Learning from Mistakes: Negative Reasoning Samples Enhance Out-of-Domain Generalization

Xueyun Tian[♠♡*], **Minghua Ma**[♢*], **Bingbing Xu**[♠†], **Nuoyan Lyu**[♠♡], **Wei Li**
Heng Dong[♣], **Zheng Chu**[♢], **Yuanzhuo Wang**[♠], **Huawei Shen**[♠♡]
[♠]State Key Laboratory of AI Safety, Beijing, 100086
[♠]Institute of Computing Technology, CAS, Beijing, China
[♢]Harbin Institute of Technology, Harbin, China
[♡]University of Chinese Academy of Sciences, Beijing, China
[♣]Tsinghua University, Beijing, China
{tianxueyun23z, xubingbing, lvnuoyan23z, wangyuanzhuo, shenhuawei}@ict.ac.cn
{mhma, zchu}@ir.hit.edu.cn
weili.ucas.ict@gmail.com, drdhxi@gmail.com

Abstract

Supervised fine-tuning (SFT) on chain-of-thought (CoT) trajectories demonstrations is a common approach for enabling reasoning in large language models. Standard practices typically only retain trajectories with correct final answers (*positives*) while ignoring the rest (*negatives*). We argue that this paradigm discards substantial supervision and exacerbates overfitting, limiting out-of-domain (OOD) generalization. Specifically, we surprisingly find that incorporating *negative* trajectories into SFT yields substantial OOD generalization gains over *positive-only* training, as these trajectories often retain valid intermediate reasoning despite incorrect final answers. To understand this effect in depth, we systematically analyze data, training dynamics, and inference behavior, identifying 22 recurring patterns in *negative* chains that serve a dual role: they moderate loss descent to mitigate overfitting during training and boost policy entropy by 35.67% during inference to facilitate exploration. Motivated by these observations, we further propose **Gain-based LOss Weighting** (GLOW), an adaptive, sample-aware scheme that exploits such distinctive training dynamics by rescaling per-sample loss based on inter-epoch progress. Empirically, GLOW efficiently leverages unfiltered trajectories, yielding a 5.51% OOD gain over *positive-only* SFT on Qwen2.5-7B and boosting MMLU from 72.82% to 76.47% as an RL initialization. Code is available at Github[1].

1 Introduction

Recent studies (Yang et al., 2025a; Zelikman et al., 2022; Mukherjee et al., 2023; Shao et al., 2024) have established Supervised Fine-Tuning (SFT) as a foundational post-training component. SFT adapts base models with curated instruction data, often incorporating Chain-of-Thought (CoT) trajectories to enhance reasoning capabilities. The training target typically includes the reasoning trace followed by the final answer, optimized via standard next-token prediction. The resulting model frequently serves as the initialization for subsequent reinforcement learning (RL).

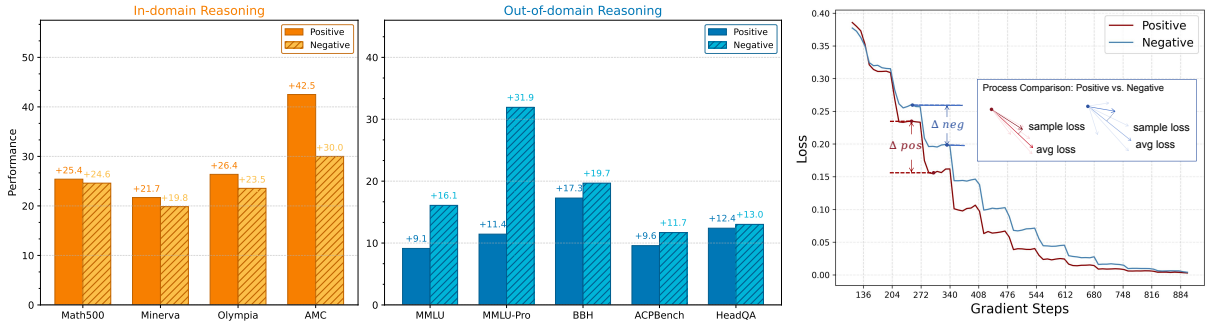
However, existing SFT on distilled CoT trajectories still faces two practical limitations that compromise both effectiveness and efficiency (Luo et al., 2024a; Chu et al., 2025; Gupta et al., 2025; Deb et al., 2025): (i) **Poor generalization**. Models may overfit to domain-specific shortcuts within demonstrations rather than acquiring transferable reasoning skills (Press et al., 2022; Han et al., 2025), leading to limited transferability to out-of-distribution (OOD) tasks. (ii) **Data inefficiency**. Current pipelines typically distill CoT trajectories from a stronger teacher and then apply rejection sampling (Ahn et al., 2024) that retains only *positive* trajectories. This wastes supervision and may discard traces that contain useful intermediate reasoning signals (Hamdan and Yuret, 2025; Luo et al., 2024b; Li et al., 2025).

We argue that these typically discarded *negatives* offer a promising opportunity to alleviate both limitations, as these often include valid intermediate reasoning and diverse modes. To investigate this, we distill math reasoning trajectories from Qwen3-8B (Yang et al., 2025a) and compare student models trained only on *positives* versus only on *negatives*. Figure 1a shows a surprising result: **models trained only on *negatives* outperform those trained only on *positives* on many benchmarks, with larger gains on OOD evaluations.**

This counterintuitive effect motivates a deeper

[*]Equal contribution
[†]Corresponding author
[1]https://github.com/Eureka-Maggie/GLOW

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 29689–29711
July 2-7, 2026



(a) (b)

Figure 1: (a) Qwen2.5-14B: SFT on *positives* improves in-domain math but transfers weakly to other reasoning tasks, whereas SFT on *negatives* yields broader cross-domain gains. Bars show final accuracy, and "+" indicates absolute improvement over the base model. (b) Qwen2.5-32B: training loss on MMLU. Red denotes *positive-only* SFT and blue denotes *negative-only* SFT. Δ is the per-sample inter-epoch loss difference.

analysis of *negatives* across data, optimization, and inference. **Regarding data**, we identify 9 error types with 22 recurring patterns (Table 3). This diversity exposes the model to broad error regimes, fostering intrinsic reasoning signals that generalize across contexts. **In terms of optimization**, *negative-only* SFT shows slower convergence yet steady performance gains across epochs (Figure 1b, Table 10). The consistently smaller inter-epoch loss reduction (Δ) implies a more challenging optimization landscape that resists rapid convergence, thereby mitigating shortcut overfitting and compelling the model to learn robust reasoning features rather than spurious correlations. **For inference**, training on *negatives* significantly boosts policy entropy and pass@k on OOD tasks (Appendix A.9), thereby facilitating diverse exploration and enhancing generalization, respectively. **Overall, these insights reveal a cohesive mechanism: the diverse patterns in *negatives* act as a natural regularizer that modulates training dynamics to prevent shortcut learning while increasing inference entropy to encourage exploration, collectively unlocking superior OOD generalization.**

Motivated by these observations, we seek to synergize the strengths of *positive* and *negative* trajectories within the SFT framework. To achieve this, we propose **G**ain-based **LO**ss **W**eighting (GLOW), a dynamic reweighting scheme utilizing the entire dataset to maximize sample efficiency without explicit filtering. During training, GLOW measures each sample's gain as its inter-epoch loss reduction and adaptively upweights those with low gain. Such samples, typically aligning with the *negatives* with small Δ, signal insufficient learning and steer optimization toward undercovered reasoning patterns. Empirically, GLOW yields consistent gains across model families and scales: on Qwen2.5-7B, it improves average performance by 2.14% over mixed-data training and OOD performance by 5.51% over *positive-only* SFT, and as an RL initialization, it further boosts MMLU from 72.82% to 76.47% under the same RL setup (Table 9).

