Program-of-Thought Reveals LLM Abstraction Ceilings

Mike Zhou^1 **Fenil Bardoliya**^{2} **Vivek Gupta**^{2} **Dan Roth**^{1}
^{1}University of Pennsylvania ^{2}Arizona State University
{mikezhou,danroth}@seas.upenn.edu {fbardoli,vgupt140}@asu.edu

Abstract

Large language models (LLMs) are often claimed to exhibit reasoning ability when supervised with chain-of-thought (CoT) traces. True reasoning, however, requires invariance: isomorphic problems should yield identical solutions regardless of superficial variation. We test this property by evaluating base and reasoning-optimized models—including LLaMA, Mistral, Qwen, GPT-OSS, and Deepseek—on isomorphic variants from GSM8K and MATH. All models exhibit substantial accuracy drops under perturbation. To assess whether training can induce invariance, we fine-tune models with Program-of-Thought (PoT) supervision under concrete and masked formulations. PoT fine-tuning increases behavioral cross-variant consistency but does not significantly reduce the accuracy gap, and these gains fail to transfer across prompting formats and domains. Our central finding is that models converge toward stable but systematically incorrect behaviors: *consistency without correctness*. This dissociation suggests that current reasoning supervision teaches models to reproduce solution templates rather than to abstract mathematical structure.

1 Introduction

Many mathematical word problems admit multiple surface forms that are logically *isomorphic*: the wording, numeric instantiation, or order of presentation may change, yet the underlying computational structure remains invariant. A system that truly abstracts reasoning should solve all members of an isomorphic family with equal reliability. In program-mediated reasoning, this means the solution procedure should be identical, aside from variable naming, regardless of surface perturbations.

Large language models (LLMs) achieve strong performance on mathematical benchmarks like GSM8K and MATH (Cobbe et al., 2021; Hendrycks et al., 2021), particularly with Chain-of-Thought (CoT) prompting (Wei et al., 2022). However, this performance degrades sharply under numeric perturbations (Mirzadeh et al., 2025; Yu et al., 2024). These failures remain ambiguous: when models produce different CoT traces for isomorphic problems, we cannot tell, for example, whether they miss structural equivalence or make arithmetic errors. Program-of-Thought (PoT) prompting (Chen et al., 2023) resolves this ambiguity, as models should generate identical programs for isomorphic variants, making divergence a clear failure of abstraction rather than computation.

To operationalize[1] this separation, we employ PoT prompting on perturbation datasets like ReasonAgain (Yu et al., 2024), which pairs isomorphic variants with canonical programs. We ask: *can targeted training induce the invariances that genuine reasoning requires?* We find that it cannot. PoT fine-tuning reduces variance across problem classes without significantly reducing accuracy drops. This dissociation between consistency and correctness is our primary contribution.

We first demonstrate that, under an all-variants accuracy (AVA) metric, current models suffer sharp performance drops (Δ_{AVA}). We then test whether PoT-based fine-tuning can reduce this brittleness. We train models under two regimes: (*i*) **Concrete**, using natural language paired with executable programs, and (*ii*) **Masked**, replacing numbers and entities with typed variables. PoT supervision increases consistency across isomorphic variants, yet the perturbation gap (Δ_{AVA}) remains largely unchanged. These gains also fail to transfer across prompting formats (PoT to CoT) or domains (GSM8K to MATH). Under stricter analysis, PoT training consolidates errors rather than correcting them—models learn to fail consistently, demonstrating that structured supervision can induce procedural stability without mathematical abstraction.

[1] code is available at: https://cogcomp.seas.upenn.edu/page/publication_view/1098

Findings of the Association for Computational Linguistics: EACL 2026, pages 4911–4919
March 24-29, 2026 ©2026 Association for Computational Linguistics

2 Related Work

Reasoning Decomposition and Program-Based Diagnostics. Prompting methods such as CoT, self-consistency, least-to-most decomposition, and PoT improve performance on mathematical reasoning benchmarks by encouraging intermediate structure (Wei et al., 2022; Wang et al., 2023; Zhou et al., 2023; Chen et al., 2023). PoT differs by representing reasoning as executable programs, separating symbolic structure from arithmetic execution. Program-Aided Language Models have been shown to improve accuracy using Python programs as intermediate solutions (Gao et al., 2023).

