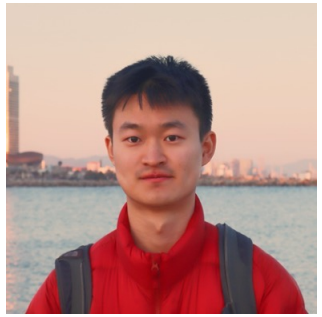
AbstRaL: Augmenting LLMs' Reasoning by Reinforcing Abstract Thinking



Silin Gao[1,2]

Antoine Bosselut[2]

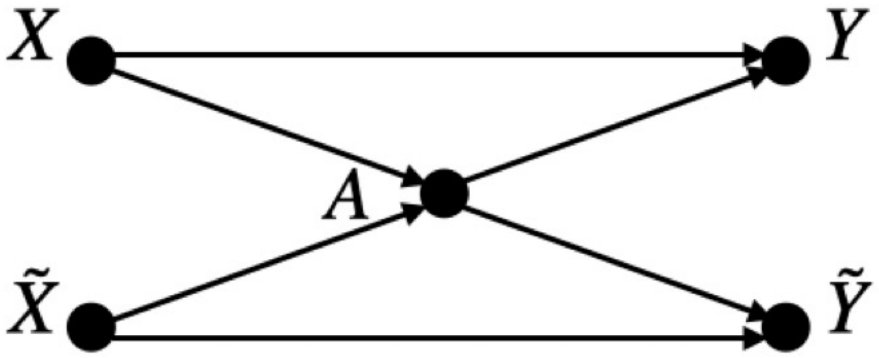
Samy Bengio[1,2]

Emmanuel Abbe[1,2]

[1] Apple

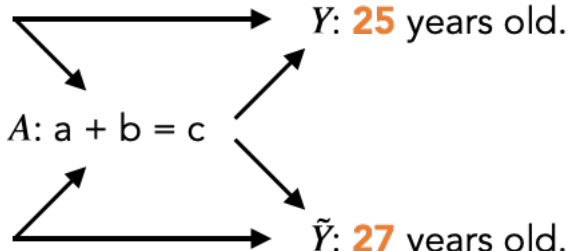
[2] EPFL

Robustness of Reasoning



X: Alice is 20 years old.
Bob is 5 years older than Alice. How old is Bob?

$\tilde{X}$: Tom is 24 years old.
Jerry is 3 years older than Tom. How old is Jerry?



Two problems $\boldsymbol{X}$ and $\tilde{\boldsymbol{X}}$, with solutions $\boldsymbol{Y}$ and $\tilde{\boldsymbol{Y}}$, share the same high-level knowledge or reasoning schema.

Such problems can be handled by a common abstraction $\boldsymbol{A}$

$(\boldsymbol{X}, \boldsymbol{Y}) \rightarrow (\tilde{\boldsymbol{X}}, \tilde{\boldsymbol{Y}})$ illustrates a distribution shift

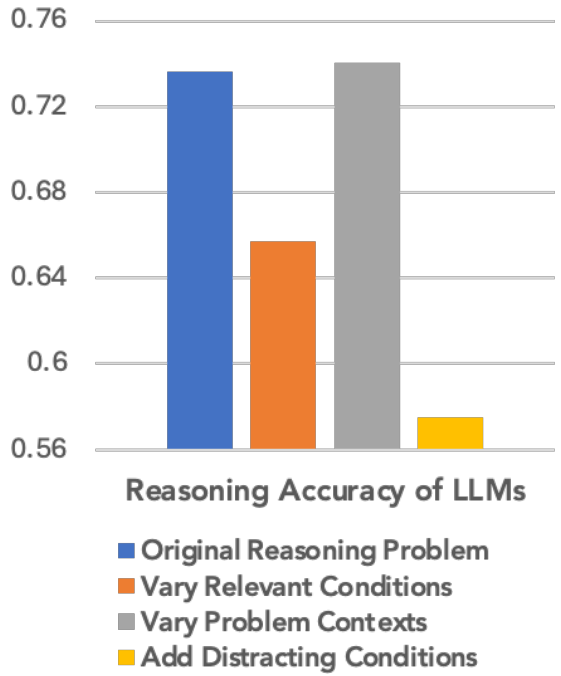
We expect a robust LLM reasoner to understand the underlying abstraction $\boldsymbol{A}$ (abstract thinking), and therefore achieve: $p(\boldsymbol{Y}|\boldsymbol{X}) \approx p(\tilde{\boldsymbol{Y}}|\tilde{\boldsymbol{X}})$

