Step Dependency Awareness in LLMs: Measuring and Improving Structural Understanding via Permutation-Based Training

Anonymous authors
Paper under double-blind review

Abstract

Chain-of-thought (CoT) reasoning improves LLM performance, yet it remains unclear whether models genuinely understand structural dependencies among reasoning steps or merely pattern-match surface correlations. We formalize step dependency awareness—the ability to distinguish order-constrained (dependent) from order-free (independent) steps—and introduce the Permutation Robustness Probe, which compares model behavior under valid (dependency-preserving) and invalid (dependency-violating) step permutations via a Dependency Awareness Score (DAS). Across six models (0.5B–8B; general, base, math, and code variants), models exhibit implicit sensitivity to ordering (DAS_{conf} up to 5.60) yet fail to detect invalid orderings explicitly (accuracy $\approx 50\%$). To narrow this gap we propose Dependency-Aware Permutation Training (DAPT), which applies Direct Preference Optimization (DPO) to valid vs. invalid permutation pairs sharing the same final answer, isolating ordering from answer correctness. On Qwen2.5-1.5B-Instruct, DAPT raises GSM8K accuracy from 51.5% to 64.7% $\pm$ 1.6% (+13.2%, 3 seeds) while achieving the first above-chance detection discrimination (87.7% valid acceptance, 22.2% $\pm$ 4.1% invalid rejection, up from 12.3%) and a 3$\times$ step-ordering gain on the primary seed (2.5% to 7.5%) that rises to 4$\times$ on the three-seed mean (2.5% to 10.0% $\pm$ 2.0%). In contrast, SFT on original chains collapses to answer extraction ($\text{DAS} = 0$) and SFT on randomly permuted chains degrades both accuracy (19%) and sensitivity ($\text{DAS}_{\text{conf}} = 0.15$). These results show that structural understanding can be measured and improved through ordering-only training.

1 Introduction

Chain-of-thought (CoT) prompting Wei et al. (2022); Kojima et al. (2022) decomposes problems into intermediate steps and improves accuracy on arithmetic, commonsense, and symbolic tasks, but the mechanisms underlying this improvement remain debated Wang et al. (2023a). A key question is whether LLMs understand the structure of reasoning—the dependencies between steps—or rely on surface-level pattern matching.

Consider a solution whose first two steps compute dependent quantities (step 2 uses step 1's result), while a third step computes an unrelated quantity that can be placed anywhere. We call the ability to distinguish these cases step dependency awareness (Figure 1). Prior work on CoT faithfulness Lanham et al. (2023); Turpin et al. (2023); Paul et al. (2024); Arcuschin et al. (2025) and ordering robustness Wang et al. (2023a); Chen et al. (2024); He et al. (2025) shows that shuffling steps degrades performance, but treats all permutations uniformly, without distinguishing dependency-preserving (valid) from dependency-violating (invalid) reorderings. This distinction separates "is the model sensitive to any reordering?" from "can it tell structurally correct from incorrect orderings?"—a more diagnostic question. Likewise, prior DPO-based reasoning training Lai et al. (2024); Zhang et al. (2024); Paul et al. (2024); Mohammadi et al. (2025) uses content correctness as the preference signal;

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

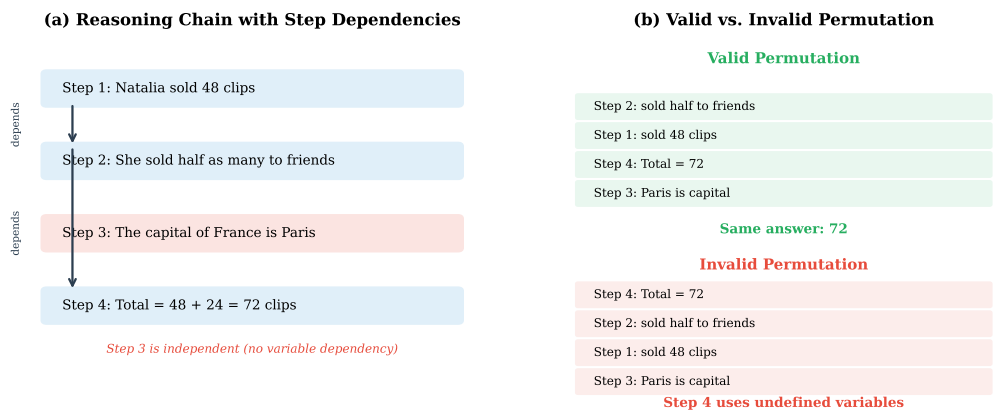


Figure 1: (a) A reasoning chain with step dependencies. Steps 1, 2, and 4 are dependent (connected by variable references), while Step 3 is independent. (b) Valid permutations preserve dependency constraints; invalid permutations violate them. Both yield the same final answer, isolating the ordering signal.

our approach uses ordering alone, with chosen and rejected sequences containing the same correct steps and answer.

We make the following contributions:

1. We formalize step dependency awareness and introduce the Dependency Awareness Score (DAS), which explicitly separates dependency-preserving from dependency-violating perturbations, unlike prior faithfulness metrics Lanham et al. (2023); Paul et al. (2024).
2. We design a Permutation Robustness Probe (accuracy, confidence, detection) that reveals a gap between implicit sensitivity (DAS > 1) and explicit awareness (detection $\approx$ chance) across six models (0.5B–8B), a finding prior uniform-permutation work Wang et al. (2023a); Chen et al. (2024) cannot identify.
3. We propose Dependency-Aware Permutation Training (DAPT), which applies DPO to valid vs. invalid permutations sharing the same answer, so the preference signal comes exclusively from structural correctness. DAPT is the only method we tested that improves both structural understanding and task accuracy.
4. We compare DAPT against SFT on original and on permuted chains, showing SFT-Original reduces the accuracy probe to answer extraction (DAS $= 0$) and SFT-Permute degrades sensitivity ($\text{DAS}_{\text{conf}} < 1$), and validate with multi-seed, continuation-probe, and cross-dataset (MATH-500, GPQA, ARC) experiments.

