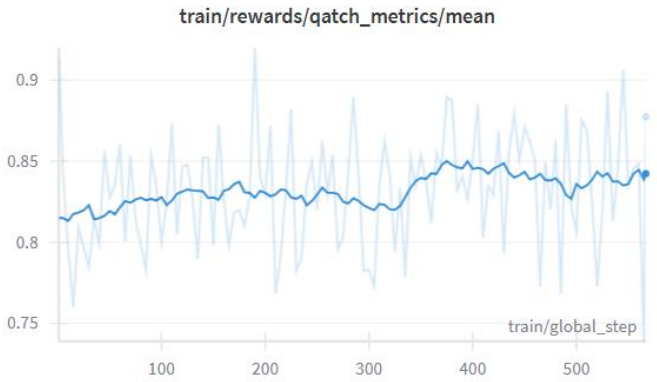
Start from a base/SFT model:

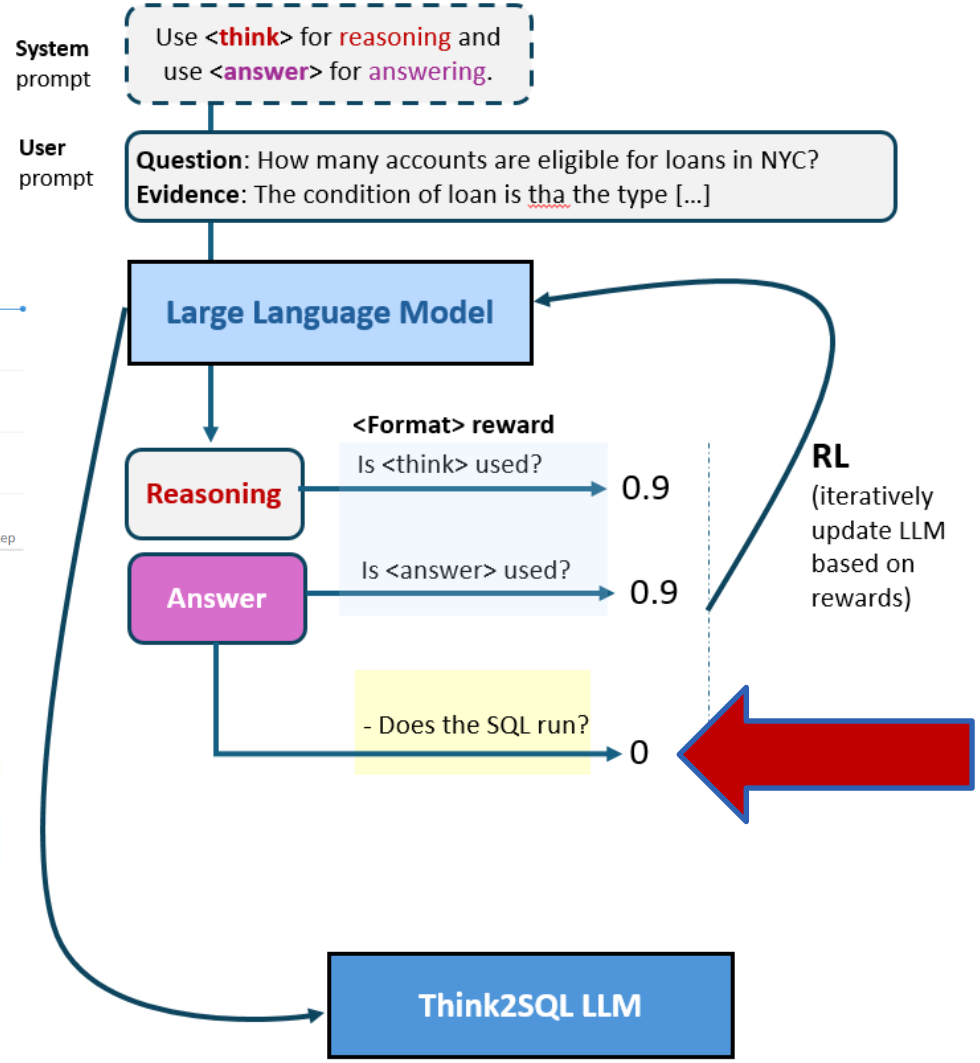
1. Generate G-possible answers
2. For each answer, calculate the rewards
3. Update the *advantage* of each response based on all the rewards
4. Update the model parameters based on each response and its relative advantage