LLMs are Poor at Generalizing to Distribution Shifts

Instantiation Shifts: paraphrase, varying contexts and/or relevant conditions, *etc.*

Interferential Shifts: adding distracting (topic-relevant but useless) conditions

GSM-Plus[1]: GSM8K Problems + Distribution Shifts



[1]Li et al., 2024. "GSM-Plus: A Comprehensive Benchmark for Evaluating the Robustness of LLMs as Mathematical Problem Solvers."

A Common Strategy: Robustifying by Instantiation

Apple
EPFL

Learning more instances of the reasoning problem to anticipate potential distribution shifts.

	Input	Output
	x: **Li** is **12** years old. **Jung** is **2** years older than **Li**. How old is **Jung**?	y: **Jung** is **12 + 2 = 14** years old.
Synthetic Data Augmentation ⇩		
	x': **Zhou** is **15** years old. **Wu** is **5** years older than **Zhou**. How old is **Wu**?	y': **Wu** is **15 + 5 = 20** years old.
	x'': **Amy** is **37** years old. **Sam** is **19** years older than **Amy**. How old is **Sam**?	y'': **Sam** is **37 + 19 = 56** years old.
	...	...

But Computational Expensive…

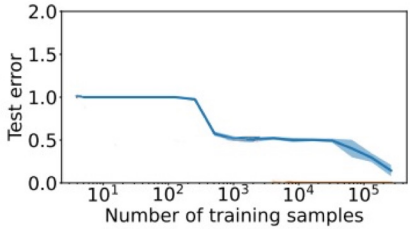
(a) Train data

x_i	y_i
a="d";b="q";print(a)	d
c="r";a="w";print(a)	w
f="y";c="u";print(f)	y
h="o";q="s";print(q)	s
...	...

(b) Test data

x_i^{test}	y_i^{test}
R="F";A="Z";print(R)	F
Q="B";V="A";print(V)	A
...	...