Robustness Under Distribution Shift. Diagnostic challenge sets suggest that strong benchmark performance may however rely on shallow heuristics; GSM-Symbolic and ReasonAgain evaluate robustness via controlled numerical and symbolic perturbations (Mirzadeh et al., 2025; Yu et al., 2024), observing substantial performance drops under CoT despite strong in-distribution accuracy. However, their evaluation conflates multiple failure modes, like arithmetic mistakes and linguistic brittleness. Recent work (Dziri et al., 2023; Shojaee et al., 2025) also reveals sharp performance degradation in LLM reasoning by studying synthetic puzzle-like tasks with controlled compositional structure. While they demonstrate brittleness under carefully constructed tasks, they do not isolate failures of abstract reasoning from failures of execution or multi-step state tracking.

Present Work. In contrast to prior work, we isolate reasoning failures from execution/computational errors by leveraging PoT's executable structure. PoT renders model reasoning as an explicit executable program, enabling failures attributable to abstract reasoning to be examined separately from other confounding sources, such as computational/execution errors, linguistic variation, or state-tracking limitations. The framework considers isomorphic problem families that differ only in numeric instantiations while requiring the same abstract computation. Robustness is characterized by whether these perturbations induce failures despite preserving the required reasoning. This comparative evaluation exposes a failure mode overlooked by prior work: models may achieve high accuracy on individual instances while remaining brittle to perturbations that should not affect the correctness of the underlying output.

3 Experimental Setup

Data and Perturbations. We evaluate all models on GSM8K and MATH (Cobbe et al., 2021; Hendrycks et al., 2021) using ReasonAgain's (Yu et al., 2024) perturbations that alter surface numbers while preserving program structure: 6,701 perturbed + 1,121 original GSM8K items; 1,513 perturbed + 268 original MATH items.

Models. We evaluate LLaMA-3.1-8B-Instruct, Mistral-7B-Instruct, Llama-4 Maverick-17b-128e-Instruct (Grattafiori et al., 2024; Jiang et al., 2023; Meta AI, 2025), and 3 reasoning models: Qwen-3-8B-Instruct, GPT-OSS-120B, and Deepseek-V3.2-Exp (Yang et al., 2025; Agarwal et al., 2025; DeepSeek-AI et al., 2025). LLaMA-3.1-8B and Mistral-7B are additionally fine-tuned.

Supervision Regimes. We fine-tune with LoRA (Hu et al., 2021) on GSM8K Python traces with two regimes: *(i) PoT–Concrete*: original problems with gold programs; *(ii) PoT–Masked*: names/numbers replaced by typed variables (e.g., `alice_books:int`). Settings: 3 epochs, rank-16, $\alpha = 32$, LR = 10^{-5}, effective batch = 64.

Prompting Modes. All base models are evaluated zero-shot with **CoT** (Wei et al., 2022), **CoT+Self-Consistency (SC)** (Wang et al., 2023), and **PoT** (Chen et al., 2023). Non-reasoning models are also evaluated with **Direct** mode, while reasoning models default to CoT. For fine-tuned models we restrict evaluation to **PoT** and **CoT**. Code in all PoT settings is extracted using lightweight `<python>` tags with sandboxed execution. All PoT prompts include a brief one-shot example that specifies the correct extraction template.

Decoding. Fine-tuned models use greedy decoding (temp = 0) for deterministic stability metrics (shared/new/recovered errors). Base models follow standard practice: 5 samples (temp = 0.7, top-p = 0.9) with accuracy averaged, except SC uses 10 samples with majority voting. We use a maximum generation length of 512 tokens for non-reasoning models and 2048 tokens for reasoning models.