Our contributions can be summarized as follows:

- **Systematic investigation of *negatives*:** We demonstrate negative trajectories significantly enhance OOD generalization. A unified analysis across data, optimization, and inference reveals that exposure to diverse error patterns mitigates overfitting and fosters exploration.
- **Adaptive Training Strategy:** We propose a sample-aware reweighting strategy for utilizing unfiltered data. By modulating loss based on inter-epoch learning progress, GLOW prioritizes underexplored patterns, enabling efficient and generalizable SFT.
- **Superior SFT Generalization and RL Initialization:** Experiments validate GLOW across diverse benchmarks. It yields consistent OOD improvements and transfers effectively to RL, serving as a superior initialization that amplifies the gains from RL.

2 Related Works

Supervised Fine-Tuning for Reasoning SFT is a widely adopted approach for strengthening the reasoning ability of large language models (Wei et al., 2021; Ouyang et al., 2022). A common recipe is that we distill CoT trajectories from stronger teacher models and use them to supervise smaller or less capable students (Shao et al., 2024;

Zheng et al., 2025; Yu et al., 2025b). To ensure data quality, conventional pipelines often employ rejection sampling as a rigorous filter, retaining only those trajectories that yield correct final answers (Ahn et al., 2024). Such CoT-based SFT can transfer long-form reasoning patterns and often provides a strong initialization for subsequent reinforcement learning (Lewkowycz et al., 2022; Shao et al., 2024). However, this heavy filtering discards a substantial portion of available trajectories, wastefully discarding rich supervisory information.

Learning from Negative Data Prior work leverages negative samples mainly in three ways: prompting, fine-tuning, and reinforcement learning. Prompt-based methods place negative examples in the context to steer generation away from undesired behaviors (Gao and Das, 2024; Alazraki et al., 2025). Their effectiveness, however, depends on the model's existing instruction-following and reasoning ability, which limits their impact on weak students. Fine-tuning-based approaches use negative data more indirectly. A common strategy is to convert initially incorrect trajectories into positive CoT supervision via teacher rewriting or refinement (Yu et al., 2025a; Pan et al., 2025; An et al., 2023). Other works add explicit markers or prefixes to separate correct from incorrect samples during training (Wang et al., 2024a; Tong et al., 2024). These methods do not establish whether learning from raw incorrect trajectories themselves improves generalization.

Domain Generalization in LLMs Most fine-tuning work improves reasoning within a single domain, such as mathematics or code, while cross-domain transfer remains underexplored. Huan et al. (2025) show that SFT on math induces substantial representation shifts that can degrade general capabilities. Wu et al. (2025) propose knowledge index and information gain to separate knowledge from reasoning, and find that SFT on math offers limited benefit in knowledge-intensive domains. Yang et al. (2025b) and Zhao et al. (2025) further argue that SFT often learns superficial reasoning traces and transfers poorly across domains. These studies are primarily diagnostic and do not develop methods or examine how data selection and supervision signals affect cross-domain generalization.

3 The Surprising Phenomenon: Negatives Generalize Better

In this section, we describe the empirical phenomenon that motivates our study: fine-tuning on negative reasoning samples can enhance OOD generalization more effectively than fine-tuning on *positive* samples. We first detail the controlled experiments designed to validate this phenomenon and then present results that demonstrate its consistency across diverse benchmarks and model scales.

3.1 Data Construction and Training Setup

Using Qwen3-8B, we distill trajectories from OpenMathReasoning (Moshkov et al., 2025) and MMLU (Hendrycks et al., 2021b), labeling those matching the ground truth as *positive* and others as *negative*. We construct balanced datasets of complete reasoning chains to fine-tune Qwen-2.5 (from 3B to 32B) and Llama-3.1 8B. See Appendix A.1 for detailed configurations.

3.2 Negatives Surpass Positives in OOD

As shown in Table 1 and Table 2, we surprisingly find that training on *negative* samples, although it yields smaller improvements than *positive* samples on in-domain performance, consistently improves OOD generalization. Overall, models trained on *negative* math reasoning samples achieve an average improvement of 11.97% on general reasoning tasks and 4.11% on other reasoning tasks. Similarly, models trained on *negative* MMLU samples gain an average of 1.98% on mathematical reasoning and 1.35% on other reasoning benchmarks. Although mathematical problems are generally more suitable for constructing reasoning-focused data, the same trend is observed for models trained on MMLU, indicating that the benefit of *negative* samples for OOD generalization is not limited to a specific domain. These observations motivate a deeper analysis into the underlying factors that make *negative* samples more effective for enhancing OOD reasoning performance.

4 Why Negative is Better

To explain why *negatives* benefit OOD generalization, we analyze the phenomenon from data, optimization, and inference perspectives. Empirically, *positives* tend to share a small set of success pattern, while *negatives* exhibit much richer failure modes. We first characterize the diversity introduced by *negatives*. We then examine training dynamics to