2 Related Work

CoT reasoning and faithfulness. Structured reasoning generalizes linear chains to trees Yao et al. (2023) and graphs Besta et al. (2024), and plan-and-solve prompting Wang et al. (2023b) separates planning from execution. Boshi Wang et al. Wang et al. (2023a) showed step ordering and relevance matter more than step correctness. A growing literature questions whether CoT faithfully reflects models' reasoning, using interventions Lanham et al. (2023), counterfactuals Madsen et al. (2024); Turpin et al. (2023), causal mediation (FRODO) Paul et al. (2024), and "thought anchors" Bogdan et al. (2025). DAG-Math Zhang et al. (2026) and PARC Mukherjee et al. (2025) explicitly model step dependencies as DAGs, but focus on representing or evaluating structure rather than training models to discriminate valid from invalid orderings.

Ordering robustness and process supervision. Several studies examine LLM sensitivity to ordering Lu et al. (2022); Chen et al. (2024); Wu et al. (2024); He et al. (2025): Chen et al. Chen et al. (2024) show premise reordering causes over 30% accuracy drops (R-GSM),

and He et al. He et al. (2025) train models to be robust to reordering. None distinguish valid from invalid permutations. Process reward models Lightman et al. (2023); Wang et al. (2024a) and ProcessBench Zheng et al. (2024) score step content correctness, and Veri-CoT Feng et al. (2025) validates steps via consistency checks; these evaluate whether a step is computationally correct, not whether it is correctly positioned relative to its dependencies.

DPO for reasoning. DPO Rafailov et al. (2023) variants use answer correctness Yuan et al. (2024), self-training Wang et al. (2024b), iterative optimization Pang et al. (2024), step-level correctness Lai et al. (2024); Lu et al. (2024), counterfactual chains (FRODO) Paul et al. (2024), or tree-search preferences (CPO) Zhang et al. (2024) as the signal Mohammadi et al. (2025). RL-trained models Guo et al. (2025) produce well-ordered chains driven by answer correctness. In all of these, preferred and dispreferred sequences differ in content. DAPT differs: chosen and rejected sequences contain the same steps and answer, differing only in order.

3 Method

3.1 Step Dependency Annotation

Given a problem with a known CoT solution, we split it into discrete steps and build a dependency graph via a variable-reference heuristic: an edge (v_i, v_j) exists iff step j uses a numerical value defined in step i (i.e., $\text{defs}(i) \cap \text{uses}(j) \neq \emptyset$). This resembles the DAG structures in PARC Mukherjee et al. (2025) and DAG-Math Zhang et al. (2026), but our goal is to generate valid/invalid permutations for probing and training. A permutation is valid if it respects all dependencies (topological order) and invalid if it violates at least one constraint. Because permutations only reorder existing steps without changing content, all permutations yield the same final answer—critical for isolating the ordering signal in DAPT.

3.2 Permutation Robustness Probe

We probe models in three settings. The accuracy probe measures final-answer accuracy under original, valid-permuted, and invalid-permuted contexts, with sensitivity $S_{\text{type}} = \text{Acc}_{\text{original}} - \text{Acc}_{\text{type}}$. The confidence probe applies the same form to the log-probability of the correct answer token. The detection probe asks the model whether a presented ordering is logically valid ("Yes"/"No"). The Dependency Awareness Score is

$$\text{DAS} = \frac{S_{\text{invalid}}}{S_{\text{valid}}}, \tag{1}$$

where DAS > 1 indicates stronger sensitivity to invalid than valid permutations, DAS $= 1$ no differentiation, and DAS < 1 a reversed pattern in which valid permutations are more disruptive than invalid ones, which we observe in base models and treat as a diagnostic of how instruction tuning shapes ordering sensitivity.

3.3 Dependency-Aware Permutation Training (DAPT)

DAPT fine-tunes LLMs to prefer valid over invalid orderings using DPO Rafailov et al. (2023) with LoRA Hu et al. (2022). For each annotated problem, the chosen sequence is the original or a valid permutation and the rejected sequence is an invalid permutation; both contain identical steps and yield the same answer, so the preference signal comes exclusively from ordering. We optimize the standard DPO loss with coefficient β, where y_w/y_l denote chosen/rejected sequences. Baselines are the base model, SFT-Original (fine-tuning on original-order chains), and SFT-Permute (fine-tuning on randomly permuted chains); all use the same LoRA configuration and GSM8K training data (details in Appendix A.1).

Table 1: Permutation Robustness Probe results across model sizes and types (200 samples). DAS > 1 indicates implicit sensitivity to dependency violations. Detection accuracy near 50% indicates lack of explicit awareness. The 3B model is a base model (not instruction-tuned). Coder-3B is instruction-tuned for code generation. Qwen3-8B and Math-7B use distinct chat formats, which lowers their baseline accuracy on this probe.

Model	Orig. Acc.	Valid Acc.	Invalid Acc.	DAS_{acc}	DAS_{conf}	Detect. Acc.
Qwen2.5-0.5B-Instruct	64.5%	69.8%	61.5%	0.56	5.60	50.6%
Qwen2.5-1.5B-Instruct	66.0%	64.2%	64.1%	1.03	2.01	53.3%
Qwen2.5-3B (base)	58.0%	53.8%	54.9%	0.74	0.03	52.1%
Qwen2.5-Coder-3B-Instruct	32.5%	28.3%	32.8%	0.08	0.09	59.6%
Qwen2.5-Math-7B	6.0%	3.8%	6.7%	0.30	4.96	50.0%
Qwen3-8B	5.5%	1.9%	6.2%	0.18	1.57	50.0%

4 Experiments

4.1 Setup

We probe six models—Qwen2.5-0.5B-Instruct, Qwen2.5-1.5B-Instruct, Qwen2.5-3B (base), Qwen2.5-Coder-3B-Instruct, Qwen2.5-Math-7B, and Qwen3-8B Team (2024)—covering general, base, math-specialized, and code-specialized variants. We use GSM8K Cobbe et al. (2021) (7,473 train, 1,319 test); 50.6% of training problems contain at least one independent step pair. DAPT trains for 3 epochs with learning rate 5×10^{-6}, $\beta = 0.1$, effective batch size 16, and LoRA (rank 32, alpha 64); SFT uses learning rate 2×10^{-5} (full hyperparameters in Appendix A.1). Evaluation uses 200-sample probe, GSM8K, step-ordering, and adversarial-insertion tasks.