Metrics and Normalization. We report exact-match accuracy (after deterministic answer extraction) and stability metrics over original–perturbed families of isomorphic variants. We define an *all-variants accuracy (AVA)* that measures mean accuracy across all perturbed variants (AVA $= \frac{1}{N}\sum_{i=1}^{N} a_i,\ a_i \in \{0,1\}$) and a distance Δ_{AVA}

(a) Non-Reasoning Models

	GSM8k			MATH		
Prompt	**Orig.**	**AVA**	Δ_{AVA}	**Orig.**	**AVA**	Δ_{AVA}
Llama-3.1-8B						
Direct	82.60	62.54	20.06	77.23	47.72	29.51
CoT	91.92	72.50	19.47	84.78	52.54	32.24
CoT+SC	91.97	72.47	19.50	85.07	52.54	32.53
PoT	88.93	75.72	**13.21**	81.94	59.35	**22.59**
Mistral-7B						
Direct	51.76	39.30	12.46	25.76	19.37	**6.39**
CoT	61.91	44.59	17.32	32.91	20.51	12.40
CoT+SC	61.91	44.61	17.30	32.84	20.51	12.33
PoT	57.57	49.19	**8.38**	31.34	24.42	6.92
Llama-4-Maverick-17B-128e						
Direct	98.93	80.98	17.95	99.78	71.75	28.03
CoT	99.03	81.12	17.88	99.93	71.49	28.44
CoT+SC	99.16	82.95	16.21	100.00	71.70	28.30
PoT	97.04	83.53	**13.51**	94.40	70.75	**23.65**

(b) Reasoning Models

	GSM8k			MATH		
Prompt	**Orig.**	**AVA**	Δ_{AVA}	**Orig.**	**AVA**	Δ_{AVA}
Qwen-3-8B						
CoT	93.67	69.02	24.65	89.78	45.15	44.63
CoT+SC	94.02	69.53	24.49	90.30	45.47	44.83
PoT	69.99	55.91	**14.08**	47.46	36.61	**10.85**
GPT-OSS-120B						
CoT	97.29	72.56	24.73	99.93	71.76	28.17
CoT+SC	97.50	72.98	24.52	100.00	72.18	**27.82**
PoT	94.59	80.10	**14.49**	93.66	64.29	29.37
DeepSeek-V3.2-Exp						
CoT	97.00	73.83	23.17	99.55	67.63	31.92
CoT+SC	97.77	73.86	23.91	100.00	69.47	30.53
PoT	96.81	77.04	**19.77**	96.34	66.53	**29.81**

Table 1: Model accuracy on GSM8k dataset under PoT prompting after Fine Tuning. Orig. = accuracy on original questions; AVA = all-variants accuracy; Δ_{AVA} = accuracy drop. Accuracies are given as percentages.

which denotes the original-AVA accuracy gap, thereby measuring robustness. We decompose outcomes into: *new errors* (original correct, any perturbation wrong), *shared errors* (original wrong, any perturbed wrong), and *recovered errors* (original wrong, any perturbation correct).

4 Results and Analysis

4.1 Aggregate Error Report for Base Models

Across all baselines, the gap between per-instance and all-variants accuracy Δ_{AVA} is substantial, highlighting widespread instability under surface perturbations. On GSM8K, non-reasoning models lose roughly 16–19%, while reasoning models drop by 18–24% under CoT evaluation—indicating that explicit reasoning traces do not necessarily confer greater robustness. The gap widens further on the MATH dataset, where Δ_{AVA} often exceeds 20%. Although prompting on PoT slightly reduces Δ_{AVA} in all but two cases, the gap remains sizable, suggesting that PoT prompting improves robustness but that current models still lack the deeper generalization required for stable reasoning.

Among programs that executed successfully, we identify two dominant error categories: logical errors, arising from skipped steps and faulty intermediate reasoning or control flow; and rounding or discrete-threshold errors (e.g., "minimum number of books"). In both cases, the generated code is syntactically valid and executes without error, yet produces wrong answers. Overall, these results suggest that PoT prompting improves robustness but current models remain brittle under perturbation.