		Math Reasoning (In-Domain)					General Reasoning (Out-of-Domain)				Other Reasoning (Out-of-Domain)		
Model	Setting	Math500	Minerva	Olympia	AMC	Avg.	MMLU	MMLU-Pro	BBH	Avg.	ACPBench	HeadQA	Avg.
Qwen2.5-3B	Base	52.60	21.32	22.52	32.50	32.24	31.88	12.54	27.75	24.06	23.31	33.15	28.23
	Full	60.80	26.10	23.26	35.00	36.29	64.13	38.66	52.29	51.69	32.68	62.69	47.69
	Positive	61.60	25.74	24.44	42.50	38.57	54.45	25.62	44.35	41.47	30.21	59.81	45.01
	Negative	58.60	23.53	24.15	42.50	37.20	64.09	39.20	53.87	52.39	33.06	63.13	48.10
	Δ(pos-neg)	+3.00	+2.21	+0.29	0.00	+1.38	-9.64	-13.58	-9.52	-10.91	-2.85	-3.32	-3.09
Qwen2.5-7B	Base	58.40	26.84	26.07	52.50	40.95	55.80	26.56	51.10	44.49	28.77	57.29	43.03
	Full	76.60	40.07	38.96	55.00	52.66	72.24	53.71	70.84	65.60	38.27	72.06	55.17
	Positive	78.00	36.76	41.78	57.50	53.51	61.03	32.70	60.58	51.44	33.38	68.60	50.99
	Negative	77.60	40.44	38.37	57.50	53.48	73.11	53.74	71.73	66.19	38.98	71.81	55.40
	Δ(pos-neg)	+0.40	-3.68	+3.41	0.00	+0.03	-12.08	-21.04	-11.15	-14.76	-5.60	-3.21	-4.41
Qwen2.5-14B	Base	62.60	26.84	27.56	40.00	39.25	64.68	35.77	59.27	53.24	37.04	68.75	52.90
	Full	86.80	47.79	52.30	82.50	67.35	81.56	67.63	80.90	76.70	48.13	81.44	64.79
	Positive	88.00	48.53	53.93	82.50	68.24	73.81	47.21	76.54	65.85	46.62	81.15	63.89
	Negative	87.20	46.69	51.11	70.00	63.75	80.77	67.70	78.95	75.81	48.73	81.77	65.25
	Δ(pos-neg)	+0.80	+1.84	+2.82	+12.50	+4.49	-6.96	-20.49	-2.41	-9.95	-2.11	-0.62	-1.37
Qwen2.5-32B	Base	63.20	34.19	26.52	35.00	39.73	68.34	39.80	58.65	55.60	38.63	68.45	53.54
	Full	92.20	52.57	57.19	85.00	71.74	85.22	73.10	83.53	80.62	50.67	84.90	67.79
	Positive	91.40	50.74	60.89	85.00	72.01	79.01	54.31	80.61	71.31	49.96	83.15	66.56
	Negative	92.20	50.74	58.37	95.00	74.08	85.47	73.53	84.51	81.17	51.80	85.27	68.54
	Δ(pos-neg)	-0.80	0.00	+2.52	-10.00	-2.07	-6.46	-19.22	-3.90	-9.86	-1.84	-2.12	-1.98
Llama3.1-8B	Base	2.80	1.10	0.44	0.00	1.09	66.49	0.47	2.33	23.10	5.18	2.30	3.74
	Full	41.20	18.01	14.67	15.00	22.22	62.48	36.88	55.12	51.49	32.96	65.90	49.43
	Positive	37.80	18.01	10.37	12.50	19.67	41.95	23.15	45.07	36.72	31.20	47.81	39.51
	Negative	34.40	18.38	9.19	20.00	20.49	62.14	36.22	54.85	51.07	33.31	65.17	49.24
	Δ(pos-neg)	+3.40	-0.37	+1.18	-7.50	-0.82	-20.19	-13.07	-9.78	-14.35	-2.11	-17.36	-9.74

Table 1: Cross-domain performance on **math reasoning**. "Avg." is the within-group average. orange highlights in-domain benchmarks where *positives* outperform *negatives*, and blue highlights OOD benchmarks where *negatives* outperform *positives*. The higher score in each pair is highlighted accordingly.

		Math Reasoning (Out-of-Domain)					General Reasoning (In-Domain)				Other Reasoning (Out-of-Domain)		
Model	Setting	Math500	Minerva	Olympia	AMC	Avg.	MMLU	MMLU-Pro	BBH	Avg.	ACPBench	HeadQA	Avg.
Qwen2.5-3B	Base	52.60	21.32	22.52	32.50	32.24	31.88	12.54	27.75	24.06	23.31	33.15	28.23
	Full	58.20	23.16	25.19	35.00	35.39	66.74	40.82	53.35	53.64	35.70	67.61	51.66
	Positive	59.20	27.21	25.04	30.00	35.36	67.88	42.56	52.84	54.43	34.93	67.69	51.31
	Negative	59.60	28.31	25.48	40.00	38.35	65.42	38.55	52.28	52.08	36.13	68.85	52.49
	Δ(pos-neg)	-0.40	-1.10	-0.44	-10.00	-2.99	+2.46	+4.01	+0.56	+2.34	-1.20	-1.16	-1.18
Qwen2.5-7B	Base	58.40	26.84	26.07	52.50	40.95	55.80	26.56	51.10	44.49	28.77	57.29	43.03
	Full	75.60	38.60	40.15	47.50	50.46	73.14	51.15	71.30	65.20	42.18	72.76	57.47
	Positive	74.40	37.50	39.85	50.00	50.44	73.42	53.22	68.23	64.96	40.32	74.25	57.29
	Negative	77.00	37.13	42.07	60.00	54.05	71.23	45.79	69.46	62.16	42.61	73.38	58.00
	Δ(pos-neg)	-2.60	+0.37	-2.22	-10.00	-3.61	+2.19	+7.43	-1.23	+2.80	-2.29	+0.87	-0.71
Qwen2.5-14B	Base	62.60	26.84	27.56	40.00	39.25	64.68	35.77	59.27	53.24	37.04	68.75	52.90
	Full	82.20	43.01	51.85	70.00	61.77	78.13	59.57	80.56	72.75	48.87	79.94	64.41
	Positive	80.20	42.28	50.96	72.50	61.49	80.09	65.26	80.21	75.19	48.56	80.53	64.55
	Negative	83.00	45.22	48.89	65.00	60.53	76.83	56.03	80.15	71.00	48.27	80.56	64.42
	Δ(pos-neg)	-2.80	-2.94	+2.07	+7.50	+0.96	+3.26	+9.23	+0.06	+4.18	+0.29	-0.03	+0.13
Qwen2.5-32B	Base	63.20	34.19	26.52	35.00	39.73	68.34	39.80	58.65	55.60	38.63	68.45	53.54
	Full	86.60	46.69	55.70	80.00	67.25	79.06	61.15	79.94	73.38	49.89	83.01	66.45
	Positive	85.20	46.69	56.15	75.00	65.76	81.97	68.54	81.60	77.37	50.35	82.90	66.63
	Negative	86.40	47.06	56.89	72.50	65.71	77.99	58.34	80.71	72.35	51.20	82.39	66.80
	Δ(pos-neg)	-1.20	-0.37	-0.74	+2.50	+0.05	+3.98	+10.20	+0.89	+5.02	-0.85	+0.51	-0.17
Llama3.1-8B	Base	2.80	1.10	0.44	0.00	1.09	66.49	0.47	2.33	23.10	5.18	2.30	3.74
	Full	20.00	15.81	6.52	2.50	11.21	66.49	40.56	53.73	53.59	36.06	69.55	52.81
	Positive	15.60	11.76	3.85	7.50	9.68	64.73	39.74	45.39	49.95	29.61	67.69	48.65
	Negative	23.00	16.18	6.67	10.00	13.96	64.63	38.85	53.23	52.24	37.15	69.80	53.48
	Δ(pos-neg)	-7.40	-4.42	-2.82	-2.50	-4.29	+0.10	+0.89	-7.84	-2.28	-7.54	-2.11	-4.83

Table 2: Cross-domain performance on **general reasoning**. "Avg." is the within-group average. orange highlights in-domain benchmarks where *positives* outperform *negatives*, and blue highlights OOD benchmarks where *negatives* outperform *positives*. The higher score in each pair is highlighted accordingly.