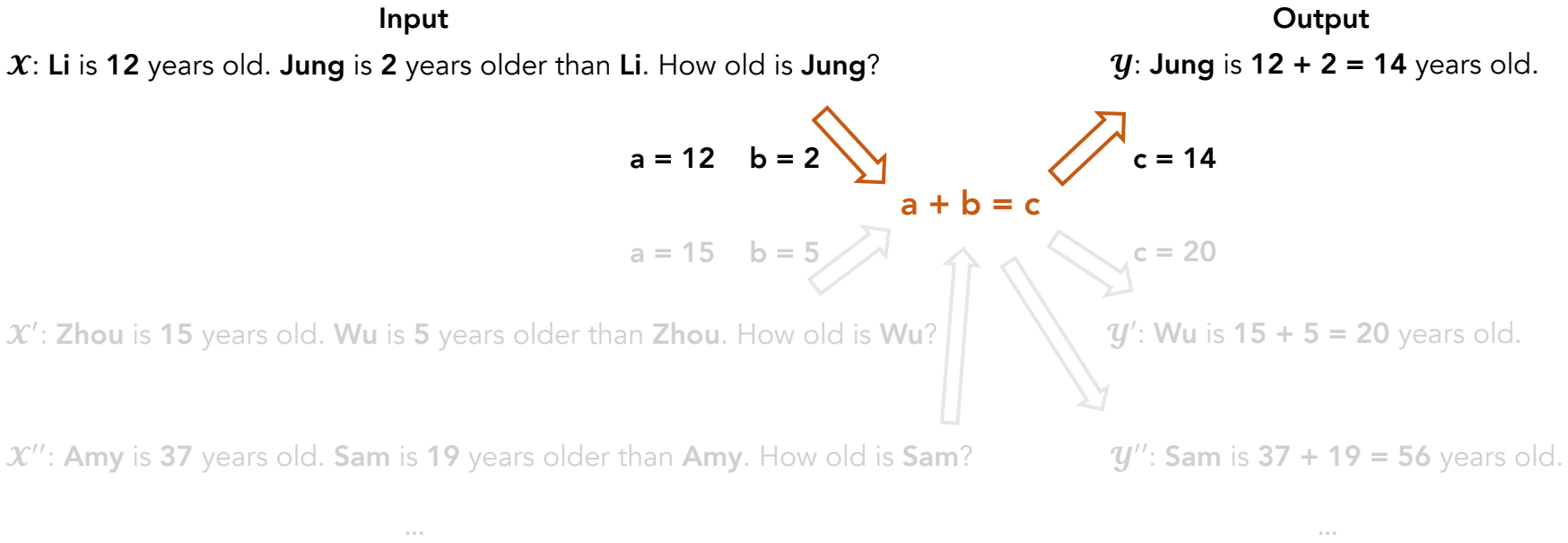
(c) Transformer performance



Even on simple letter printing task, a large amount of training samples were required to mitigate test-time generalization error.[1]

[1]Boix et al., 2024. "When can transformers reason with abstract symbols?"

Our Strategy: Robustifying by Abstraction



We train LLMs to directly learn the abstract thinking!

Modeling more general "abstraction" of reasoning, without scaling up the training data.