4.2 Fine-tuning Results

The base model results raise a natural question: can targeted training induce the invariance that genuine reasoning requires? We investigate this by fine-tuning on PoT traces under two regimes (Concrete and Masked) and evaluate along three dimensions: whether consistency improves within distribution, whether gains transfer across tasks/prompting formats, and whether consistency brings correctness. Our findings reveal a dissociation. PoT supervision increases behavioral consistency, causing models to succeed and fail more uniformly across isomorphic variants, but does not significantly reduce the accuracy gap. The gains are representation-bound and do not transfer to Chain-of-Thought evaluation or to out-of-domain problems. Most notably, models learn to fail the same way.

Operational-level Semantics. At the operator frontier, even after fine-tuning on executable traces that explicitly contain rounding (e.g., minimum number of books = ceil) and fencepost logic, models still mishandle these discrete thresholds at test time. Roughly 6–9% of GSM8K is rounding-sensitive and continues to fail (Tab. 2), indicating that PoT training imparts format more readily than calibrated semantics. We therefore report both *Non-Rounded* and *Auto-Round* accuracies, and report error classification (as defined in metrics in § 3) on responses within the Auto-Round category (though results in the Non-Rounded setting can be found in Appendix A). On MATH, Auto-Round yields negligible gains, likely because many problems

	Non-Rounded			Auto-Round	
FineTune	**Orig.**	**AVA**	**Δ_{AVA}**	**AVA**	**Round Err.**
GSM8k					
Llama-3.1-8B					
Masked	88.22	74.57	**13.65**	83.05	8.48
Concrete	88.22	74.55	13.67	83.01	8.46
Base	83.59	69.15	14.44	76.03	6.88
Mistral-7B					
Masked	60.93	53.69	**7.23**	59.76	6.07
Concrete	61.02	53.66	7.36	59.73	6.07
Base	58.16	49.64	8.52	55.30	5.66
MATH					
Llama-3.1-8B					
Masked	81.34	59.86	21.48	60.25	0.39
Concrete	79.85	60.57	**19.28**	61.05	0.48
Base	82.46	61.61	20.85	62.40	0.79
Mistral-7B					
Masked	27.24	22.50	4.74	23.45	0.95
Concrete	26.12	23.21	**2.91**	24.40	1.19
Base	29.10	25.68	3.43	27.34	1.66

Table 2: Model accuracy under PoT prompting after fine-tuning (greedy decoding). Orig. = accuracy on original questions; AVA = all-variants accuracy; Δ_{AVA} = accuracy drop; Round Err. = Auto-Round AVA − Non-Rounded AVA.

naturally admit non-integer outputs (e.g., rates or averages), so enforcing rounding does not systematically correct errors.

PoT Training Induces In-Distribution Behavioral Consistency. Across models, PoT supervision primarily stabilizes behavior rather than improve Δ_{AVA} (Tab. 2), as reflected in substantial reductions in both new and recovered errors (Tab. 3). For LLAMA-3.1-8B, new errors decrease from 431 to 212–213 and recovered errors from 167 to 87 (Masked/Concrete), with MISTRAL-7B exhibiting a similar reduction in both categories. The same pattern holds under non-auto-rounded inference (see Appendix A). Under a stricter error classification, PoT training does not randomly reduce error counts, but instead concentrates failures onto subsets of problems across equivalent perturbations, yielding greater uniformity in failure patterns without improved correctness, explaining why consistency gains do not translate to improved Δ_{AVA} (see Appendix A). For example, a model that originally answered incorrectly but recovered on some perturbations may, after training, answer all variants incorrectly, suppressing recovered errors while lowering AVA. Manual inspection of 20 consistent failures reveals that fine-tuning consolidates model behavior. In some cases, models converge to a single error repeated across all variants. In others, models converge to a small set of distinct error patterns. In both cases, consistency increases without accuracy drops significantly improving.