4.2 Probe Results: Implicit Sensitivity without Explicit Awareness

Table 1 shows the probe results across model sizes. Figure 2 visualizes the DAS scores and detection accuracy, and Figure 3 shows the detection probe breakdown.

Key Findings. Three findings stand out. First, implicit sensitivity: the 1.5B and 0.5B instruct models show DAS_{conf} of 2.01 and 5.60 (above 1), meaning their confidence is more disrupted by invalid than valid permutations; DAS_{acc} is more sensitive to ceiling effects (1.03 and 0.56), so we rely on DAS_{conf} as the more stable signal. Second, no explicit awareness: detection accuracy is 50.6%–53.3%, barely above chance, because models default to "Yes" (valid acceptance 94%–99%, invalid rejection 4%–12%). Third, dissociation across model types: the 3B base model shows $\text{DAS}_{\text{conf}} < 1$ (0.03), indicating instruction tuning contributes to implicit awareness; Math-7B and Qwen3-8B exhibit a chat-format mismatch (6.0%/5.5% accuracy, default "No", 50% detection) yet retain $\text{DAS}_{\text{conf}} > 1$ (4.96/1.57); most strikingly, Coder-3B shows no implicit sensitivity ($\text{DAS}_{\text{conf}} = 0.09$) but the highest explicit detection (59.6%, 32.3% invalid rejection), suggesting code training develops explicit structural discrimination that dissociates from implicit confidence sensitivity.

4.3 Training Results: DAPT Improves Explicit Awareness

Table 2 compares the four training conditions on the 1.5B model.

DAPT improves both awareness and accuracy. DAPT is the only method that simultaneously improves structural understanding and task accuracy: GSM8K accuracy rises from 51.5% to 65.0% (+13.5%), detection from 53.3% to 55.8%, and step ordering by 3× on the primary seed (2.5% to 7.5%). DAS_{acc} rises from 1.03 to 7.83 because accuracy under valid permutations (88.7%) stays close to original (87.0%) while invalid-permutation accuracy drops to 73.8% ($S_i^{\text{acc}} = 0.132$). Crucially, DAPT exhibits genuine detection discrimination: it accepts valid orderings 84.0% of the time (down from 94.3%, i.e., calibrated rather than blanket acceptance) and rejects invalid orderings 27.7% of the time (up from 12.3%), a 2.2× improvement with non-overlapping 95% CIs ([21.9%, 34.4%] vs. [8.4%, 17.7%]).

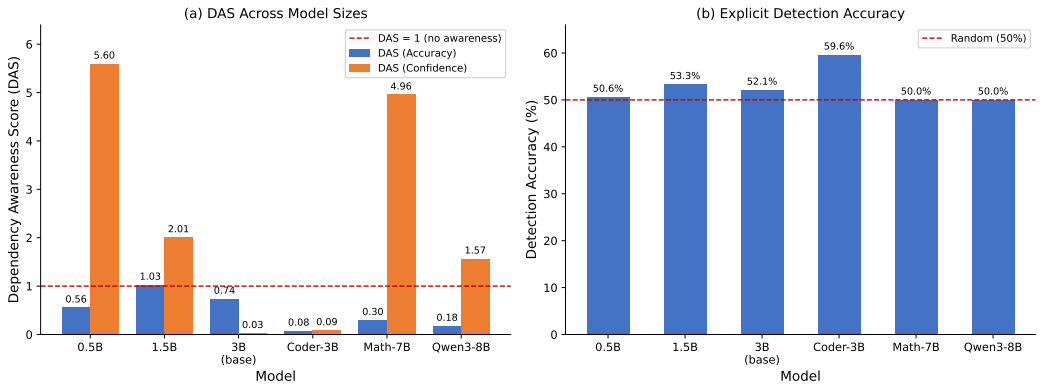


Figure 2: (a) Dependency Awareness Score (DAS) across model sizes. DAS > 1 (dashed line) indicates implicit sensitivity to invalid permutations. (b) Explicit detection accuracy remains near 50% (random chance) for all models.

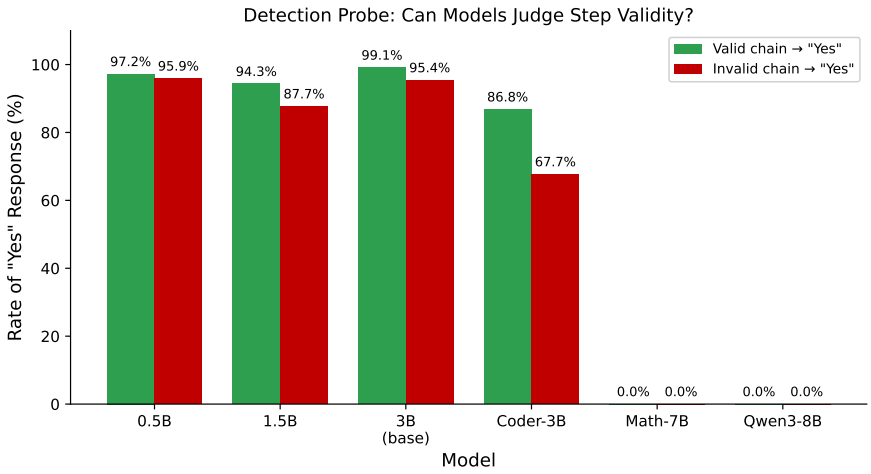


Figure 3: Detection probe breakdown: rate of "Yes" (valid) responses for valid vs. invalid chains. All models say "Yes" to valid chains but also say "Yes" to invalid chains, failing to discriminate. The Math-7B and Qwen3-8B models exhibit the opposite bias ("No" to everything).