AbstRaL Improves Robustness to Distribution Shifts

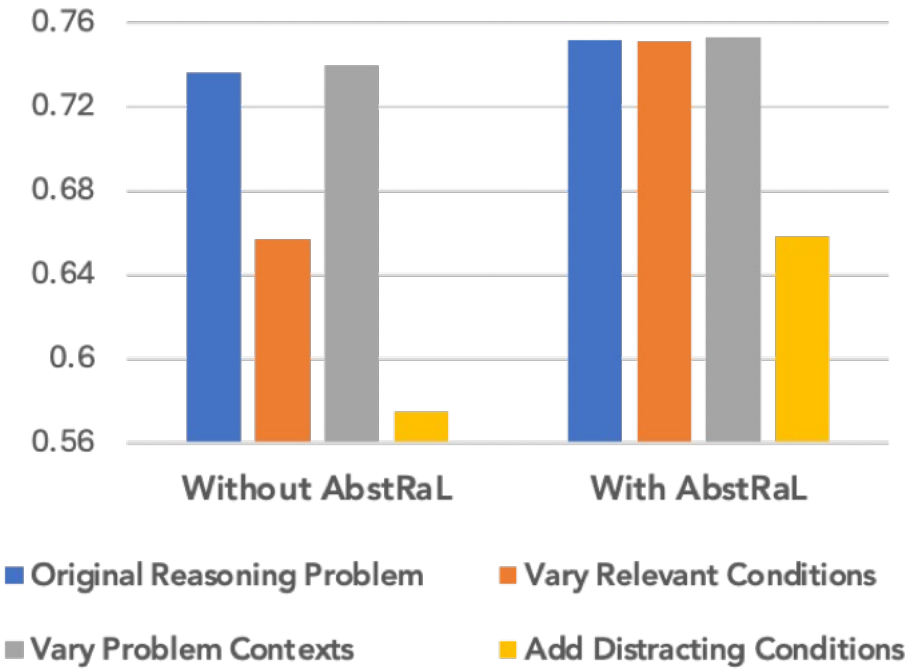
Apple
EPFL

Our Reinforced AbstRaction Learning framework: AbstRaL

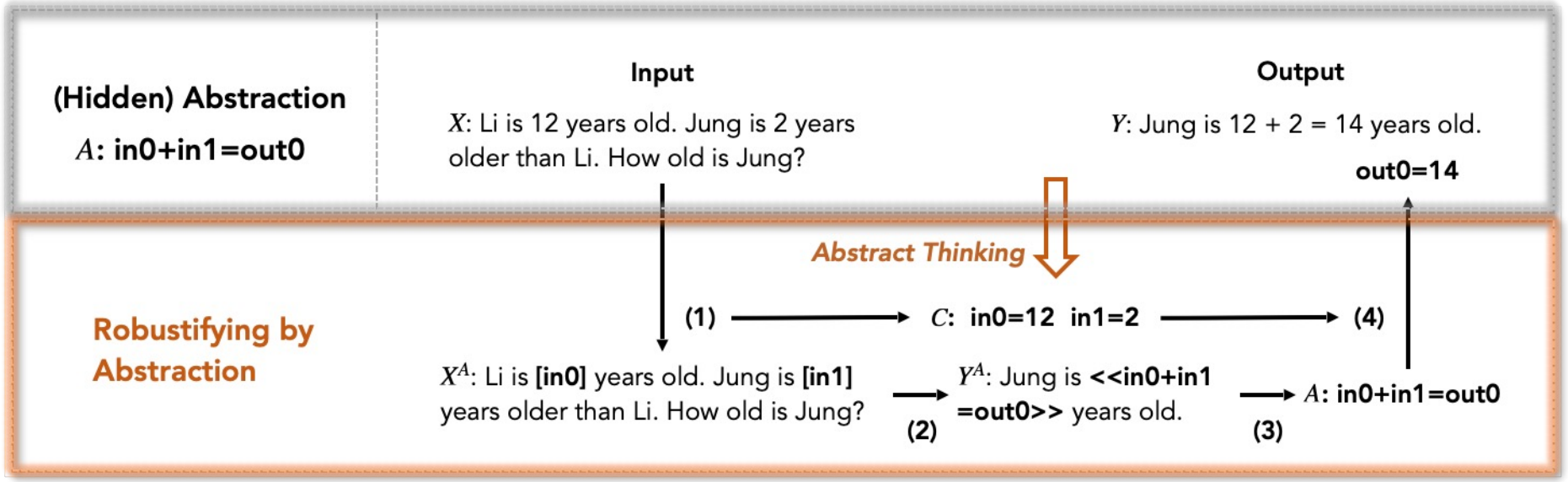
AbstRaL **almost reverts** performance drop caused by varying relevant conditions,

and **significantly mitigates** the interference of distracting conditions.

GSM-Plus: GSM8K Problems + Distribution Shifts



Overview of AbstRaL Framework



(1) Condition Recognition Tool

(2) Abstract Reasoning (Core Step)

(3) Abstraction Retrieval Tool

(4) Symbolic Derivation Tool

LLMs are trained on abstract reasoning with **reinforcement learning**, based on our **granularly-decomposed** abstract reasoning data.

GranulAR Schema

Apple
EPFL

Granularly-decomposed Abstract Reasoning (GranulAR)

- LLMs have learned **fine-grained** reasoning strategies at either pre-training[1] or post-training[2] phase, such as Chain-of-Thought (CoT) and Socratic problem decomposition as representatives.
- GranulAR **integrates abstract reasoning** with these pre-learned beneficial strategies.

X^A: Zhang is **[in0]** times as old as Li. Li is **[in1]** years old. Zhang's brother Jung is **[in2]** years older than Zhang. How old is Jung?