SFT Improves Execution but Not Reasoning Invariance. We fine-tune on GSM8K and evaluate on MATH to test cross-task transfer. In-domain consistency improves substantially, but cross-task Δ_{AVA} and consistency remain largely unchanged, with MATH accuracy slightly decreasing ($\sim 2\%$).

Invariance metrics show no clear improvement outside the training domain—new and recovered errors sometimes increase. Normalized consistency metrics vary only modestly (Fig. 1). More consistent gains across these measures would be indicative of broadly reusable reasoning abstractions; instead, the observed improvements appear largely confined to domain-specific policies.

The small cross-task accuracy drop reflects expected task-shift effects rather than overfitting: AVA and consistency remain stable under perturbation, indicating preserved reasoning behavior.

CoT vs PoT Evaluation and Masked Training. The gains in consistency are also channel-bound. When evaluated under Chain-of-Thought prompting, PoT-tuned models show no systematic improvement (Fig. 1), and MISTRAL-7B shows inconsistent results (Tab. 3). This pattern reveals that the competence induced by PoT is *representation-specific*. PoT improves procedural fluency, the ability to emit stable, program-shaped solutions for isomorphic inputs, but that fluency does not transfer to form-agnostic understanding that would transfer

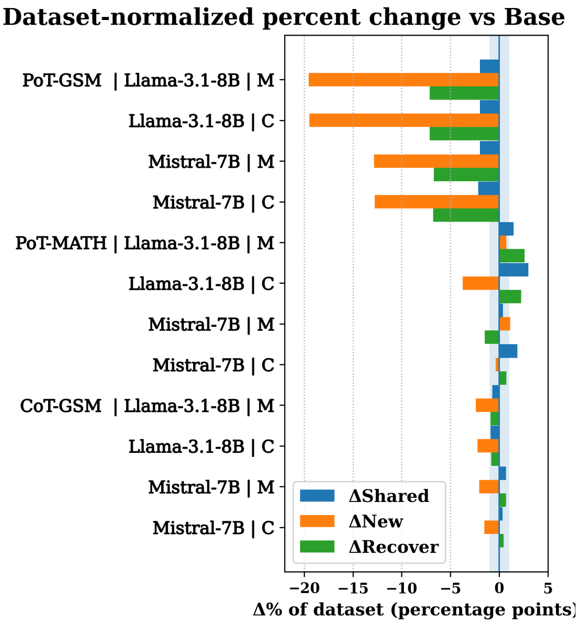

Figure 1: Dataset-normalized percent changes in error categories relative to base model. M = Masked, C = Concrete.

	Program-of-Thought (GSM8K)			Program-of-Thought (MATH)			Chain-of-Thought (GSM8K)		
FineTune	Shared	New	Recover	Shared	New	Recover	Shared	New	Recover
Llama-3.1-8B-Instruct									
Masked	115	212	87	42	141	28	99	584	90
Concrete	115	213	87	46	129	27	97	586	91
Base	137	431	167	38	139	21	107	611	100
Mistral-7B-Instruct									
Masked	405	272	232	177	49	32	445	507	247
Concrete	403	273	231	181	45	38	441	513	244
Base	427	416	307	176	46	36	437	530	239

Table 3: Error distributions from fine-tuned models. Shared = shared errors; New = new errors; Rec = recovered errors.

to free-form reasoning.

This interpretation is reinforced by masked training results. Masked and concrete supervision yield comparable in-distribution accuracy but diverge under distribution shift: PoT stabilizes reasoning procedure without grounding numerical semantics.

5 Implications and Future Directions

The central finding of this work is that consistency and correctness are dissociable. All models we evaluate exhibit substantial gaps between original and all-variants accuracy under PoT prompting. Explicit reasoning traces do not necessarily confer robustness to surface variation.

PoT supervision does not resolve this fragility; it reshapes it. Fine-tuning reduces new and recovered errors, yet Δ_{AVA} remains largely unchanged. What improves is not correctness but uniformity: models shift towards failing on the same problems across isomorphic variants; they learn to fail the same way. These gains are also representation-bound—they do not transfer across prompting formats or domains. This dissociation has direct implications for how we evaluate reasoning.