https://newsletter.maartengrootendorst.com/p/a-visual-guide-to-reasoning-llms

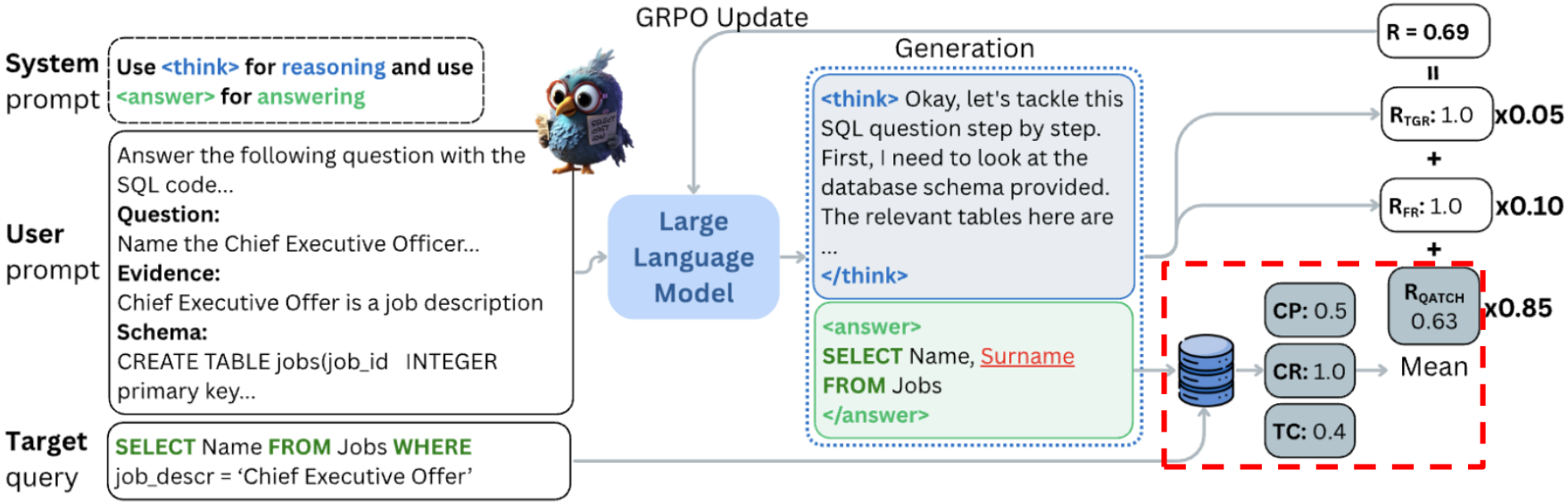
RLVR for **Text2SQL**







RLVR for Text2SQL with **Partial** Rewards



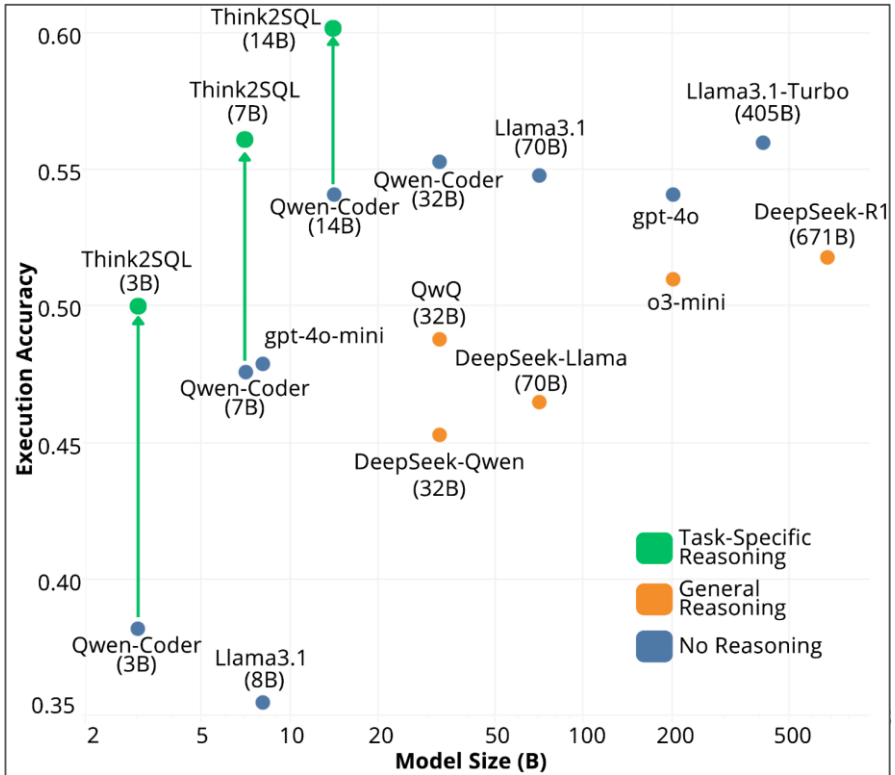
Partial rewards evaluate **precision**, **recall**, and **cardinality** of partially correct answers