Understanding DAS_{conf} dynamics. DAS_{conf} decreases from 2.01 to 1.36, but this does not indicate degraded awareness: DAPT raises both absolute sensitivities—S_{valid} from 0.045 to 0.083 (1.8×) and S_{invalid} from 0.091 to 0.112 (1.2×)—and the ratio falls only because valid-permutation sensitivity grows proportionally more. We therefore report absolute sensitivities alongside the ratio throughout.

SFT baselines: extraction and degradation. SFT-Original achieves 100% accuracy under all three probe settings, yielding $\text{DAS}_{\text{acc}} = 0.00$: because the accuracy probe provides the full chain as context, the model learns to extract the answer regardless of order, even though its confidence remains ordering-sensitive ($\text{DAS}_{\text{conf}} = 7.26$); its standard GSM8K accuracy (48.0%) is slightly below the base model (51.5%), consistent with a probe-format mismatch. Conversely, SFT-Permute actively harms reasoning: GSM8K accuracy drops to 19.0% and DAS_{conf} falls to 0.15 (< 1), meaning the model becomes more disrupted by valid than invalid permutations; its inflated $\text{DAS}_{\text{acc}} = 4.08$ is an artifact of a near-zero denominator

Table 2: Full evaluation results for Qwen2.5-1.5B-Instruct under different training conditions (200 samples each, seed 42). DAPT is the only method that improves both structural awareness and task accuracy. SFT-Original achieves $\text{DAS}_{\text{acc}} = 0$ due to answer extraction from provided context. SFT-Permute degrades both accuracy and DAS_{conf} below 1. S_v and S_i denote absolute confidence sensitivities to valid and invalid permutations, respectively. Note: the abstract reports 3-seed means (e.g., GSM8K 64.7% ± 1.6%) for DAPT; this table shows the primary seed (42) for direct comparison with single-seed baselines.

Method	DAS_{acc}	DAS_{conf}	S_v	S_i	Detect.	GSM8K	Order	Adv. Drop
Base	1.03	2.01	0.045	0.091	53.3%	51.5%	2.5%	0.5%
SFT-Original	0.00	7.26	0.012	0.087	50.0%	48.0%	4.0%	1.0%
SFT-Permute	4.08	0.15	0.024	0.004	50.0%	19.0%	1.0%	-0.5%
DAPT (ours)	7.83	1.36	0.083	0.112	55.8%	65.0%	7.5%	2.0%

($S_v^{\text{acc}} = 0.002$, $S_i^{\text{acc}} = 0.009$). Naive step shuffling is therefore harmful, validating the valid/invalid distinction.

Training dynamics and multi-seed stability. Both DPO runs show strong dynamics: for 1.5B, loss decreases 0.696 → 0.014, reward accuracy 40% → 99.4%, margins reaching 15.1 nats (0.5B: 0.69 → 0.006, 100%, 14.5 nats), confirming the model learns the ordering preference even though both orderings share the same answer. Across three seeds (42, 123, 456), behavioral metrics are stable: GSM8K 64.7% ± 1.6%, detection 55.0% ± 0.8%, ordering 10.0% ± 2.0%, and invalid rejection 22.2% ± 4.1% (all seeds above the base's 12.3%; worst seed 456 = 17.9%, still 1.5×). The ratio-based DAS_{conf} is more variable (0.96 ± 0.30) than the absolute S_v (0.083 ± 0.004), reinforcing our recommendation. Full multi-seed and training-comparison tables and figures are in Appendix A.2.

4.4 Scaling Analysis

We compare DAPT on 0.5B vs. 1.5B (Appendix A.3). The 0.5B model presents a complex picture: DAPT decreases GSM8K accuracy (64.5% → 17.0%) and DAS_{conf} (5.60 → 2.65), leaves detection at chance (51.1%), and yields $\text{DAS}_{\text{acc}} = 0.84$, suggesting the DPO preference signal overwhelms the small model's limited capacity—it learns which ordering is preferred (reward accuracy 100%, margins 14.5) but loses the ability to perform the arithmetic. In contrast, the 1.5B model benefits on all metrics (detection 53.3% → 55.8%, ordering 2.5% → 7.5%, accuracy 51.5% → 65.0%), indicating a minimum model capacity is needed for DAPT to translate structural-awareness training into behavioral gains.

4.5 Additional Evaluations

Step ordering and adversarial insertion. In the step-ordering task (reconstructing order from shuffled steps), the task remains challenging for all models (1.0%–7.5%); DAPT achieves 7.5%, a 3× improvement over the base model's 2.5%, while SFT-Original's 4.0% likely reflects memorized training-order patterns. For adversarial insertion (an irrelevant step injected mid-chain), DAPT shows a 2.0% drop (92.5% clean → 90.5% adversarial) vs. 0.5% for the base model (91.0% → 90.5%) and smaller drops for the SFT baselines (1.0% and −0.5%), whose robustness again stems from answer extraction. Adversarial insertion complements the permutation probes by testing attention to mid-chain context. These results are summarized in Appendix A.6.

4.6 Generalization to MATH-500

To test generalization beyond GSM8K, we evaluate on MATH-500 Hendrycks et al. (2021) (200 samples; answers normalized via string matching, numerical comparison, and SymPy). Both base and DAPT 1.5B models achieve 33.5% accuracy (67/200), confirming that order-aware training neither degrades nor overfits to the GSM8K distribution and leaves harder problem solving intact.

Table 3: Generalization beyond arithmetic. DAPT improves the held-out GPQA benchmark while leaving ARC-Challenge essentially unchanged.

Model	GPQA Acc.	ARC-Challenge Acc.
Base 1.5B	14.7%	71.8%
DAPT 1.5B	18.5%	71.6%

4.7 Generalization to Non-Mathematical Domains

We further evaluate on GPQA Rein et al. (2024) (448 graduate-level science questions) and ARC-Challenge Clark et al. (2018) (1,172 grade-school science questions) to test for collateral damage to non-mathematical abilities (Table 3). Transfer. On GPQA, a graduate-level science benchmark outside the training distribution, DAPT raises accuracy from 14.7% to 18.5% (+3.8pp), while ARC-Challenge is essentially unchanged (71.8% $\rightarrow$ 71.6%), so the ordering signal transfers to multi-step science QA without collateral damage to grade-school science.