Error Categories	OpenMathReasoning	MMLU
Calculation	27	9
Completeness	11	28
Evaluation System	2599	2024
Formal	57	123
Knowledge	27	199
Logical	195	4116
Programming	8	5
Understanding	435	1056
Special Cases	301	1137
Total	**3660**	**8697**

Table 3: Error categorization in the negative OpenMathReasoning and MMLU samples.

show how this diversity shapes optimization. Finally, we analyze inference behavior to connect these effects to improved OOD performance.

4.1 Data Perspective

Following (He et al., 2025), we observe that reasoning errors manifest in 9 major types and 22 subtypes. For each *negative* trajectory in OpenMathReasoning and the MMLU training set, we use Gemini-2.5-Pro (Comanici et al., 2025) to assign an error label (the prompt is in Appendix A.11). Table 3 shows a broad and diverse distribution that spans logical mistakes, comprehension errors, and other failure modes. This diversity implies that *negatives* cover substantially more heterogeneous reasoning patterns than *positives*, which tend to follow more uniform solution templates. The full label definitions are provided in Appendix A.4.

***Negatives* improve OOD generalization by exposing the model to diverse error regimes, which encourages invariant reasoning features.** We view error categories in *negatives* as environments in the sense of IRM (Arjovsky et al., 2019) (formalized in Appendix A.3), where generalization benefits from signals that remain stable across heterogeneous environments. Each error type defines a distinct failure regime. *Negatives* are not pure noise, since many trajectories contain partially valid reasoning segments (Figure 10), and performance continues to improve over epochs when training on *negatives* (Table 10). This diversity compels the model to learn invariant features stable across distinct regimes, whereas positives cover fewer paths and offer weaker incentives for such stability.

4.2 Training Perspective

To characterize learning dynamics, we log training loss every 10 steps for models fine-tuned on *positives* and *negatives* from math reasoning and MMLU. We use Qwen2.5-32B as a representative example (Figure 1b) with additional training curves are deferred to Appendix A.6. Across settings, the loss exhibits a consistent stage-wise pattern. With *positives*, loss drops abruptly near epoch boundaries and converges faster early on. With *negatives*, loss decreases more smoothly and gradually, yet converges to a comparable level.

We attribute the loss disparity to signal diversity: homogeneous *positives* drive rapid early drops via redundant updates, whereas heterogeneous *negatives* induce steadier, broader progress. Table 4 confirms this early gap ($\Delta_{\text{pos}} > \Delta_{\text{neg}}$). This sustained descent reflects reduced shortcut fitting, aligning with the superior OOD generalization of *negatives*.

Importantly, loss on *negatives* keeps decreasing throughout training (Figures 1b and 5) and is accompanied by steady gains on benchmarks at multiple training checkpoints (Table 10 and Appendix A.7). This indicates that *negatives* provide learnable supervision rather than noise. They combine partially valid reasoning with diverse failure patterns, yielding sustained training signals and promoting robust reasoning over memorizing narrow solution templates.

Overall, these results indicate that the training value of *negatives* lies in their diversity: they slow early loss descent while providing heterogeneous optimization signals that broaden reasoning patterns and improve OOD generalization.

Model	$\Delta^{\text{epoch } 2-1}_{\text{avg\_loss}}$	$\Delta^{\text{epoch } 3-2}_{\text{avg\_loss}}$	$\Delta^{\text{epoch } 4-3}_{\text{avg\_loss}}$	$\Delta^{\text{epoch } 5-4}_{\text{avg\_loss}}$
Qwen2.5-3B	0.014957	0.013486	0.015686	0.014000
Qwen2.5-7B	0.009729	0.022514	0.014172	0.001156
Qwen2.5-14B	0.008515	0.017786	0.011157	0.005472
Qwen2.5-32B	0.007143	0.018200	0.015557	0.003772
Llama3.1-8B	0.015586	0.023344	0.005571	0.004915

Table 4: Comparison of per-epoch loss drops under *positive-only* and *negative-only* SFT on MMLU. Each entry reports $\Delta_{\text{pos}} - \Delta_{\text{neg}}$, where Δ is the average loss decrease within an epoch. Interpretation focuses on relative differences across epochs.

4.3 Inference Perspective

We examine how *negative* supervision changes inference behavior. We use token-level policy entropy as a proxy for uncertainty and exploration during reasoning, and therefore interpret it cautiously together with complementary evidence. Let M_{pos} be the model fine-tuned on *positives* from OpenMathReasoning, and M_{neg} be the model fine-tuned on *negatives*. To evaluate both in-domain and OOD behavior, we distill reference trajecto-

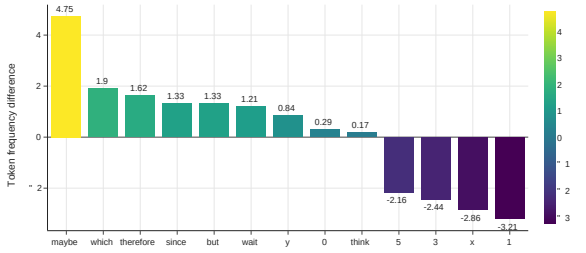


Figure 2: Token frequency differences between M_{neg} and M_{pos} on digits and high-entropy tokens.

Model	Setting	Data	$\bar{H}_{\text{think}}$	$\bar{H}_{\text{ans}}$	ΔH
M_{pos}	Off-policy	Math	0.909	0.708	0.202
		Other	1.138	0.873	0.265
	On-policy	Math	0.753	0.601	0.153
		Other	0.669	0.757	-0.088
M_{neg}	Off-policy	Math	1.212	0.883	0.329
		Other	1.427	0.992	0.435
	On-policy	Math	1.011	0.772	0.239
		Other	0.917	0.783	0.134

Table 5: Policy entropy analysis on M_{pos} and M_{neg}.

ries from Qwen3-8B on an in-domain math set ("Math") and an OOD set ("Other"). We define the thinking span as tokens between <think> and </think>, and the answer span as tokens after </think>. We compute entropy under two protocols. **Off-policy** evaluates entropy along the teacher trajectory (teacher forcing). **On-policy** evaluates entropy along the model's own generated trajectory under a fixed decoding rule. Entropy is computed from raw logits with $T{=}1$ and includes special boundary tokens.