S. Papicchio, P. Papotti, and L. Cagliero. Qatch: Automatic evaluation of sql-centric tasks on proprietary data. ACM Transactions on Intelligent Systems and Technology, 2025.

Think2SQL

Research Question:
To what extent do **reasoning** capabilities embedded or learned by LLMs **enhance Text2SQL performance** across **different training paradigms**?

- Impact of **LLM training regimes**: Zero-Shot, SFT, RLVR, and their combination
- Impact of **rewards** for RLVR



Takeaway 1: SFT with reasoning traces improves performance

SFT with reasoning improves over base models, especially for **smaller LLMs**.

Model	Simple	Medium	Challenging	Weighted AVG
Qwen-3B	0.469	0.267	0.196	0.382
Qwen-3B-SFT$_{NT}$	0.510	0.325	0.224	0.427
Qwen-3B-SFT	0.531	0.366	0.301	0.460
Qwen-7B	0.548	0.388	0.294	0.476
Qwen-7B-SFT$_{NT}$	0.537	0.310	0.273	0.443
Qwen-7B-SFT	0.573	0.40	0.294	0.494

Takeaway 2: Reasoning with RL improves performance

Task-specific reasoning is better than general reasoning.

Model	Reasoning	Simple	Medium	Challenging	Weigthed AVG
Open-source LLMs (< 10B)					
DeepSeek-Qwen-1.5B	✓	0.056	0.004	0.0	0.035
Qwen2.5-Coder-0.5B	✓	0.126	0.033	0.035	0.089
DeepSeek-Qwen-7B	✓	0.297	0.113	0.049	0.218
Qwen2.5-Coder-1.5B	✓	0.351	0.184	0.077	0.275
Llama-8b	✗	0.436	0.260	0.133	0.355
Qwen2.5-Coder-3B	✗	0.469	0.267	0.196	0.382
Qwen2.5-Coder-7B	✗	0.548	0.388	0.294	**0.476**
Open-source LLMs (10-100B)					
DeepSeek-Qwen-32B	✓	0.542	0.347	0.217	0.453
DeepSeek-Llama-70B	✓	0.552	0.371	0.203	0.465
QwQ-32B	✓	0.550	0.427	0.280	0.488
Qwen2.5-Coder-14B	✗	0.610	0.456	0.364	0.541
Llama-70B	✗	0.618	0.469	0.350	0.548
Qwen2.5-Coder-32B	✗	0.623	0.482	0.329	**0.553**
Open-source LLMs (>100B)					
DeepSeek-R1	✓	0.588	0.440	0.294	0.518
Llama-405B-Turbo	✗	0.630	0.477	0.371	**0.560**
Closed-source LLMs					
gpt-4o-mini-2024-07-18	✗	0.545	0.401	0.301	0.479
o3-mini-2025-01-31	✓	0.561	0.406	0.329	0.510
gpt-4o-2024-08-06	✗	0.619	0.447	0.343	**0.541**
Our Models					
Think2SQL-0.5B	✓	0.342	0.122	0.086	0.254
Think2SQL-1.5B	✓	0.526	0.333	0.226	0.442
Think2SQL-3B	✓	0.574	0.403	0.336	0.500
Think2SQL-7B	✓	0.628	0.482	0.385	0.561
Think2SQL-14B	✓	0.654	0.543	0.462	**0.602**

