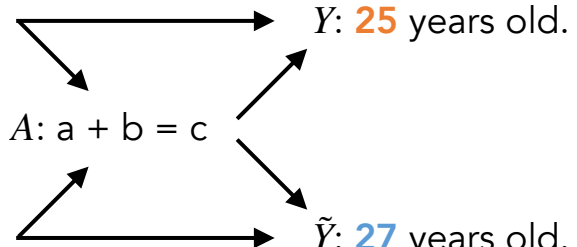
AbstRaL: Augmenting LLM's Reasoning by Reinforcing Abstract Thinking

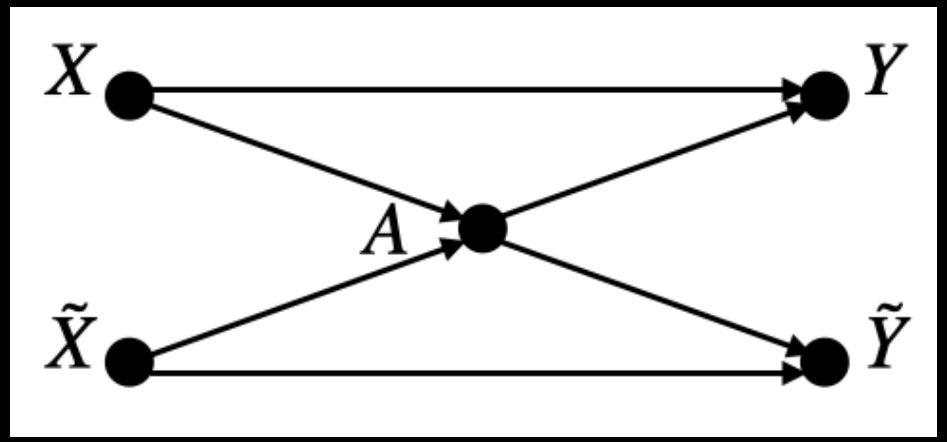
Silin Gao Antoine Bosselut Samy Bengio Emmanuel Abbe

Are LLM truly reasoning? E.g., abstracting?

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

$\tilde{X}$: **Aryo** is **24** years old.
Amin is **3** years older than **Aryo**. How old is **Amin**?





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

We would hope that a robust LLM reasoner can understand the underlying abstraction A **(abstract thinking)**, and therefore achieve: $p(Y \mid X) \approx p(\tilde{Y} \mid \tilde{X})$

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

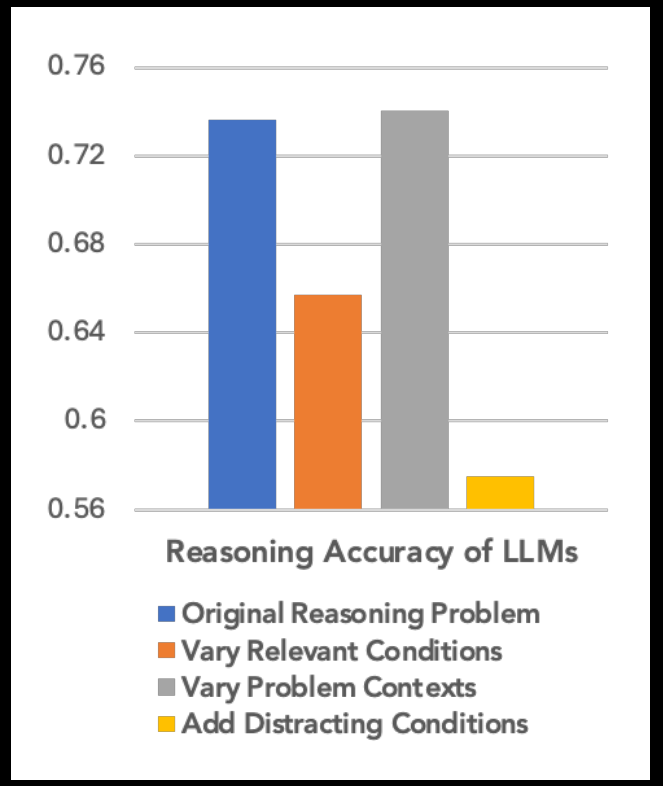
Can problems be handled by the LLM with a common abstraction A?