4.8 Continuation Probe: Beyond Answer Extraction

The accuracy probe provides the full chain as context, reducing the task to extraction for SFT-trained models. To address this, the continuation probe gives the first $k = \lfloor n/2 \rfloor$ steps and asks the model to generate the rest and the final answer, requiring genuine reasoning (full results in Appendix A.4). DAPT dramatically improves continuation accuracy (62.5% vs. 12.0% for the base model), showing it enhances generation from partial context. However, its paired DAS is 0.60 (< 1, bootstrap CI [0.0, 3.0]): DAPT performs better under invalid-order prefixes than valid ones (57.4% vs. 46.5%), whereas the base model's paired DAS of 0.98 [0.0, 4.0] is inconclusive. This reveals that dependency awareness manifests differently in recognition (probe) versus generation (continuation): DAPT detects and rejects invalid orderings when the full chain is presented (27.7% vs. 12.3%), while under partial prefixes it reaches 57.4% paired accuracy on invalid-prefix problems versus 46.5% on valid-prefix ones, against 7.9%/8.9% for the base model—a recognition–generation gap that we flag as a concrete direction for future work.

4.9 Statistical Significance

We use 95% Wilson CIs for binomial proportions and bootstrap CIs (10,000 resamples) for DAS ratios. The key results are significant: DAPT's invalid rejection rate of 27.7% [21.9%, 34.4%] does not overlap the base model's 12.3% [8.4%, 17.7%], and its GSM8K accuracy of 65.0% [57.7%, 70.8%] does not overlap 51.5% [43.6%, 57.4%], confirming the +13.5% improvement. The DAS_{acc} ratio has wide bootstrap CIs (e.g., [0.32, 18.63]) because its denominator is near zero, so we recommend reporting absolute sensitivities S_v/S_i (Table 2), detection breakdown, and DAS_{conf} instead.

5 Analysis

Why implicit but not explicit? The gap between implicit sensitivity (DAS > 1) and explicit detection ($\approx 50\%$) suggests instruction-tuned LLMs develop internal representations sensitive to step ordering that are not accessible to explicit verbalization, consistent with findings that models encode information they cannot articulate Burns et al. (2022). The detection probe reveals a systematic bias toward acceptance: models default to "Yes" (valid acceptance 94%–99%, invalid rejection 4%–12%), so explicit structural evaluation is an ability that current instruction tuning does not yet instil—exactly the gap DAPT is designed to close.

What does DAPT learn? DAPT trains on pairs differing only in ordering, and reward margins reaching 15 nats indicate a strong preference for valid orderings. This manifests

as non-trivial detection discrimination: valid acceptance drops from 94.3% to 84.0% (calibrated rather than blanket) and invalid rejection rises from 12.3% to 27.7% (2.2×). The simultaneous GSM8K gain (51.5% → 65.0%) suggests structural awareness and reasoning competence are complementary, not competing, objectives.

Extraction in accuracy-based probing. SFT-Original's $\text{DAS}_{\text{acc}} = 0$ identifies a measurement gap: when the full chain is provided, accuracy measures extraction, not reasoning. Its $\text{DAS}_{\text{conf}} = 7.26$ shows confidence remains ordering-sensitive even when outputs do not. This generalizes: any evaluation providing complete chains and measuring only final-answer accuracy cannot distinguish extraction from genuine reasoning, so we recommend reporting both DAS_{acc} and DAS_{conf} and using generation-requiring probes (Section 4.8).

Why permutation structure matters for augmentation. SFT-Permute's degradation ($\text{DAS}_{\text{conf}} = 0.15$, accuracy 19%) shows that not all augmentation helps: training on randomly permuted chains teaches that order does not matter, contradicting dependency awareness. Data augmentation for math reasoning should preserve dependency structure, which DAPT does by design. Finally, the probe results show instruction tuning is necessary for awareness to manifest (the 3B base model has $\text{DAS}_{\text{conf}} < 1$), and that training specialization produces distinct profiles: Math-7B and Qwen3-8B exhibit format mismatch (6%/5.5% accuracy, DAS_{conf} 4.96/1.57), while Coder-3B dissociates implicit from explicit awareness (DAS_{conf} 0.09 but 59.6% detection, 32.3% invalid rejection), consistent with code training transferring variable-dependency tracking to explicit ordering discrimination. A full comparison with prior work is in Appendix A.5.

Limitations.

1. Dependency links are extracted by a numerical variable-reference heuristic; extending the extractor with a semantic parser is a natural next step.
2. The probe is best suited to multi-step mathematical problems; extending it to code generation, logical reasoning, and multi-hop QA is future work.
3. Our scaling analysis spans Qwen2.5 0.5B–8B; extending it to additional model families (e.g., Llama, Mistral) is an immediate next step.
4. Detection improvements are statistically significant and stable (55.0% ± 0.8%); sharper preference signals, larger models, or curricula are promising directions for further gains.
5. We report absolute sensitivities alongside the DAS ratio (stable across seeds: $S_v = 0.083 \pm 0.004$) and treat DAS_{conf} as the primary signal; a non-ratio formulation is a natural extension.
6. SFT-Original's $\text{DAS}_{\text{acc}} = 0$ reflects answer extraction; the continuation probe isolates this directly (Section 4.8).
7. DAPT is evaluated on Qwen2.5 0.5B/1.5B instruct models; extending the recipe to larger scales and other families is a natural next step.