Formally, let $\mathcal{V}$ be the vocabulary and θ the model parameters. The token-level entropy at step t is

$$p_t(v) \triangleq p_\theta(v \mid x, y_{<t}),$$
$$H_t(\theta \mid x, y_{<t}) = -\sum_{v \in \mathcal{V}} p_t(v) \log p_t(v). \quad (1)$$

where $p_\theta(\cdot \mid x, y_{<t})$ is the softmax distribution induced by the pre-softmax logits. For sample i, let $\mathcal{T}^{(i)}$think and $\mathcal{T}^{(i)}$ans denote token indices in the thinking and answer spans, determined by the teacher trajectory (off-policy) or the model trajectory (on-policy). We report mean span entropy:

$$\bar{H}^{(i)}_{\text{think}} = \frac{1}{\left|\mathcal{T}^{(i)}_{\text{think}}\right|} \sum_{t \in \mathcal{T}^{(i)}_{\text{think}}} H_t,$$
$$\bar{H}^{(i)}_{\text{ans}} = \frac{1}{\left|\mathcal{T}^{(i)}_{\text{ans}}\right|} \sum_{t \in \mathcal{T}^{(i)}_{\text{ans}}} H_t. \quad (2)$$

and the boundary drop:

$$\Delta H^{(i)} = \bar{H}^{(i)}_{\text{think}} - \bar{H}^{(i)}_{\text{ans}}. \quad (3)$$

As presented in Table 5, M_{neg} maintains higher entropy throughout the thinking span and displays a sharper decline at the answer boundary. We view this pattern as suggestive of broader exploration followed by firmer commitment, rather than as standalone proof of a specific internal mechanism. Regarding baselines, off-policy entropy is inherently higher because teacher forcing exposes the model to contexts that can have low probability under its own policy. Under distribution shift, however, the models diverge: while M_{neg} remains relatively stable, the on-policy OOD entropy margin of M_{pos} can even reverse. We therefore interpret this reversal as suggestive of over-specialization to in-domain templates, not as conclusive evidence on its own.

We further localize where uncertainty concentrates. Figure 2 compares high-entropy token usage in generated trajectories. Relative to M_{pos}, M_{neg} produces more discourse and hesitation tokens (e.g., "maybe," "wait," "but") and fewer numerals, a token-level pattern consistent with more budget allocated to connective exploration before concrete computation. Figure 11 illustrates the same effect qualitatively. Appendix A.10 provides a complementary length analysis under identical decoding settings: negative-trained models mainly expand the <think> span, while keeping the answer span comparable or slightly shorter. Taken together, these aligned indicators are consistent with the interpretation that M_{neg} maintains more plausible continuations and explores more reasoning paths before settling on an answer.

Overall, these results suggest that *negative*-based supervision is associated with more exploratory reasoning and sharper final commitment at inference time, a pattern that is consistent with its stronger cross-domain generalization.

		Math Reasoning (In-Domain)					General Reasoning (Out-of-Domain)				Other Reasoning (Out-of-Domain)		
Model	Setting	Math500	Minerva	Olympia	AMC	Avg.	MMLU	MMLU-Pro	BBH	Avg.	ACPBench	HeadQA	Avg.
Qwen2.5-3B	Full	60.80	26.10	23.26	35.00	36.29	64.13	**38.66**	52.29	51.69	32.68	62.69	47.69
	GLOW	**62.80**	**27.21**	**24.30**	**42.50**	**39.20**	**64.49**	38.63	**53.20**	**52.11**	**33.66**	**63.38**	**48.52**
Qwen2.5-7B	Full	76.60	40.07	38.96	55.00	52.66	72.24	53.71	70.84	65.60	38.27	72.06	55.17
	GLOW	**79.60**	**40.07**	**41.04**	**60.00**	**55.18**	**73.99**	**55.77**	**71.99**	**67.25**	**39.19**	**72.50**	**55.85**
Qwen2.5-14B	Full	86.80	47.79	52.30	82.50	67.35	81.56	67.63	80.90	76.70	48.13	81.44	64.79
	GLOW	**87.80**	**52.21**	**52.44**	**82.50**	**68.74**	**82.53**	**68.70**	**81.65**	**77.63**	**49.51**	**82.35**	**65.93**
Qwen2.5-32B	Full	92.20	52.57	57.19	85.00	71.74	85.22	73.10	83.53	80.62	50.67	84.90	67.79
	GLOW	**93.40**	**54.41**	**59.11**	**92.50**	**74.86**	**85.51**	**74.14**	**83.98**	**81.21**	**51.97**	**85.19**	**68.58**
Llama3.1-8B	Full	41.20	18.01	14.67	15.00	22.22	62.48	36.88	55.12	51.49	32.96	65.90	49.43
	GLOW	**44.60**	**20.59**	**15.11**	**17.50**	**24.45**	**63.80**	**38.34**	**58.17**	**53.44**	**35.04**	**66.70**	**50.87**

Table 6: Cross-domain performance of models trained on the **math reasoning** dataset. "Avg." denotes the average score within each group. **Bold** indicates the best results under the same model.

		Math Reasoning (Out-of-Domain)					General Reasoning (In-Domain)				Other Reasoning (Out-of-Domain)		
Model	Setting	Math500	Minerva	Olympia	AMC	Avg.	MMLU	MMLU-Pro	BBH	Avg.	ACPBench	HeadQA	Avg.
Qwen2.5-3B	Full	58.20	23.16	25.19	35.00	35.39	66.74	40.82	**53.35**	53.64	35.70	67.61	51.66
	GLOW	**61.40**	**29.41**	**25.78**	**40.00**	**39.15**	**67.09**	**41.27**	52.61	**53.66**	**36.20**	**69.15**	**52.68**
Qwen2.5-7B	Full	75.60	38.60	40.15	47.50	50.46	73.14	**51.15**	71.30	65.20	42.18	72.76	57.47
	GLOW	**78.20**	**41.18**	**43.70**	**60.00**	**55.77**	**74.51**	51.13	**71.99**	**65.88**	**43.56**	**75.35**	**59.46**
Qwen2.5-14B	Full	82.20	43.01	51.85	70.00	61.77	78.13	59.57	80.56	72.75	48.87	79.94	64.41
	GLOW	**85.00**	**48.09**	**54.22**	**70.00**	**64.33**	**79.97**	**62.78**	**82.32**	**75.02**	**50.95**	**82.20**	**66.58**
Qwen2.5-32B	Full	86.60	46.69	55.70	80.00	67.25	79.06	61.15	79.94	73.38	49.89	83.01	66.45
	GLOW	**89.00**	**47.06**	**58.67**	**82.50**	**69.31**	**80.81**	**64.72**	**81.98**	**75.84**	**52.08**	**83.73**	**67.91**
Llama3.1-8B	Full	20.00	15.81	6.52	2.50	11.21	66.49	40.56	53.73	53.59	36.06	69.55	52.81
	GLOW	**24.80**	**20.59**	**6.96**	**12.50**	**16.21**	**68.52**	**42.96**	**57.53**	**56.33**	**39.72**	**72.57**	**56.15**

Table 7: Cross-domain performance of models trained on the **general reasoning** dataset. "Avg." denotes the average score within each group. **Bold** indicates the best results under the same model.