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

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

Input	Output
X: **Bob** is **12** years old. **Alice** is **2** years older than **Bob**. How old is **Alice**?	Y: **Alice** 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'': **Aryo** is **37** years old. **Amin** is **19** years older than **Aryo**. How old is **Amin**?	Y'': **Amin** is **37 + 19 = 56** years old.
…	…

But Computational Expensive…

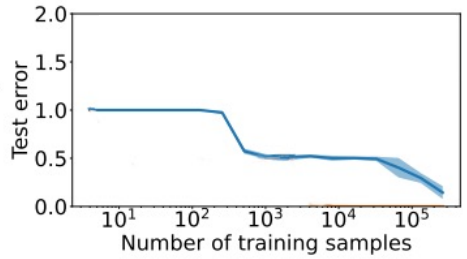
(a) Train data

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



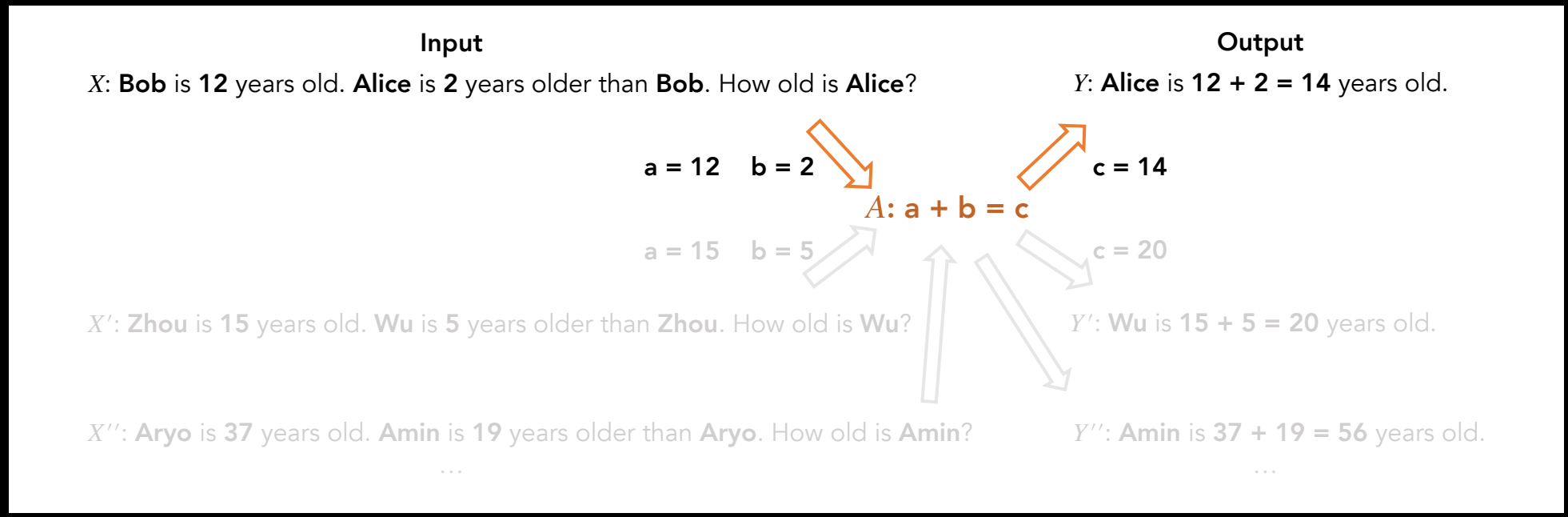
Even on simple token printing task, a large amount of training samples are required to mitigate test-time generalization error in from scratch training.[1]

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

Ayn Rand

"**Abstract ideas** are conceptual integrations which subsume an incalculable number of concretes—and without abstract ideas you would not be able to deal with concrete, particular, real-life problems. You would be in the position of a new born infant."

Our Strategy: Robustifying by Abstraction



We train LLMs to directly learn the abstract thinking!