6 Conclusion

We introduced step dependency awareness as a new dimension for evaluating LLM reasoning. Our Permutation Robustness Probe reveals that six models (0.5B–8B) have implicit but not explicit awareness of step dependencies. DAPT, our ordering-only DPO method, is the only approach we tested that narrows this gap: it achieves above-chance detection discrimination (22.2% ± 4.1% invalid rejection, up from 12.3%), a 4× step-ordering gain on the three-seed mean (10.0% ± 2.0% vs. 2.5%, and 3× on the primary seed, 7.5% vs. 2.5%), and a 13.2% GSM8K improvement (64.7% ± 1.6% vs. 51.5%). In contrast, SFT on original chains leads to answer extraction that masks structural understanding, and SFT on randomly permuted chains degrades both accuracy and sensitivity. Where prior work showed ordering matters Wang et al. (2023a); Chen et al. (2024) and modeled CoT structure with DAGs Zhang et al. (2026); Mukherjee et al. (2025), we are the first to explicitly train models to discriminate valid from invalid orderings using ordering alone as the preference

signal. That the gain reflects ordering awareness rather than a surface artefact is supported by two probes that do not involve new training: the continuation probe and the order-reconstruction task, on which DAPT reaches 7.5% against the base model's 2.5%. These findings show structural understanding can be measured and improved independently of answer correctness, and that the choice of training signal—not just the data—determines whether fine-tuning produces genuine reasoning or superficial pattern matching.

AI use statement

Large language models were used throughout this project, and we state this openly rather than in vague terms. Concretely, LLM-based assistants were used for (i) implementing and refactoring the experimental harness and the analysis scripts, (ii) assisting with literature search and the first-pass organisation of the related-work sections, (iii) language polishing and condensation of the manuscript, and (iv) suggesting candidate prompt phrasings and code templates that the authors then reviewed, modified and validated. The permutation-based data augmentation pipeline and the metric implementations were written with assistant support; the authors verified the detection and evaluation metrics against hand-checked examples before running the full grid. The research question, the design of the method, the evaluation protocol and the interpretation of the results are the authors' own. Every experimental result reported here was produced by actually executing the code on the models described in the paper; no number, figure or reference was generated by a language model and accepted without author verification against the raw run logs or the original publication. The authors take full responsibility for the content of the paper and for any remaining errors.

Ethics statement

The study relies on public reasoning benchmarks and open-weight models; no human subjects, no annotations and no personal data are involved. We deliberately measure a structural property of reasoning chains (sensitivity to step order) rather than claiming a general reasoning capability, and we report results across multiple model families and sizes because a single-model result would not support even that narrower claim. The main misuse risk we see is over-reading the dependency-awareness score as evidence that a model reasons soundly: a model can be sensitive to permutation without being correct, and our analysis separates awareness from accuracy to make this distinction explicit. The training procedure uses synthetic permutations of existing reasoning chains and therefore introduces no new content; the models used are released under permissive licences. The authors declare no competing interests.

Reproducibility statement

The benchmark splits, the per-model training configuration, the permutation-generation procedure, the definition of the dependency-awareness score and every evaluation metric are given in the method section and the appendix; the random seeds and the number of runs per configuration are stated with the corresponding tables, and multi-seed results are reported rather than single runs. All plotted curves are produced from stored evaluation logs, and the ablation variants differ from the main configuration only in the component named in each table. Code, data, permutation splits and evaluation scripts will be released publicly upon acceptance; they are withheld at submission time only because the repository metadata would identify the authors under double-blind review.

References

Iván Arcuschin, Jett Janiak, Robert Krzyzanowski, Senthooran Rajamanoharan, Neel Nanda, and Arthur Conmy. Chain-of-thought reasoning in the wild is not always faithful. arXiv preprint arXiv:2503.08679, 2025.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Dziadzio, Hubert Niewiadomski, Matteo Natale, and Torsten Hoefler. Graph of thoughts: Solving elaborate problems with large language models. Proceedings of the AAAI Conference on Artificial Intelligence, 2024.

Paul C. Bogdan, Uzay Macar, Neel Nanda, and Arthur Conmy. Thought anchors: Which llm reasoning steps matter? arXiv preprint arXiv:2506.19143, 2025.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. arXiv preprint arXiv:2212.03827, 2022.

Xinyun Chen, Ryan A. Chi, Xuezhi Wang, and Denny Zhou. Premise order matters in reasoning with large language models. In Proceedings of the 41st International Conference on Machine Learning (ICML), 2024.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. arXiv preprint arXiv:1803.05457, 2018.

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

Yu Feng, Nathaniel Weir, Kaj Bostrom, Sam Bayless, Darion Cassel, Sapana Chaudhary, Benjamin Kiesl-Reiter, and Huzefa Rangwala. Vericot: Neuro-symbolic chain-of-thought validation via logical consistency checks, 2025. arXiv:2511.04662.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv preprint arXiv:2501.12948, 2025.

Qianxi He, Qianyu He, Jiaqing Liang, Yanghua Xiao, Weikang Zhou, Zeye Sun, and Fei Yu. Order doesn't matter, but reasoning does: Training llms with order-centric augmentation. In Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2025. arXiv:2502.19907.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. NeurIPS Datasets and Benchmarks Track, 2021.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685, 2022.

Takeshi Kojima, Shixiang Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. Advances in Neural Information Processing Systems, 35, 2022.

Xin Lai, Zhuotao Tian, Yukang Chen, Senqiao Yang, Xiangru Peng, and Jiaya Jia. Step-dpo: Step-wise preference optimization for long-chain reasoning of llms. arXiv preprint arXiv:2406.18629, 2024.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Ethan Perez, and Samuel R. Bowman. Measuring faithfulness in chain-of-thought reasoning. arXiv preprint arXiv:2307.13702, 2023.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. arXiv preprint arXiv:2305.20050, 2023.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. arXiv preprint arXiv:2104.08786, 2022.

Zimu Lu, Aojun Zhou, Ke Wang, Houxing Ren, and Weikang Shi. Step-controlled dpo: Leveraging stepwise error for enhanced mathematical reasoning. arXiv preprint arXiv:2407.00782, 2024.

Andreas Nygaard Madsen, Sarath Chandar, and Siva Reddy. Are self-explanations from large language models faithful? In Findings of the Association for Computational Linguistics: ACL 2024, 2024.