First, *measuring consistency is independently valuable*. A model that fails inconsistently is still searching; a model that fails consistently has stabilized on something learnable but wrong. Distinguishing these cases requires tracking not just accuracy but error structure across perturbations. Current benchmarks obscure this. A model with 90% accuracy and high Δ_{AVA} has learned to succeed on specific surface forms; a model with 85% accuracy and low Δ_{AVA} may have learned something more general. We argue that future evaluations should report both metrics and treat the gap as a signal of reasoning quality.

Second, *what supervision strategy could stabilize free-form reasoning?* PoT supervision induces in-distribution consistency but does not transfer to CoT evaluation. Reasoning models trained on CoT still exhibit substantial accuracy drops under perturbation, suggesting that neither approach induces robust CoT reasoning. Whether any supervision strategy can remains an open question.

Why does training induce consistency before correctness? One possibility is that these are distinct stages: models first stabilize on a procedure, then that procedure may or may not become correct. Our results demonstrate that the first stage can occur without the second. Consistency, then, may not be evidence of understanding, but evidence of commitment to a pattern. Studying learning dynamics directly could clarify whether this interpretation holds, which we leave to future work.

6 Conclusion

Program-of-Thought prompting is often regarded as a robustness technique. Our results partially support this view: PoT supervision does improve consistency across isomorphic variants. However, they also clarify its limits.

Fine-tuning causes models to converge to stable behaviors that do not transfer: gains vanish under CoT evaluation and domain shift. The consistency we observe is uniformity of behavior, including uniform failure. Models learn to fail the same way.

The picture that emerges is one of dissociation. Procedural consistency and semantic invariance are distinct, and progress on one does not necessarily entail progress on the other. The challenge is not simply to improve accuracy, but to understand when consistency reflects genuine abstraction versus memorization of a wrong procedure. Closing the gap may require rethinking what reasoning supervision is meant to teach.

Limitations

While our study provides meaningful insights into LLM abstraction under numeric perturbations, several important limitations remain.

Dataset Scope and Applicability to Other Benchmarks. Our study evaluates perturbation drops on GSM8K and MATH, fine-tuning only on GSM8K. These datasets consist of problems whose solutions can be expressed as parameterized procedures with numeric inputs to a fixed reasoning template. Under numeric substitution, the core computation structure remains unchanged, allowing perturbations to isolate whether models reliably apply the same method across instantiations.

This property does not generally hold for many harder competition-style benchmarks (e.g., AIME), where intended solutions often depend on instance-specific arithmetic relationships among the given numbers. Naively changing numbers in such problems frequently alters which strategies are valid, thereby changing the effective task. Although one can sometimes construct constraint-preserving variants by explicitly maintaining these relationships, such perturbations evaluate invariant recovery across a generated family rather than robustness of a fixed solution template under numeric variation. As a result, perturbation consistency measured on those datasets is not directly comparable to the consistency studied here.

Accordingly, while our results characterize robustness for broadly parameterized mathematical reasoning, they may not extend to domains in which solvability depends on delicate number-specific structure that is not preserved under straightforward perturbations.

Perturbation Type Coverage. ReasonAgain's perturbations vary numeric values while preserving program structure, isolating sensitivity to control-flow changes but not semantic robustness against synonym substitutions or rephrasings. Our analysis thus focuses on operational brittleness (e.g., rounding, boundary conditions) rather than linguistic brittleness. Future work should extend PoT evaluation to meaning-preserving perturbations like paraphrases and name changes, testing whether models can generate functionally equivalent programs or are merely relying on surface features.

Label Noise in ReasonAgain. Although ReasonAgain's programmatic extraction workflow efficiently generates gold answers, roughly 8% of perturbations carry incorrect labels in GSM8K and 17% in MATH—stemming from template drift, rounding mismatches, or extraction bugs. While filtering these noisy examples only modestly affects the observed robustness deltas, it highlights the need for more stringent validation (e.g., cross-verification with human annotation) to ensure future robustness benchmarks are both comprehensive and accurate.