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

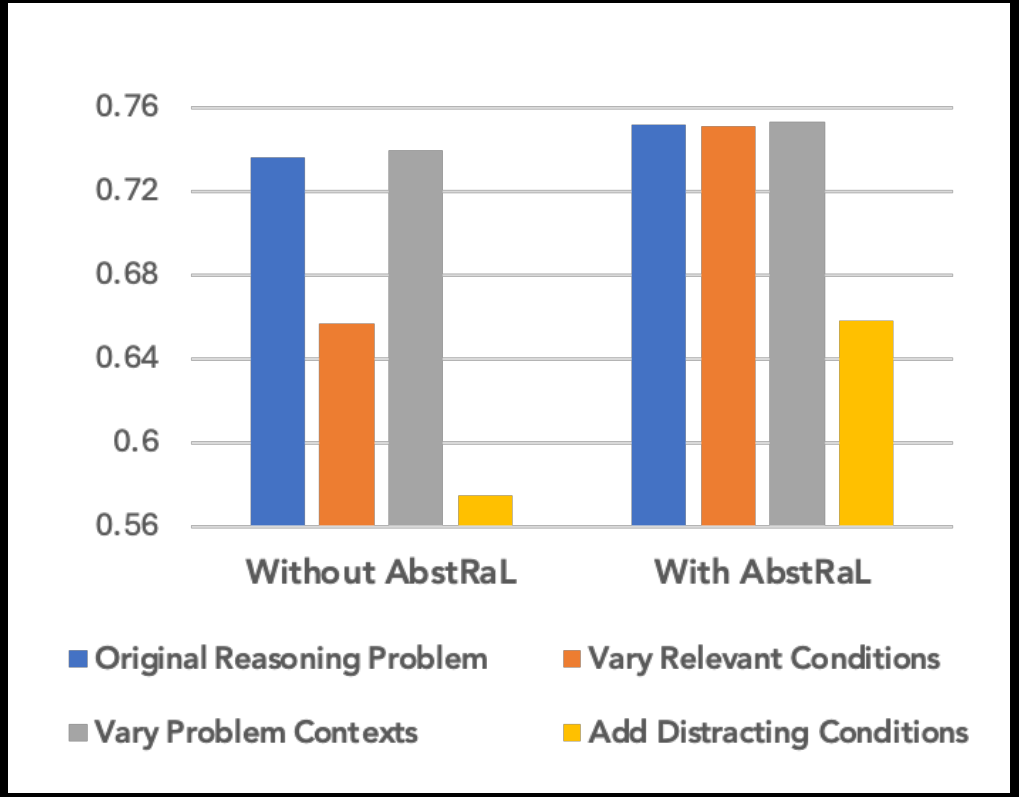
AbstRaL Improves Robustness to Distribution Shifts

Our Reinforced AbstRaction Learning framework: AbstRaL

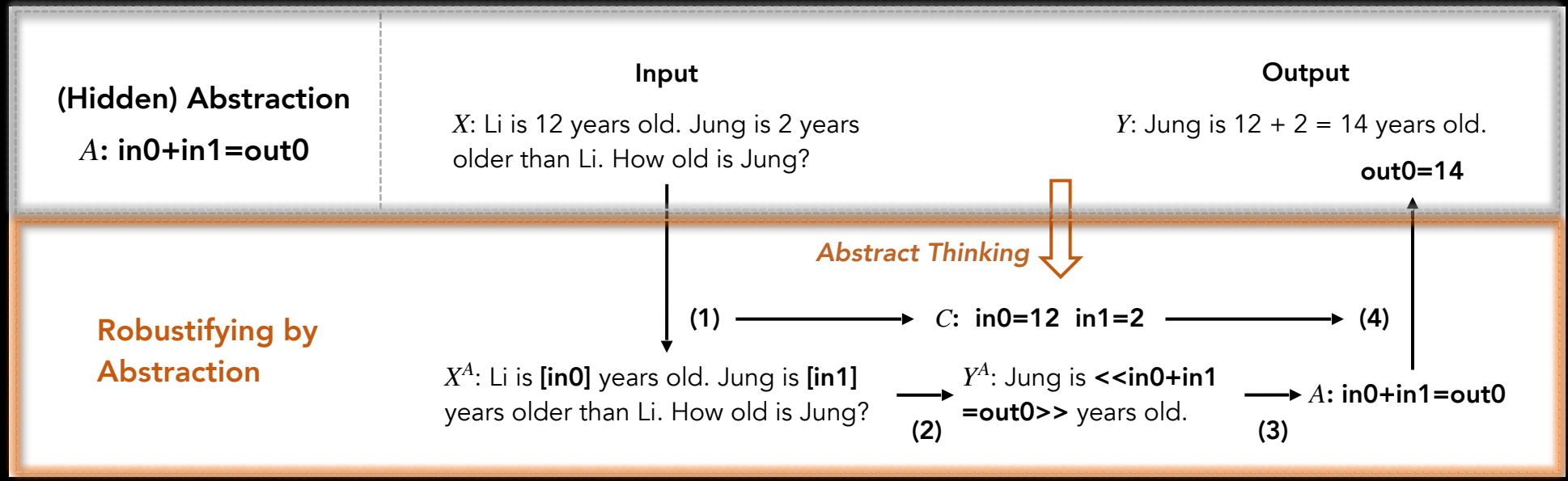
AbstRaL **almost reverts** performance drop caused by varying relevant conditions,