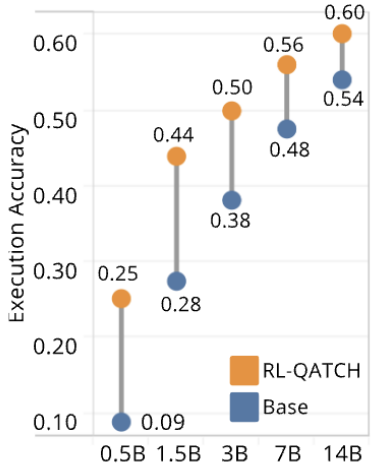


Figure 3: `Qwen-Coder` **size vs. EX. on BIRD Dev.**

Table 3: **Improvement with R_{QATCH} different model families.**

Model	EX
Llama-8B	0.355
Llama-8B-RL	0.509 ▲43%
DeepSeek-Qwen-7B	0.218
DeepSeek-Qwen-7B-RL	0.307 ▲41%

Takeaway 3: Dense Rewards work better

Dense Rewards as R_QATCH are more effective than sparse rewards like R_EX for RL.
RL only beats SFT+RL on BIRD test data.

Model	Simple	Medium	Challenging	Weighted AVG
Qwen-3B	0.469	0.267	0.196	0.382
Qwen-3B-SFT$_{NT}$	0.510	0.325	0.224	0.427
Qwen-3B-SFT	0.531	0.366	0.301	0.460
Qwen-3B-RL$_{EX}$	0.569	0.381	0.273	0.485
Qwen-3B-RL$_{QATCH}$	**0.574** ▲22%	**0.403** ▲51%	0.336	**0.500** ▲31%
Qwen-3B-SFT-RL$_{EX}$	0.560	0.370	**0.343** ▲75%	0.482
Qwen-3B-SFT-RL$_{QATCH}$	0.564	0.372	0.305	0.482
Qwen-7B	0.548	0.388	0.294	0.476
Qwen-7B-SFT$_{NT}$	0.537	0.310	0.273	0.443
Qwen-7B-SFT	0.573	0.40	0.294	0.494
Qwen-7B-RL$_{EX}$	0.619	0.463	0.406	0.552
Qwen-7B-RL$_{QATCH}$	**0.628** ▲15%	**0.482** ▲24%	0.385	**0.561** ▲18%
Qwen-7B-SFT-RL$_{EX}$	0.590	0.422	0.343	0.516
Qwen-7B-SFT-RL$_{QATCH}$	0.602	0.442	**0.427** ▲45%	0.537

Takeaway 4: SFT + RL is more generalizable

Combining **SFT+RL** yields the **most generalizable models** across diverse datasets.
This suggests that **SFT+RL** enables models to **better adapt** to **unseen scenarios**

Model	Spider	Spider-Syn	Spider-DK	KaggleDBQA
Qwen-3B	0.725	0.632	0.602	0.252
Qwen-3B-SFT	0.770	0.718	0.634	0.312
Qwen-3B-RL$_{QATCH}$	0.770	0.717	**0.680** ▲13%	0.360
Qwen-3B-SFT-RL$_{QATCH}$	**0.772** ▲6%	**0.725** ▲15%	0.655	**0.387** ▲54%
Qwen-7B	0.776	0.703	0.652	0.247
Qwen-7B-SFT	0.799	0.774	0.652	0.380
Qwen-7B-RL$_{QATCH}$	0.822	0.769	**0.731** ▲12%	0.436
Qwen-7B-SFT-RL$_{QATCH}$	**0.826** ▲6%	**0.794** ▲13%	0.687	**0.441** ▲79%

Open 1: reasoning traces quality?

```
<think>
To solve this problem, we need to follow these steps:
1. Identify the patients who have a body mass index (BMI) recorded
in the observations table.
2. Filter the records to find the lowest BMI value.
3. Retrieve the full name (first and last) of the patient with the
lowest BMI.

The evidence provided indicates that the description we are looking for
is "Body Mass Index" and the units should be 'kg/m2'.
We also need to find the minimum value of the BMI.

We can use a subquery to find the minimum value of BMI and then
join this with the patients table to get the full name of the patient.
</think>
<answer>
SELECT p.first, p.last FROM patients p
JOIN (
    SELECT PATIENT, MIN(VALUE) AS min_bmi
    FROM observations
    WHERE DESCRIPTION = 'Body Mass Index' AND UNITS = 'kg/m2'
    GROUP BY PATIENT
) o ON p.patient = o.PATIENT
WHERE o.min_bmi IN (
    SELECT MIN(VALUE)
    FROM observations
    WHERE DESCRIPTION = 'Body Mass Index' AND UNITS = 'kg/m2'
)
</answer>
```