⇩

Y^A **(GranulAR)**:

(Decomposing and Planning) Let's think about the sub-questions we need to answer. **Q1**: How old is Zhang? **Q2**: How old is Jung?

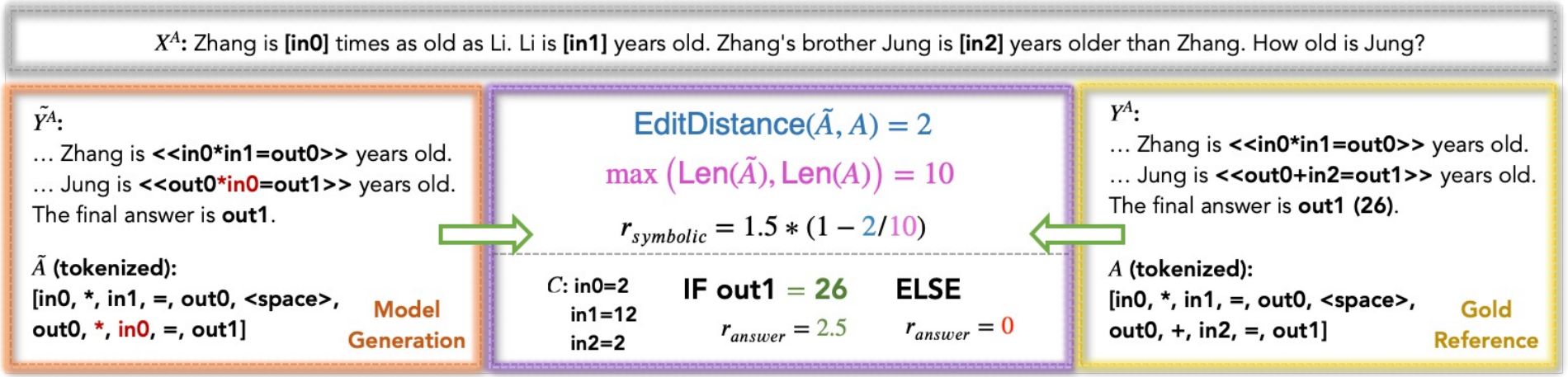
(CoT with Quoting Abstract Symbols) Let's answer each sub-question one by one.
Q1: How old is Zhang? Li is **[in1]** years old, Zhang is **[in0]** times as old as Li, so Zhang is **<<in0*in1=out0>>** years old.
Q2: How old is Jung? Zhang is **[out0]** years old, Jung is **[in2]** years older than Zhang, so Jung is **<<out0+in2=out1>>** years old.

(Conclusion) **The final answer is [out1]**.

[1]Yang et al., 2024. "Do large language models latently perform multi-hop reasoning?"
[2]Kumar et al., 2025. "Llm post-training: A deep dive into reasoning large language models."