5 From Negatives to Effective Full-Data Training

In this section, we move beyond the empirical finding that *negatives* improve OOD generalization. Training on *negatives* alone remains a rejection-based strategy and still fails to use supervision efficiently. Our goal is to improve both in-domain and OOD performance while using data more effectively. We therefore target the training objective and propose a simple mechanism that adapts sample weights based on learning progress.

5.1 GLOW: Gain-Based Loss Weighting

Our analysis suggests that *negatives* help by injecting optimization diversity, which broadens the learned reasoning space. This motivates reweighting SFT toward undercovered patterns. GLOW quantifies each sample's gain by its inter-epoch loss reduction. A small gain indicates limited effective coverage under the current trajectory. GLOW then upweights such samples via an adaptive scaling function, steering updates toward complementary directions and improving generalization.

Let $\ell_i^{(t)}$ denote the loss of sample i at epoch t. We quantify a sample's learning progress as its inter-epoch loss reduction: $\Delta_i^{(t)} = \ell_i^{(t-1)} - \ell_i^{(t)}$. A small $\Delta_i^{(t)}$ indicates that the sample remains insufficiently learned and may encode underrepresented patterns, whereas a large $\Delta_i^{(t)}$ suggests diminishing marginal utility. We therefore upweight small-Δ samples via

$$w_i^{(t)} = \alpha\big(1 - \sigma\left(\beta \Delta_i^{(t)}\right)\big), \tag{4}$$

where $\sigma(\cdot)$ is the sigmoid function and α, β are scaling hyperparameters. For the first epoch, we set $w_i^{(1)} = 1$ for all samples. The reweighted objective is

$$\mathcal{L}_{\text{GLOW}}^{(t)}(\theta) = \sum_{i=1}^{N} w_i^{(t)}\, \ell_i(\theta). \tag{5}$$

Why it works. The inter-epoch loss reduction $\Delta_i^{(t)}$ measures how much sample i benefits from recent updates. Under standard L-smoothness, for an update $\theta' = \theta - \eta G^{(t)}$ with $G^{(t)} = \frac{1}{N}\sum_j w_j^{(t)} \nabla \ell_j(\theta)$, a first-order expansion gives

$$\Delta_i^{(t)} \approx \ell_i(\theta) - \ell_i(\theta') \approx \eta \big\langle \nabla \ell_i(\theta),\, G^{(t)} \big\rangle,$$

so $\Delta_i^{(t)}$ is closely tied to the alignment between the current update direction and the sample gradient. Thus, small $\Delta_i^{(t)}$ indicates patterns weakly covered by optimization, whereas large $\Delta_i^{(t)}$ suggests diminishing marginal utility. Since $w_i^{(t)} = \alpha\Big(1 - \sigma(\beta\Delta_i^{(t)})\Big)$ is decreasing in $\Delta_i^{(t)}$, GLOW prioritizes small-Δ samples to steer training toward complementary directions, increasing gradient diversity and exploration to improve generalization. See Appendix A.2 for detailed derivation.

5.2 Discussion with Prior Works

Parallel to our focus on negative trajectories, recent reasoning-oriented RL approaches also leverage negative signals, yet primarily to penalize undesired behaviors via reward structuring and credit assignment (Zhu et al., 2025; Liu et al., 2025; Yang et al., 2025c; Nan et al., 2025). In contrast, GLOW investigates *negatives* within the SFT stage and establishes them as a source of direct supervision: rather than merely being suppressed, *negatives* are utilized to broaden the reasoning space, thereby enhancing OOD generalization.

Regarding objective design, prior SFT reweighting typically targets optimization imbalance by utilizing current loss to down-weight easy samples, a process that is effectively memoryless (Lin et al., 2017; Bengio et al., 2009). In contrast, GLOW targets coverage: it upweights samples exhibiting stagnant progress, directing optimization toward underexplored reasoning patterns.

5.3 Experimental Results

Building on the theoretical analysis, we empirically validate the effectiveness of GLOW in the SFT stage. All other experimental settings are the same as 3.1 and details are described in Appendix A.1.

GLOW improves cross-domain generalization without sample filtering. We apply GLOW to a randomly shuffled mixture of *positives* and *negatives* and observe consistent gains across domains and model scales. For brevity, we report only standard SFT on the mixed data and GLOW. Results for *positive-only* and *negative-only* SFT are provided in Tables 1 and 2. Table 6 demonstrates that GLOW consistently enhances in-domain performance across all model scales and achieves superior OOD results. Specifically, for Qwen2.5-7B, GLOW reaches 55.18 in-domain and 67.25 OOD while maintaining competitive general reasoning abilities. Similar gains are observed in models trained on general reasoning data. Further results in Table 7 show that on Qwen2.5-14B, GLOW boosts OOD math and reasoning performance by 2.56 and 2.17 points, respectively. Overall, GLOW maximizes data utilization by learning from all trajectories, yielding robust gains across diverse benchmarks and settings. We also do the ablation of hyperparameters in Appendix A.5.

Setting	Train	Test	$\bar{H}_{\text{think}}$	$\bar{H}_{\text{ans}}$	ΔH
Full	Math	Math	0.36	0.22	0.14
		Other	1.24	1.38	-0.14
	MMLU	Math	0.54	0.34	0.20
		Other	0.96	0.98	-0.02
GLOW	Math	Math	0.71	0.35	0.36
		Other	1.52	1.30	0.22
	MMLU	Math	0.89	0.52	0.37
		Other	1.44	1.21	0.23

Table 8: Policy entropy changes with and without GLOW under various settings.

Setting	Math500	Minerva	AMC	MMLU	MMLU-Pro
SFT	76.60	40.07	55.00	72.24	53.71
GLOW	79.60	40.07	**60.00**	73.99	55.77
SFT + RL	78.20	40.24	52.50	72.82	53.95
GLOW + RL	**80.20**	**42.28**	57.50	**76.47**	**57.37**

Table 9: Controlled comparison of post-training on GSM8K for Qwen2.5-7B-base. GLOW improves OOD metrics both before and after RL, indicating a stronger initialization for RL post-training.

GLOW tends to upweight *negatives*. We further examine the samples prioritized by GLOW. Appendix A.8 reveals that GLOW prioritizes *negatives* during early training stages, suggesting that gain-based weights primarily target harder and underrepresented reasoning patterns.