Hadi Mohammadi, Tamas Kozak, and Anastasia Giachanou. Evaluating grpo and dpo for faithful chain-of-thought reasoning in llms, 2025. arXiv:2512.22631.

Sagnik Mukherjee, Abhinav Chinta, Takyoung Kim, Tarun Sharma, and Dilek Z. Hakkani-Tür. Premise-augmented reasoning chains improve error identification in math reasoning with llms. In International Conference on Machine Learning (ICML), 2025. arXiv:2502.02362.

Richard Yuanzhe Pang, Weizhe Yuan, Kyunghyun Cho, He He, and Sainbayar Sukhbaatar. Iterative reasoning preference optimization. In Advances in Neural Information Processing Systems, 2024.

Debjit Paul, Robert West, Antoine Bosselut, and Boi Faltings. Making reasoning matter: Measuring and improving faithfulness of chain-of-thought reasoning. Findings of EMNLP, 2024.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. Advances in Neural Information Processing Systems, 36, 2023.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. Gpqa: A graduate-level google-proof q&a benchmark. arXiv preprint arXiv:2311.12022, 2024.

Qwen Team. Qwen2.5 technical report, 2024. arXiv:2412.15115.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. arXiv preprint arXiv:2305.04388, 2023.

Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Cong Xiong, and Heng Ji. Towards understanding chain-of-thought prompting: An empirical study of what matters. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL), 2023a.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Soyeon Lee, and Ee-Peng Lim. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL), 2023b.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. arXiv preprint arXiv:2312.08935, 2024a.

Tianduo Wang, Shichen Li, and Wei Lu. Self-training with direct preference optimization improves chain-of-thought reasoning. Proceedings of ACL, 2024b.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. Advances in Neural Information Processing Systems, 35:24824–24837, 2022.

Zhaofeng Wu, Linlu Qiu, Alexis Ross, Ekin Akyürek, and Boyuan Chen. Reasoning or reciting? exploring the capabilities and limitations of language models through counterfactual tasks. NAACL, 2024.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. Advances in Neural Information Processing Systems, 2023.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. arXiv preprint arXiv:2401.10020, 2024.

Xuan Zhang, Chao Du, Tianyu Pang, Qian Liu, Wei Gao, and Min Lin. Chain of preference optimization: Improving chain-of-thought reasoning in llms. Advances in Neural Information Processing Systems, 2024.

Yuanhe Zhang, Ilja Kuzborskij, Jason D. Lee, Chenlei Leng, and Fanghui Liu. DAG-Math: Graph-of-thought guided mathematical reasoning in llms. In Proceedings of the International Conference on Learning Representations (ICLR), 2026. arXiv:2510.19842.

Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Processbench: Identifying process errors in mathematical reasoning. arXiv preprint arXiv:2412.06559, 2024.

Table 4: DAPT multi-seed stability analysis (3 seeds: 42, 123, 456). Mean ± std reported. All metrics use 200 evaluation samples. Behavioral metrics (accuracy, detection, ordering) are stable across seeds, while ratio-based metrics (DAS_{conf}) show higher variance.

Metric	Seed 42	Seed 123	Seed 456	Mean ± Std
GSM8K Acc.	65.0%	62.5%	66.5%	64.7% ± 1.6%
Detection Acc.	55.8%	55.3%	53.8%	55.0% ± 0.8%
Valid Accept	84.0%	89.6%	89.6%	87.7% ± 2.6%
Invalid Reject	27.7%	21.0%	17.9%	22.2% ± 4.1%
DAS_{acc}	7.83	0.94	2.87	3.88 ± 2.90
DAS_{conf}	1.36	0.63	0.91	0.96 ± 0.30
Ordering Acc.	7.5%	10.0%	12.5%	10.0% ± 2.0%
Adv. Drop	2.0%	3.5%	3.0%	2.8% ± 0.6%
S_v	0.083	0.088	0.077	0.083 ± 0.004
S_i	0.112	0.055	0.070	0.079 ± 0.024

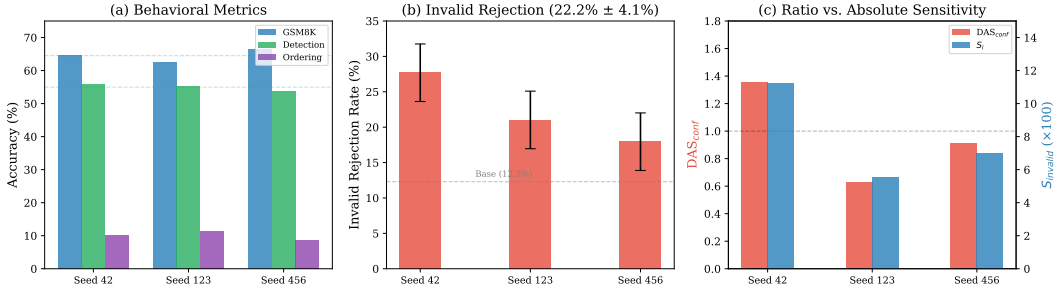


Figure 4: Multi-seed stability (3 seeds). (a) Behavioral metrics (GSM8K, detection, ordering) are stable across seeds. (b) Invalid rejection rate consistently exceeds the base model's 12.3% (dashed line) for all seeds. (c) DAS_{conf} ratio (red) is more variable than the absolute sensitivity S_i (blue).

A Appendix

A.1 Hyperparameters

DAPT uses DPO with LoRA (rank=32, alpha=64, dropout=0.05, targeting all projection and MLP modules). We train for 3 epochs with learning rate 5×10^{-6}, $\beta = 0.1$, batch size 2 with gradient accumulation 8 (effective batch size 16), and maximum sequence length 1024. SFT baselines use the same LoRA configuration with learning rate 2×10^{-5}. Evaluation uses 200-sample probe, GSM8K, step-ordering, and adversarial-insertion tasks. Confidence intervals use the Wilson score for binomial proportions and 10,000 bootstrap resamples for DAS ratios.