RL with Abstraction Rewards

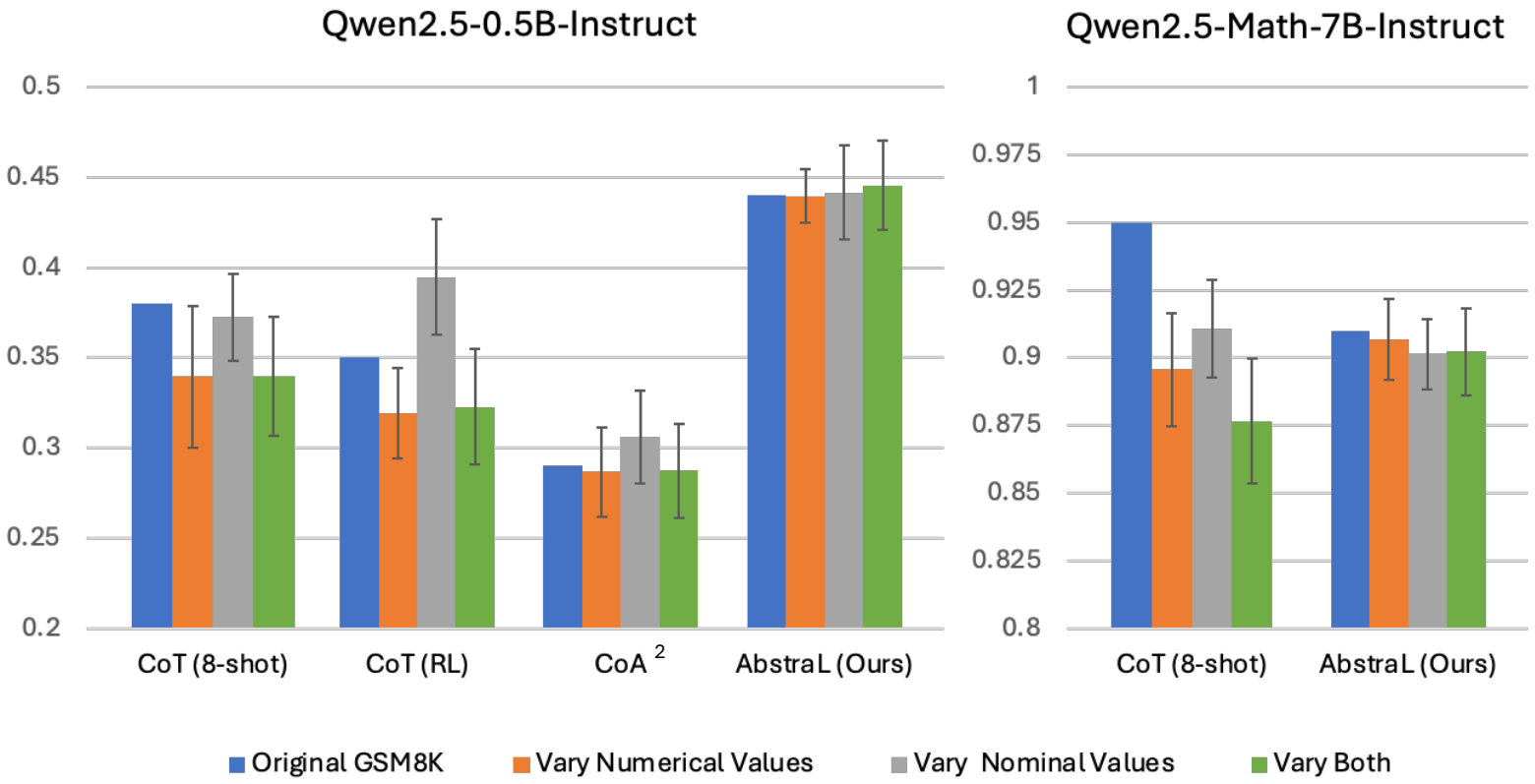
- LLMs are **poor at following in-context** demonstrations to reason in abstract manner[1].
- **Limitation of plain supervised fine-tuning (SFT)**: auto-regressive training objective forces LLMs also modeling the specific contexts.



- **Symbolic Distance Reward ($r_{symbolic}$)** granularly measures how the generated abstraction is aligned with (or close to) the expected abstraction, serving as a **milestone-style** reward that **more closely monitors** the progress of learning.
- **Answer Correctness Reward (r_{answer})** checks whether the generated abstraction can derive the **correct final answer** given the gold input conditions.

[1]Gao et al., 2025. "Efficient Tool Use with Chain-of-Abstraction Reasoning."