GLOW encourages exploration while preserving answer commitment. Table 8 shows GLOW consistently increases thinking span entropy across settings (e.g., 0.36 to 0.71 from Math to Math, and 0.96 to 1.44 from MMLU to Other domains), while answer span entropy remains stable or decreases under OOD. This suggests GLOW encourages broader exploration during reasoning while keeping answers relatively decisive, consistent with its generalization gains.

GLOW serves as a superior initialization for subsequent RL training. Starting from Qwen2.5-7B-base, we train on GSM8K with four settings: (i) standard SFT, (ii) standard SFT followed by RL post-training, (iii) SFT with GLOW, and (iv) GLOW-based SFT followed by the same

RL post-training. Using GRPO (Shao et al., 2024) with fixed RL data, optimizer, and hyperparameters, we vary only the SFT objective to isolate its impact. Table 9 shows that GLOW improves OOD performance before RL and remains superior after RL, outperforming the RL model initialized from standard SFT. This indicates GLOW yields stronger SFT initialization that transfers to RL post-training.

6 Conclusion

We show that negative reasoning trajectories can improve SFT generalization and mitigate OOD degradation. Through data, training, and inference analyses, we identify why *negatives* help and how they shape optimization and model behavior. Building on these findings, we propose Gain-based LOss Weighting (GLOW), which upweights undercovered examples using inter-epoch loss reduction, yielding more data-efficient training and consistent cross-domain gains across diverse benchmarks.

Limitations

Our study primarily examines gain-based reweighting in the supervised fine-tuning stage of reasoning post-training, and we leave its interaction with subsequent RLHF or other reinforcement learning stages as an exciting direction for future work. In addition, our experiments focus on text-only chain-of-thought data for math and multi-task knowledge benchmarks with a small set of open-source backbones, so a natural next step is to extend the same analysis and method to broader task families, larger model scales and multimodal or tool-augmented settings, building on the phenomena and gains established in this work.

Acknowledgments

This research was funded by the Key Research and Development Project of Henan Province (No. 241111211900), the Strategic Priority Research Program of the CAS under Grant (No. XDB0680302), the Director's Fund Project of State Key Laboratory of AI Safety, and the Young Elite Scientists Sponsorship Program of the Beijing High Innovation Plan (No. 20250924).

References

Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. 2024. Large language models for mathematical reasoning: Progresses and challenges. arXiv preprint arXiv:2402.00157.

Lisa Alazraki, Maximilian Mozes, Jon Ander Campos, Tan Yi-Chern, Marek Rei, and Max Bartolo. 2025. No need for explanations: Llms can implicitly learn from mistakes in-context. arXiv preprint arXiv:2502.08550.

Shengnan An, Zexiong Ma, Zeqi Lin, Nanning Zheng, Jian-Guang Lou, and Weizhu Chen. 2023. Learning from mistakes makes llm better reasoner. arXiv preprint arXiv:2310.20689.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. 2019. Invariant risk minimization. arXiv preprint arXiv:1907.02893.

Art of Problem Solving Foundation. 2023. Amc23 — 2023 american mathematics competitions test set. 40 problems drawn from the 2023 AMC 12 contests.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In Proceedings of the 26th annual international conference on machine learning, pages 41–48.

Olivier Bousquet and André Elisseeff. 2002. Stability and generalization. Journal of machine learning research, 2(Mar):499–526.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. 2025. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. arXiv preprint arXiv:2501.17161.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. arXiv preprint arXiv:2507.06261.

Rohan Deb, Kiran Thekumparampil, Kousha Kalantari, Gaurush Hiranandani, Shoham Sabach, and Branislav Kveton. 2025. Fishersft: Data-efficient supervised fine-tuning of language models using information gain. arXiv preprint arXiv:2505.14826.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. arXiv e-prints, pages arXiv–2407.

Xiang Gao and Kamalika Das. 2024. Customizing language model responses with contrastive in-context learning. In Proceedings of the aaai conference on artificial intelligence, volume 38, pages 18039–18046.

Sonam Gupta, Yatin Nandwani, Asaf Yehudai, Dinesh Khandelwal, Dinesh Raghu, and Sachindra Joshi. 2025. Selective self-to-supervised fine-tuning for generalization in large language models. arXiv preprint arXiv:2502.08130.

Shadi Hamdan and Deniz Yuret. 2025. How much do llms learn from negative examples? arXiv preprint arXiv:2503.14391.

Seungwook Han, Jyothish Pari, Samuel J Gershman, and Pulkit Agrawal. 2025. General reasoning requires learning to reason from the get-go. arXiv preprint arXiv:2502.19402.

Moritz Hardt, Ben Recht, and Yoram Singer. 2016. Train faster, generalize better: Stability of stochastic gradient descent. In International conference on machine learning, pages 1225–1234. PMLR.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, and 1 others. 2024. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. arXiv preprint arXiv:2402.14008.

Yancheng He, Shilong Li, Jiaheng Liu, Weixun Wang, Xingyuan Bu, Ge Zhang, Zhongyuan Peng, Zhaoxiang Zhang, Zhicheng Zheng, Wenbo Su, and 1 others. 2025. Can large language models detect errors in long chain-of-thought reasoning? arXiv preprint arXiv:2502.19361.

Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. 2021a. Aligning ai with shared human values. Proceedings of the International Conference on Learning Representations (ICLR).

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021b. Measuring massive multitask language understanding. Proceedings of the International Conference on Learning Representations (ICLR).

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2024. Measuring mathematical problem solving with the math dataset, 2021. arXiv preprint arXiv:2103.03874, 2.

Maggie Huan, Yuetai Li, Tuney Zheng, Xiaoyu Xu, Seungone Kim, Minxin Du, Radha Poovendran, Graham Neubig, and Xiang Yue. 2025. Does math reasoning improve general llm capabilities? understanding transferability of llm reasoning. arXiv preprint arXiv:2507.00432.

Harsha Kokel, Michael Katz, Kavitha Srinivas, and Shirin Sohrabi. 2025. Acpbench: Reasoning about action, change, and planning. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 39, pages 26559–26568.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, and 1 others. 2022. Solving quantitative reasoning problems with language models. Advances in neural information processing systems, 35:3843–3857.

Dacheng Li, Shiyi Cao, Tyler Griggs, Shu Liu, Xiangxi Mo, Eric Tang, Sumanth Hegde, Kourosh Hakhamaneshi, Shishir G Patil, Matei Zaharia, and 1 others. 2025. Llms can easily learn to reason from demonstrations structure, not content, is what matters! arXiv preprint arXiv:2502.07374.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In Proceedings of the IEEE international conference on computer vision, pages 2980–2988.

Chenxi Liu, Junjie Liang, Yuqi Jia, Bochuan Cao, Yang Bai, Heng Huang, and Xun Chen. 2025. Explore data left behind in reinforcement learning for reasoning language models. arXiv preprint arXiv:2511.04800.