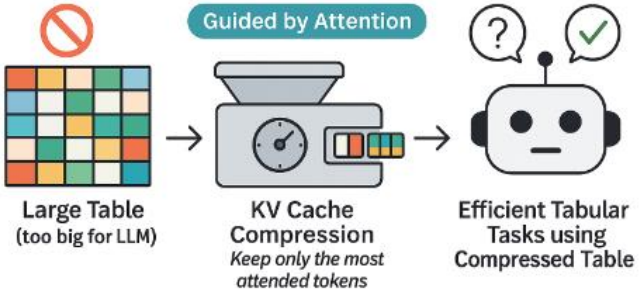
RL training

```
<think>
Okay, let's see. The question is asking for the names of products where
there have been over 3 price changes. The evidence says that "over 3
price changes" means the count of ListPrice greater than 3.
Hmm, maybe that's a misunderstanding.
Wait, count(ListPrice)>3 would mean more than three ListPrices,
but that doesn't make sense. Maybe it's actually referring to the
number of entries in ProductListPriceHistory where StartDate is not
null and ListPrice is not null. Because if a product has multiple
entries, each with different prices, that counts as a change.

So, the actual approach would be:
count the distinct Products where there are entries in
ProductListPriceHistory with ListPrice higher than a certain threshold.
But the evidence says over 3 refers to count(ListPrice)>3, which
confuses me. Maybe the intended solution counts distinct [...]
 But that might be more complex. For example:

SELECT DISTINCT p.Name
FROM "Product" p
JOIN ProductListPriceHistory h ON p.ProductID = h.ProductID
WHERE h.ListPrice > h.StartDate AND COUNT(DISTINCT h.StartDate) > 3;

But wait, that's not right.
because if the entries have the same StartDate (same ProductID),
that counts as one. For example, [...]
Find distinct products that have entries with ListPrice greater than
some threshold (say 3). For example:

SELECT DISTINCT Name
FROM Product
WHERE ProductID IN (
    SELECT DISTINCT ProductID
    FROM ProductListPriceHistory
    WHERE ListPrice > 3
);

[...]

Alternatively, the user might have intended that for each Product
in ProductListPriceHistory, if there's an entry with ListPrice >3,
then include that product. So the answer is selecting distinct names
from product, joining with ProductListPriceHistory where ListPrice >3.
</think>

<answer>
SELECT DISTINCT Name
```

SFT + RL training

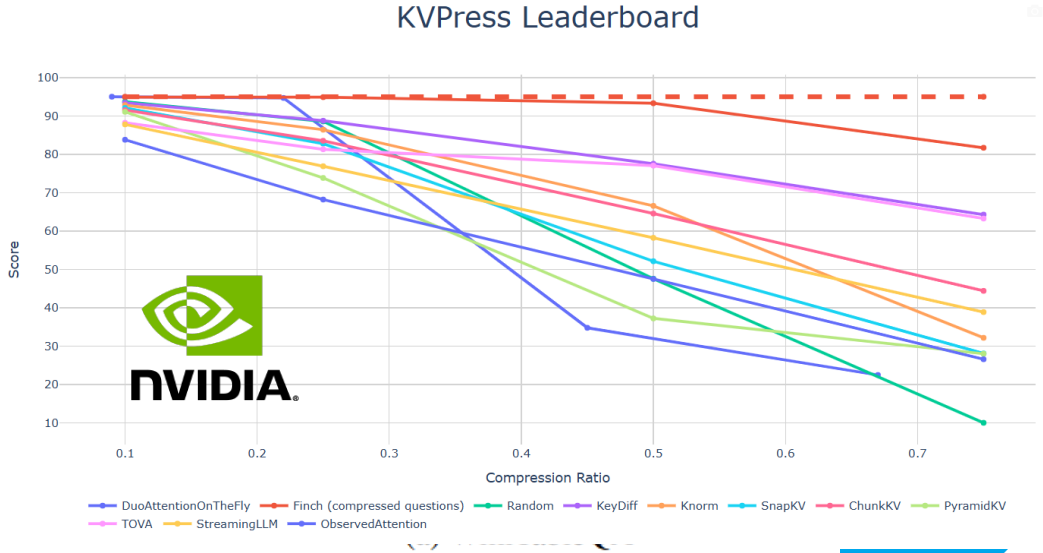
Open 2: more input, more output for better reasoning?

- Context is a bottleneck
  - Input: Opportunities for better quality
  - Output: Expensive at RL training time in terms of memory requirements
- Compression to the rescue?