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

GSM-Plus: GSM8K Problems + Distribution Shifts



Overview of AbstRaL Framework



(1) Condition Recognition Tool

(4) Symbolic Derivation Tool

(2) Abstract Reasoning (Core Step)

(3) Abstraction Retrieval Tool

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

GranulAR Schema

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?

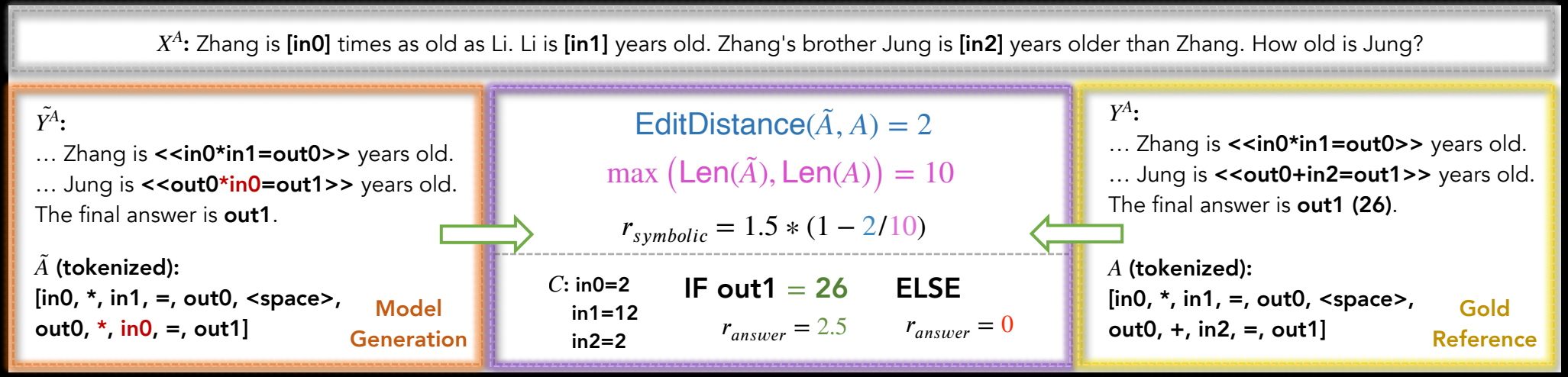
(CoT with Quoting Abstract Symbols) Let's answer each sub-question one by one.
Q1: How old is Zhang? Li is **[in1]** years old, Zhang is **[in0]** times as old as Li, so Zhang is **<<in0*in1=out0>>** years old.
Q2: How old is Jung? Zhang is **[out0]** years old, Jung is **[in2]** years older than Zhang,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