Ethics Statement

This study uses publicly available datasets (GSM8K, MATH, and related reasoning benchmarks) and open-source models (GPT-OSS, LLaMA, Mistral, Deepseek, and Qwen) for experimental evaluation. Our analysis focuses on understanding the robustness and limitations of reasoning models under perturbations and program-based supervision. No human subjects, personally identifiable information, or high-stakes decision-making scenarios were involved.

Acknowledgements

This work was partially funded by ONR Contract N00014-23-1-2364 and NSF grant #IIS-2135581. Research was also sponsored by the Army Research Office and was accomplished under Grant Number W911NF-20-1-0080. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, Che Chang, Kai Chen, and 105 others. 2025. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. 2023. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *Transactions on Machine Learning Research*. Accepted November 2023.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Chris Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

DeepSeek-AI, Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenhao Xu, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, and 245 others. 2025. Deepseek-v3.2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, Soumya Sanyal, Sean Welleck, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi. 2023. Faith and fate: Limits of transformers on compositionality. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*. Curran Associates, Inc.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. PAL: Program-aided language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 10764–10799. PMLR.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 540 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track (NeurIPS 2021)*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. In *Proceedings of the 10th International Conference on Learning Representations (ICLR)*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Téléométr Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Meta AI. 2025. Llama 4: Multimodal intelligence. `https://ai.meta.com/blog/llama-4-multimodal-intelligence/`. Accessed: 2025-10-05.

Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. 2025. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models. In *Proceedings of the 13th International Conference on Learning Representations (ICLR)*.

Parshin Shojaee, Iman Mirzadeh, Keivan Alizadeh, Maxwell Horton, Samy Bengio, and Mehrdad Farajtabar. 2025. The illusion of thinking: Understanding the strengths and limitations of reasoning models via the lens of problem complexity. *arXiv preprint*, arXiv:2506.06941.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *Proceedings of the 11th International Conference on Learning Representations (ICLR)*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS 2022)*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Xiaodong Yu, Ben Zhou, Hao Cheng, and Dan Roth. 2024. Reasonagain: Using extractable symbolic programs to evaluate mathematical reasoning. *arXiv preprint arXiv:2410.19056*.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V. Le, and Ed H. Chi. 2023. Least-to-most prompting enables complex reasoning in large language models. In *Proceedings of the 11th International Conference on Learning Representations (ICLR)*.

A Comparison Across Evaluation Criteria

Model	FineTune	Program-of-Thought (GSM8K)						Program-of-Thought (MATH)						Chain-of-Thought (GSM8K)					
		Non-auto			Strong			Non-auto			Strong			Non-auto			Strong		
		Sh	New	Rec	Sh	New	Rec	Sh	New	Rec	Sh	New	Rec	Sh	New	Rec	Sh	New	Rec
Llama-3.1-8B																			
	Masked	121	348	74	48	23	13	43	107	35	17	64	3	102	570	93	28	102	10
	Concrete	121	348	74	48	23	13	45	97	29	21	58	2	100	572	93	27	103	13
	Base	155	569	157	20	17	35	39	106	22	20	58	3	102	604	101	35	98	14
Mistral-7B																			
	Masked	416	327	211	204	30	31	180	34	35	149	20	4	444	494	256	223	115	14
	Concrete	414	328	210	203	30	32	182	31	44	148	16	5	440	499	252	221	120	13
	Base	436	453	279	158	41	39	176	33	47	141	18	1	432	521	244	216	129	11

Table 4: Error distributions under two evaluation regimes: Non-auto-round and Strong. Sh = shared errors; New = new errors; Rec = recovered errors.

A.1 Alternative Definitions of Error Categories

We begin by examining robustness to stricter semantic equivalence criteria by adopting a stronger definition of error categories. Under this **strong** classification, error types are defined using universal rather than existential quantification over the perturbation set. Specifically, a shared error corresponds to a problem that the model answers incorrectly on the original instance and on all of its perturbations. A new error is a problem that the model answers correctly on the original instance but incorrectly on all perturbations. A recovered error is a problem that the model answers incorrectly on the original instance but correctly on all perturbations. All classifications here are done in the auto-round setting.