Corallo et al. *TableKV: KV Cache Compression for In-Context Table Processing*. TRL@ACL'25

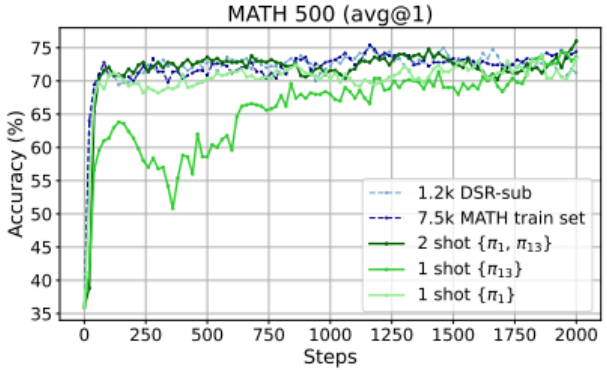
Long Context NL2SQL	BIRD Dev Ex Acc (%) Simple	Moderate	Challenging
+ All DB table schema	52.22 (-)	30.82 (-)	32.41 (-)
+ Hints	67.35 (↑ **15.13**)	51.08 (↑ **20.26**)	44.14 (↑ **11.73**)
+ Sample column values	68.11 (↑ 0.76)	53.66 (↑ **2.58**)	49.66 (↑ **5.52**)



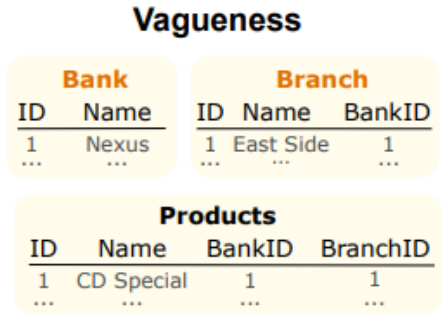
Corallo and Papotti. *FINCH: Prompt-guided Key-Value Cache Compression for Large Language Models*. TACL'24

Open 3: from pilots to solutions

- Efficiency at training
  - Running queries take a lot of time
  - Few examples to obtain comparable results?
    - MATH 500: 1 example vs 7.5K
    - Think2SQL: **256** ex. vs 9k → 2 points drop
- Efficiency at inference
  - Modeling SQL execution time in the rewards?
- Ambiguity in NL, Unanswerable Questions
  - Who issued “CD Special”?
  - Who produced “CD Special”?

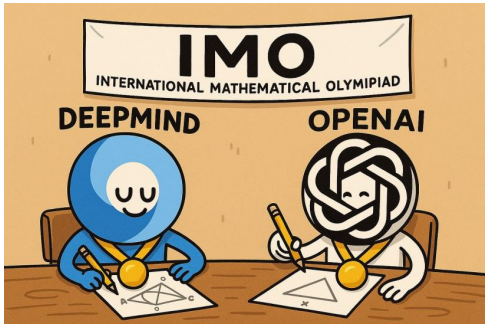


Reinforcement Learning for Reasoning in Large Language Models with One Training Example. Arxiv 2025



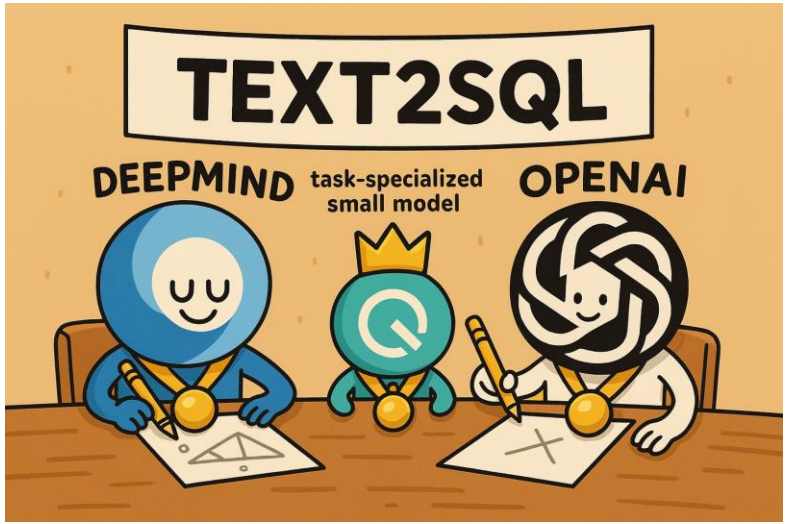
AMBROSIA: A Benchmark for Parsing Ambiguous Questions into Database Queries. Neurips 2024