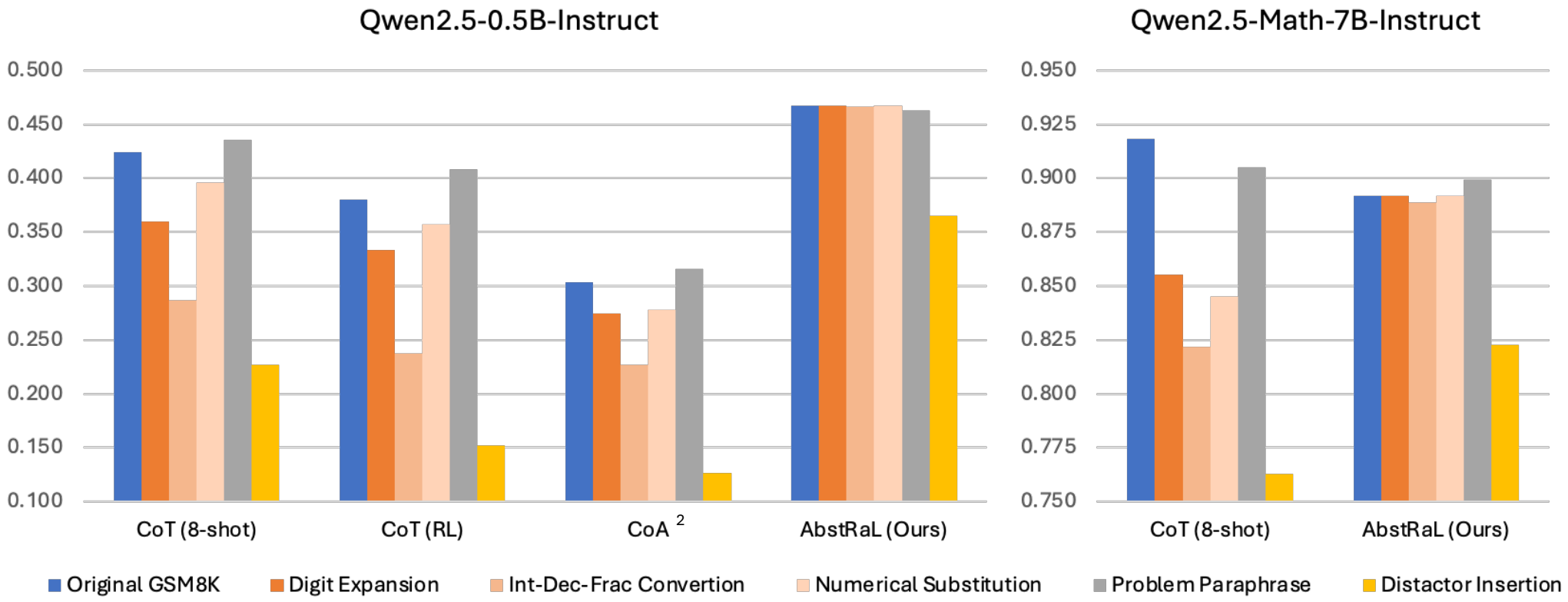
Results on GSM-Symbolic[1]



- Compared to baseline learning schemes (CoT with RL and CoA), AbstRaL is **more reliable** to augment reasoning.
- AbstRaL may **mitigate LLMs' overfitting** to the existing input conditions (or numbers), caused by potential **data contamination** at the pre-training or post-training stage.

[1]Mirzadeh et al., 2024. "GSM-Symbolic: Understanding the Limitations of Mathematical Reasoning in Large Language Models."
[2]Gao et al., 2025. "Efficient Tool Use with Chain-of-Abstraction Reasoning."