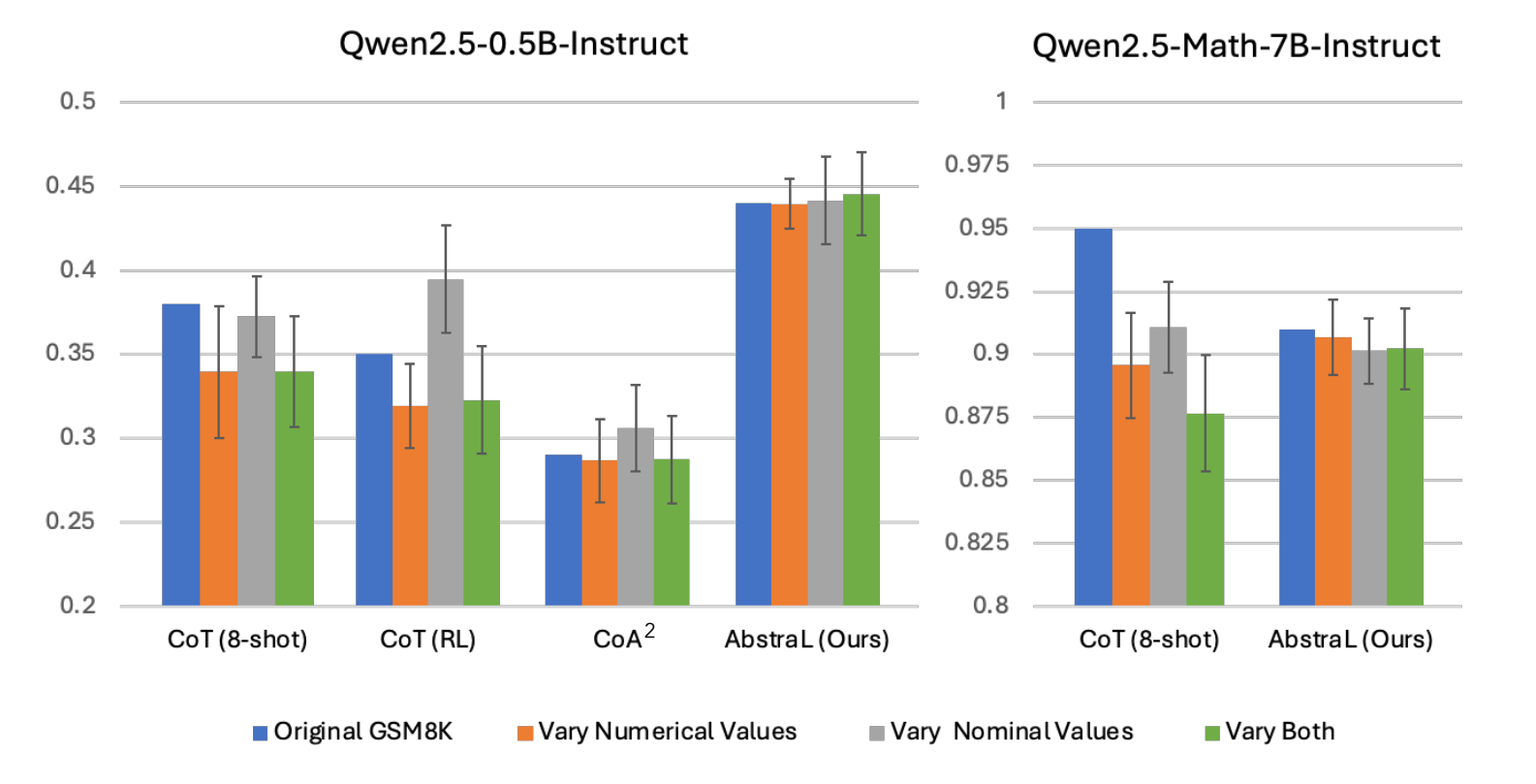
- LLMs are **poor at following in-context** demonstrations to reason in abstract manner[1].
- **Limitation of plain supervised fine-tuning (SFT)**: auto-regressive training objective forces LLMs also modeling the specific contexts.



- **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."