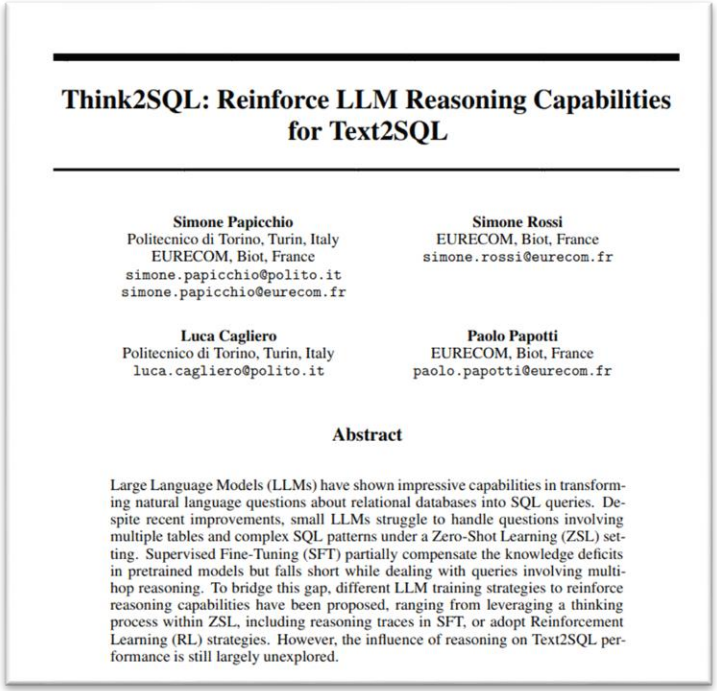
Can Human Annotation Be Replaced For Ambiguous And Unanswerable Text2SQL? [EMNLP’25]



Paper

Model





Think2SQL: Reinforce LLM Reasoning Capabilities for Text2SQL

Simone Papicchio
Politecnico di Torino, Turin, Italy
EURECOM, Biot, France
simone.papicchio@polito.it
simone.papicchio@eurecom.fr

Simone Rossi
EURECOM, Biot, France
simone.rossi@eurecom.fr

Luca Cagliero
Politecnico di Torino, Turin, Italy
luca.cagliero@polito.it

Paolo Papotti
EURECOM, Biot, France
paolo.papotti@eurecom.fr

Abstract

Large Language Models (LLMs) have shown impressive capabilities in transforming natural language questions about relational databases into SQL queries. Despite recent improvements, small LLMs struggle to handle questions involving multiple tables and complex SQL patterns under a Zero-Shot Learning (ZSL) setting. Supervised Fine-Tuning (SFT) partially compensate the knowledge deficits in pretrained models but falls short while dealing with queries involving multi-hop reasoning. To bridge this gap, different LLM training strategies to reinforce reasoning capabilities have been proposed, ranging from leveraging a thinking process within ZSL, including reasoning traces in SFT, or adopt Reinforcement Learning (RL) strategies. However, the influence of reasoning on Text2SQL performance is still largely unexplored.

https://arxiv.org/pdf/2504.15077

Thanks for your attention

Background: **GRPO Loss**

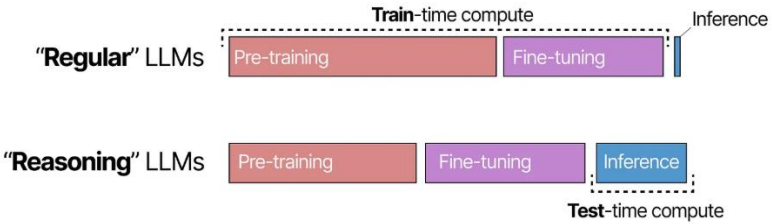
$$\mathcal{L}_{\text{GRPO}}(\boldsymbol{\theta}) = \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{T_i}\sum_{t=1}^{T_i}\min\left(p_{i,t}(\boldsymbol{\theta})A_i,\ \text{clip}\big(p_{i,t}(\boldsymbol{\theta}),1-\epsilon,1+\epsilon\big)A_i\right) - \beta\text{KL}\left[\pi_{\boldsymbol{\theta}} \parallel \pi_{\boldsymbol{\theta}_{\text{ref}}}\right]\right],$$