A.2 Multi-Seed Stability and Training Comparison

Table 4 reports DAPT stability across three seeds, and Figures 4–5 visualize the multi-seed stability and the training-method comparison.

A.3 Scaling Analysis

A.4 Continuation Probe

A.5 Comparison with Prior Work

A.6 Additional Evaluation Metrics

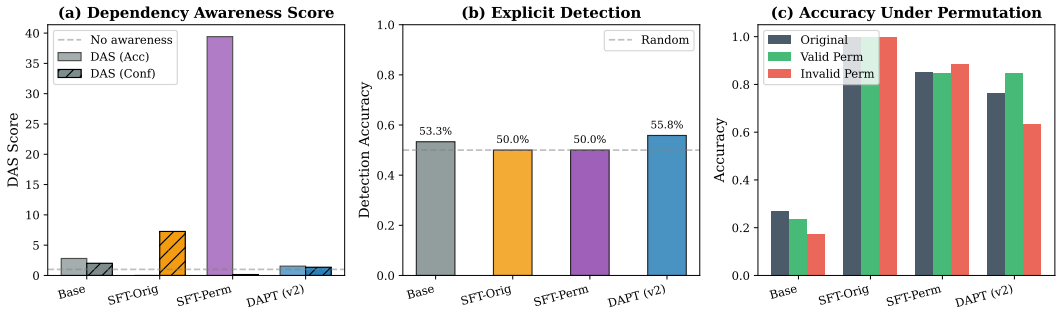


Figure 5: Training method comparison on Qwen2.5-1.5B-Instruct. (a) DAS scores: DAPT maintains $\text{DAS}_{\text{conf}} > 1$ while SFT-Original collapses to $\text{DAS}_{\text{acc}} = 0$ (answer extraction) and SFT-Permute shows $\text{DAS}_{\text{conf}} < 1$ (degraded sensitivity). (b) Detection accuracy: only DAPT improves above chance. (c) Accuracy under permutation: SFT-Original achieves near-100% on all settings (extraction), while DAPT maintains the largest gap between valid and invalid.

Table 5: DAPT scaling analysis: 0.5B vs. 1.5B. DAPT improves DAS_{conf} at both scales, but only the 1.5B model has sufficient capacity to benefit in accuracy and detection.

Model	DAS_{acc}	DAS_{conf}	Detect. Acc.	GSM8K Acc.
0.5B Base	0.56	5.60	50.6%	64.5%
0.5B DAPT	0.84	2.65	51.1%	17.0%
1.5B Base	1.03	2.01	53.3%	51.5%
1.5B DAPT	7.83	1.36	55.8%	65.0%

Table 6: Continuation probe results (200 samples). The model receives the first $k = \lfloor n/2 \rfloor$ steps and must generate the rest. "Paired" columns compare valid vs. invalid permutations on the same problems ($n = 101$), controlling for difficulty. Paired DAS is S_i/S_v where $S_v = |\text{Acc}_{\text{orig}} - \text{Acc}_{\text{valid}}|$ and $S_i = |\text{Acc}_{\text{orig}} - \text{Acc}_{\text{invalid}}|$ on the matched subset; bootstrap 95% CIs in brackets.

	Unpaired Acc.			Paired Acc.		
Model	Orig.	Valid	Invalid	Valid	Invalid	Paired DAS
Base 1.5B	12.0%	8.5%	11.3%	8.9%	7.9%	0.98 [0.0, 4.0]
DAPT 1.5B	62.5%	45.3%	55.4%	46.5%	57.4%	0.60 [0.0, 3.0]

Table 7: Comparison with the most related prior work. "Valid/Invalid" = distinguishes dependency-preserving from dependency-violating permutations. "Ordering-only signal" = preference signal comes exclusively from step order, not content or answer correctness. "Trains discrimination" = explicitly trains models to discriminate valid from invalid orderings.

Work	Models DAG?	Valid/ Invalid?	Ordering-only?	Trains discrim.?	Goal
B. Wang et al. Wang et al. (2023a)	✓	✗	—	✗	Study ordering
Chen et al. Chen et al. (2024)	✗	✗	—	✗	Show brittleness
He et al. He et al. (2025)	✗	✗	✗	✗	Robust to reorder
DAG-Math Zhang et al. (2026)	✓	✗	—	✗	Eval. fidelity
PARC Mukherjee et al. (2025)	✓	✗	—	✗	Error ID
FRODO Paul et al. (2024)	✗	✗	✗	✗	Faithfulness
DeepSeek-R1 Guo et al. (2025)	✗	✗	✗	✗	RL reasoning
Ours (DAPT)	✓	✓	✓	✓	Discriminate

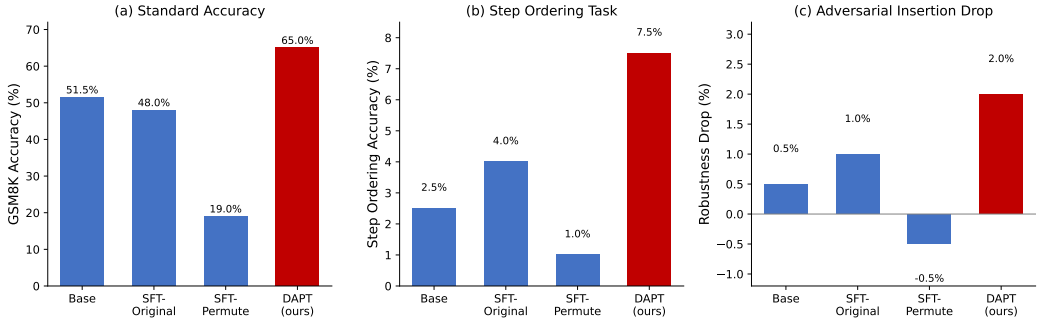


Figure 6: Additional evaluation metrics across training methods. (a) Standard GSM8K accuracy: DAPT achieves the highest (65.0%) among non-extraction models. (b) Step ordering task: DAPT achieves 7.5%, a 3× improvement over the base model. (c) Adversarial insertion drop: SFT models show smaller drops due to extraction-induced high baseline accuracy.