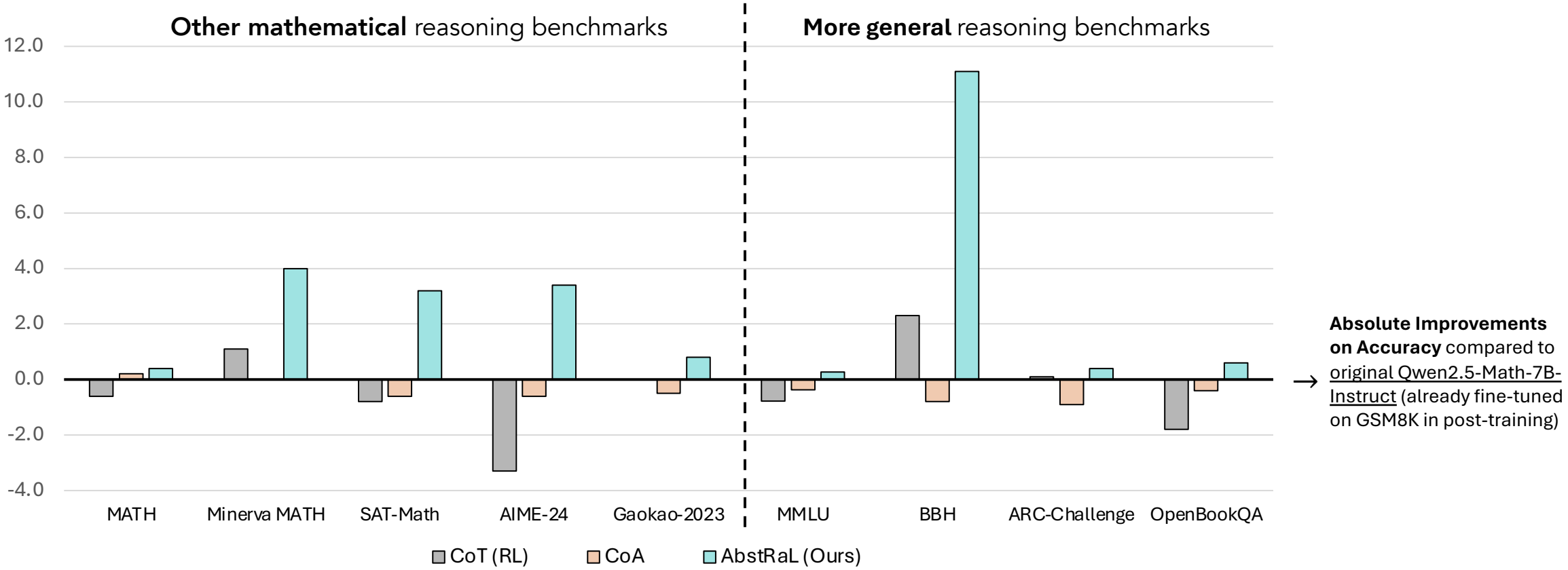
Results on GSM-Plus[1]



- AbstraL **almost reverts** the performance degradation caused by variations of input numbers (**orange**).
- AbstraL **mitigates the interference** of distracting conditions (**yellow**).

[1]Li et al., 2024. "GSM-Plus: A Comprehensive Benchmark for Evaluating the Robustness of LLMs as Mathematical Problem Solvers."
[2]Gao et al., 2025. "Efficient Tool Use with Chain-of-Abstraction Reasoning."

Generalization to OOD Tasks



→ **Absolute Improvements on Accuracy** compared to original Qwen2.5-Math-7B-Instruct (already fine-tuned on GSM8K in post-training)

- AbstRaL **consistently outperforms** baseline methods, even though the learned GSM reasoning scheme may not be directly applicable to solving the OOD tasks.
- Learning the abstract thinking underlying GSM can already **implicitly benefit** other mathematical reasoning and more general reasoning of LLMs.

Conclusion

Apple
EPFL

- LLMs, particularly **small ones**, were shown to be **non-robust** at reasoning, representatively on the task of grade school mathematics (GSM)
  → **AbstRaL** effectively **offsets** the performance drop caused by distribution shifts!

- Our **RL with abstraction rewards** and **GranulAR** reasoning schema appear to be the key to reasoning robustness; while other baseline methods (CoT with RL, CoA, *etc.*) often fail.

- Improving GSM robustness via AbstRaL also implicitly benefits LLMs' capabilities on OOD mathematical and general reasoning tasks, indicating that **abstract thinking broadly enables better generalizability**.