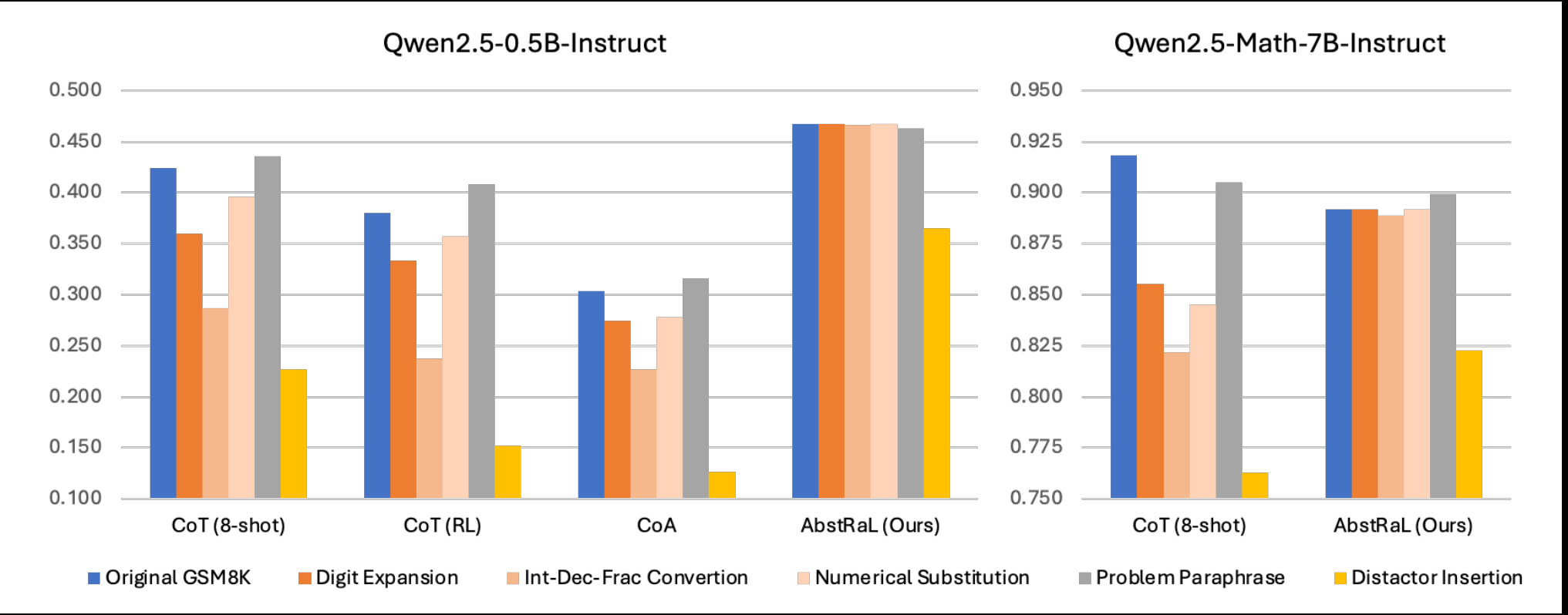
Results on Grade School Mathematics (GSM-Symbolic[1])



- 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 Grade School Mathematics (GSM-Plus)



- 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.

OOD performance (trained with GSM and tested beyond GSM)

Table 3: Zero-shot evaluation results on OOD datasets. Best results on each model are **bold**.

Model	GSM Train Method	MATH	Minerva MATH	SAT-Math	AIME24	MMLU				BBH
						STEM	Social	Humanities	Other	
Qwen2.5-0.5B-Instruct	Ori-SFT	30.0	5.5	46.9	0.0	37.5	51.5	41.6	49.7	21.0
	CoT-RL	19.2	5.5	43.8	0.0	38.5	50.8	41.2	49.1	23.2
	CoA	31.6	5.5	43.8	0.0	38.5	50.8	41.9	49.3	21.8
	AbstRaL	**34.7**	**7.4**	**62.5**	0.0	**38.8**	**53.5**	**42.2**	**50.7**	**26.3**
Qwen2.5-Math-7B-Instruct	Ori-SFT	83.5	34.6	90.6	13.3	68.4	82.5	63.2	76.4	43.7
	CoT-RL	82.9	35.7	89.8	10.0	68.5	81.5	63.0	74.4	46.0
	CoA	83.7	34.6	90.0	12.7	68.5	81.7	63.3	75.5	42.9
	AbstRaL	**83.9**	**38.6**	**93.8**	**16.7**	**68.6**	**82.7**	**63.8**	**76.5**	**54.8**

Conclusion

- LLMs, particularly **small ones**, were shown to be **non-robust** at reasoning, such as on **Apple's GSM-Symbolic**
  → **AbstRaL** allows to essentially **offset** 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.

- **Next:** expand our AbstRaL framework in a **domain adaptive** manner, to augment more **general purpose** LLMs.

More general conclusion

LLMs can have difficulties solving challenging reasoning tasks
on their own (barrier phenomena).

Rather than teaching LLMs how to produce good solutions directly,
teach LLMs how to produce good explanations.

This goes beyond just training on thinking traces,
it calls upon principles of reasoning (induction, abstraction and more).