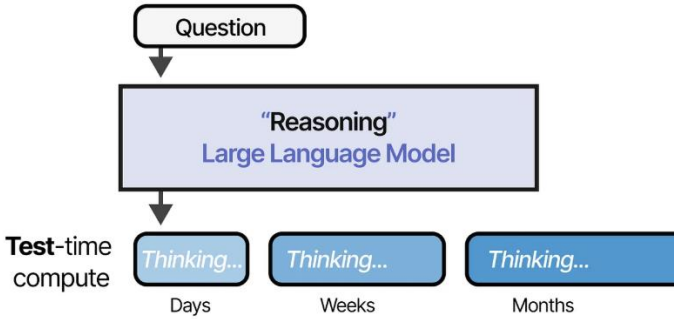
Token Level probability ratio

$$p_{i,t}(\boldsymbol{\theta}) = \frac{\pi_{\boldsymbol{\theta}}\big(y_{i,t} \mid \boldsymbol{s}_{i,t}\big)}{\pi_{\boldsymbol{\theta}_{\text{old}}}\big(y_{i,t} \mid \boldsymbol{s}_{i,t}\big)}$$

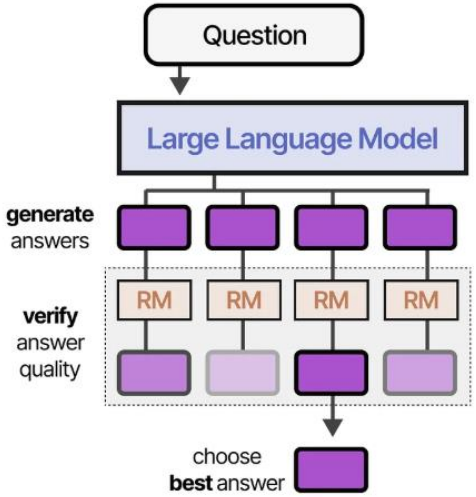
Advantage

$$A_i = \frac{R_i - \mathbb{E}[R_j]}{\sqrt{\mathbb{V}[R_j]}}, \qquad j \in \{1,\ldots,G\},$$

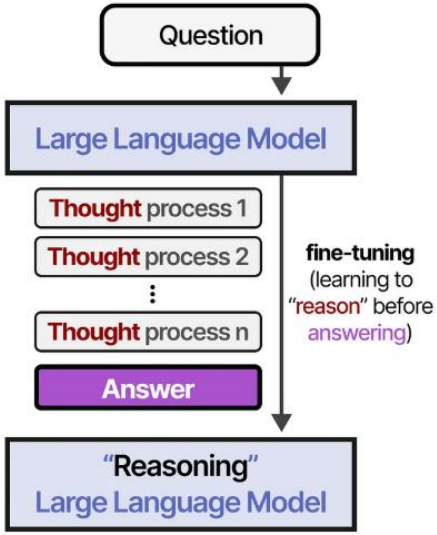




Search against Verifiers



Modifying Proposal Distribution



Impact of # examples

Table 8: Think2SQL-7B performance with reduced training samples and increased epochs, matching total gradient updates to the full-data setting.

Model	BIRD-Dev	KaggleDBQA	Spider	Spider-Syn	Spider-DK
Think2SQL-7B-9000-samples-1-epochs	**0.561**	0.441	**0.826**	**0.794**	0.697
Think2SQ-7B-256-samples-36-epochs	0.539	0.414	0.814	0.755	0.693
Think2SQ-7B-512-samples-18-epochs	0.544	**0.456**	0.813	0.773	0.701
Think2SQ-7B-1024-samples-9-epochs	0.550	0.432	0.817	0.783	**0.720**
Think2SQ-7B-2048-samples-5-epochs	0.552	0.432	0.811	0.775	0.706

Impact of RL on other tasks

Table 7: Impact of RLVR fine-tuning on diverse benchmarks for mathematical reasoning and code generation. Results are reported as mean accuracy ± standard deviation.

Model	AIME-2024@1:32	Math-500	GPQA Diamond	LiveCodeBench1:16
`Qwen2.5-Coder-7B`	0.07±0.03	**0.70±0.02**	**0.37±0.03**	**0.16±0.02**
`Qwen-7B-RL-QATCH`	0.08±0.03	**0.70±0.02**	0.34±0.03	0.15±0.01
`Qwen-7B-RL-EX`	**0.09±0.04**	0.698±0.02	0.30±0.03	**0.16±0.02**

Think2SQL

Training regimes:

1. Zero shot
2. SFT
3. RL
4. RL+SFT