While new and recovered errors generally exhibit similar directional behavior in-distribution—aside from Llama-3.1-8B, where new errors increase slightly—the most consistent effect is a substantial rise in shared errors under PoT evaluation on GSM8K. For Llama-3.1-8B, shared errors increase from 20 in the base model to 48 under PoT fine-tuning. Mistral-7B exhibits the same pattern, with shared errors rising from 158 to 203–204. This systematic increase in shared errors indicates that PoT supervision concentrates error mass into stable, repeatable failure modes under stricter semantic equivalence criteria.

As in the non-rounded analysis, this consolidation remains domain- and channel-local. Under task shift to MATH and CoT evaluation, increases

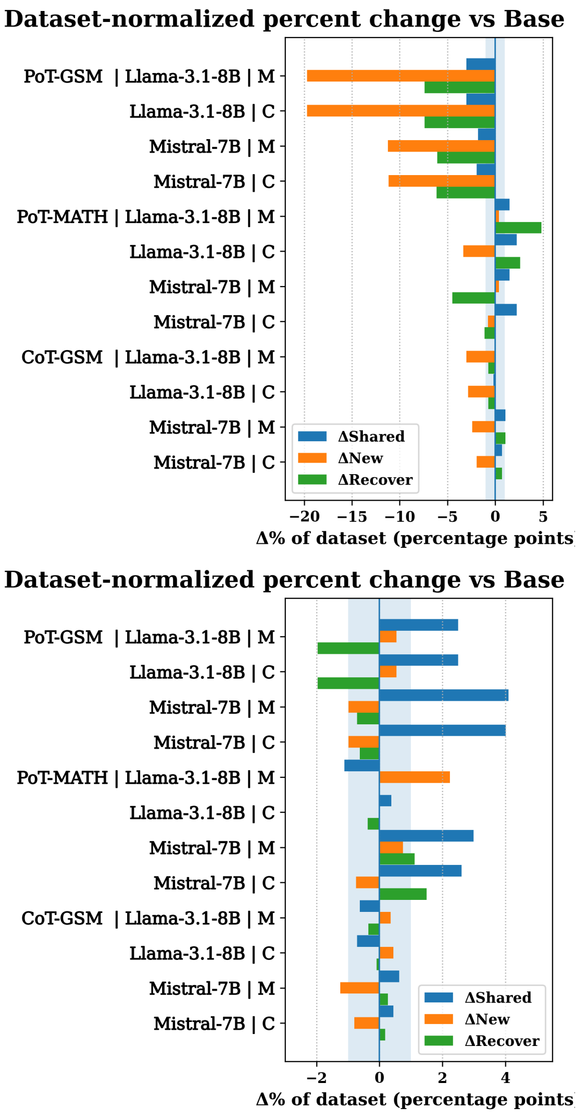

Figure 2: Dataset-normalized percent change vs base for Non-auto-round (top) and Strong definition (bottom).

in shared errors are less consistent and, in some cases, reversed. The absence of a uniform trend across these settings indicates that the consistency induced by PoT supervision does not transfer across tasks or reasoning formats.

The increase in shared errors under the strong definition reflects a smoothing effect: errors become less confined to individual phrasings and more uniform across equivalent perturbations. This may help explain why consistency gains need not improve accuracy—a model that, for example, originally answered incorrectly but recovered on some perturbations may, after training, answer all variants incorrectly, suppressing recovered errors while lowering AVA. Models become more consistent without becoming more correct.

A.2 Non-Rounded Answer Evaluation

Table 4 also reports error distributions under non-auto-rounded answer extraction. The pattern mirrors the auto-rounded evaluation in §4.2: PoT supervision improves consistency within the same reasoning channel and training distribution, but this trend does not generalize under task shift or CoT evaluation. This indicates that the consistency gains reported in §4.2 are not an artifact of answer normalization.