Junyu Luo, Xiao Luo, Xiusi Chen, Zhiping Xiao, Wei Ju, and Ming Zhang. 2024a. Semi-supervised fine-tuning for large language models. arXiv preprint arXiv:2410.14745.

Junyu Luo, Xiao Luo, Kaize Ding, Jingyang Yuan, Zhiping Xiao, and Ming Zhang. 2024b. Robustft: Robust supervised fine-tuning for large language models under noisy response. arXiv preprint arXiv:2412.14922.

Ivan Moshkov, Darragh Hanley, Ivan Sorokin, Shubham Toshniwal, Christof Henkel, Benedikt Schifferer, Wei Du, and Igor Gitman. 2025. Aimo-2 winning solution: Building state-of-the-art mathematical reasoning models with openmathreasoning dataset. arXiv preprint arXiv:2504.16891.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4. arXiv preprint arXiv:2306.02707.

Gongrui Nan, Siye Chen, Jing Huang, Mengyu Lu, Dexun Wang, Chunmei Xie, Weiqi Xiong, Xianzhou Zeng, Qixuan Zhou, Yadong Li, and Xingzhong Xu. 2025. NGRPO: Negative-enhanced group relative policy optimization. arXiv preprint arXiv:2509.18851.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. Advances in neural information processing systems, 35:27730–27744.

Zhuoshi Pan, Yu Li, Honglin Lin, Qizhi Pei, Zinan Tang, Wei Wu, Chenlin Ming, H Vicky Zhao, Conghui He, and Lijun Wu. 2025. Lemma: Learning from errors for mathematical advancement in llms. arXiv preprint arXiv:2503.17439.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. 2022. Measuring and narrowing the compositionality gap in language models. arXiv preprint arXiv:2210.03350.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. Preprint, arXiv:2402.03300.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, and 1 others. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. arXiv preprint arXiv:2210.09261.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Yongqi Tong, Dawei Li, Sizhe Wang, Yujia Wang, Fei Teng, and Jingbo Shang. 2024. Can llms learn from previous mistakes? investigating llms' errors to boost for reasoning. arXiv preprint arXiv:2403.20046.

David Vilares and Carlos Gómez-Rodríguez. 2019. Head-qa: A healthcare dataset for complex reasoning. arXiv preprint arXiv:1906.04701.

Renxi Wang, Haonan Li, Xudong Han, Yixuan Zhang, and Timothy Baldwin. 2024a. Learning from failure: Integrating negative examples when fine-tuning large language models as agents. arXiv preprint arXiv:2402.11651.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, and 1 others. 2024b. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. Advances in Neural Information Processing Systems, 37:95266–95290.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. arXiv preprint arXiv:2109.01652.

Juncheng Wu, Sheng Liu, Haoqin Tu, Hang Yu, Xiaoke Huang, James Zou, Cihang Xie, and Yuyin Zhou. 2025. Knowledge or reasoning? a close look at how llms think across domains. arXiv preprint arXiv:2506.02126.

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. 2025a. Qwen3 technical report. Preprint, arXiv:2505.09388.

Mutian Yang, Jiandong Gao, and Ji Wu. 2025b. Decoupling knowledge and reasoning in llms: An exploration using cognitive dual-system theory. arXiv preprint arXiv:2507.18178.

Zhaohui Yang, Yuxiao Ye, Shilei Jiang, Chen Hu, Linjing Li, Shihong Deng, and Daxin Jiang. 2025c. Unearthing gems from stones: Policy optimization with negative sample augmentation for LLM reasoning. arXiv preprint arXiv:2505.14403.

Erxin Yu, Jing Li, Ming Liao, Qi Zhu, Boyang Xue, Minghui Xu, Baojun Wang, Lanqing Hong, Fei Mi, and Lifeng Shang. 2025a. Self-error-instruct: Generalizing from errors for llms mathematical reasoning. arXiv preprint arXiv:2505.22591.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, and 16 others. 2025b. Dapo: An open-source llm reinforcement learning system at scale. Preprint, arXiv:2503.14476.

Yige Yuan, Teng Xiao, Shuchang Tao, Xue Wang, Jinyang Gao, Bolin Ding, and Bingbing Xu. 2025. Incentivizing reasoning from weak supervision. arXiv preprint arXiv:2505.20072.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. Star: Bootstrapping reasoning with reasoning. Advances in Neural Information Processing Systems, 35:15476–15488.

Chengshuai Zhao, Zhen Tan, Pingchuan Ma, Dawei Li, Bohan Jiang, Yancheng Wang, Yingzhen Yang, and Huan Liu. 2025. Is chain-of-thought reasoning of llms a mirage? a data distribution lens. arXiv preprint arXiv:2508.01191.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. 2025. Group sequence policy optimization. Preprint, arXiv:2507.18071.

Xinyu Zhu, Mengzhou Xia, Zhepei Wei, Wei-Lin Chen, Danqi Chen, and Yu Meng. 2025. The surprising effectiveness of negative reinforcement in LLM reasoning. arXiv preprint arXiv:2506.01347.

A Appendix

A.1 Experiments Setup

Distillation data curation We conduct experiments on mathematical reasoning and common sense, using Qwen3-8B (Yang et al., 2025a) to distill reasoning trajectories. For mathematics, we collect data from OpenMathReasoning (Moshkov et al., 2025), and for common sense from MMLU (Hendrycks et al., 2021b,a). Each trajectory is labeled as *positive* if the final answer matches the ground truth and *negative* otherwise. To ensure that all samples preserve complete reasoning structures and differ only in correctness, we discard instances exceeding 8,192 tokens. We then sample *positive* and *negative* data in a 1:1 ratio, resulting in 7.2k instances for mathematics and 17.4k for common sense.

Training Details We conduct experiments on the Qwen2.5 series (3B, 7B, 14B, 32B) (Team, 2024) and LLaMA-3.1-8B(Dubey et al., 2024). All models are fine-tuned for 20 epochs with a batch size of 128, using a cosine learning rate scheduler with 10% warm-up steps and a maximum learning rate of 5×10^{-5}. We set the training length to 20 epochs, as the loss does not converge earlier and benchmark performance continues to improve up to this point.

Evaluation Details Following Huan et al. (2025); Yuan et al. (2025), we evaluate models on three categories of benchmarks: (1) **mathematical reasoning**: MATH500 (Hendrycks et al., 2024), OlympiaBench (He et al., 2024), MinervaMath (Lewkowycz et al., 2022), and the competition-level AMC2023 (Art of Problem Solving Foundation, 2023); (2) **common sense reasoning**: MMLU, MMLU-Pro (Wang et al., 2024b), and BBH (Suzgun et al., 2022); (3) **other OOD reasoning**: ACPBench (Kokel et al., 2025) for planning, and HeadQA (Vilares and Gómez-Rodríguez, 2019) for medicine. Model performance is measured by accuracy. Evaluation uses the codebase from (Yuan et al., 2025), with sampling temperature 0.6, top-p 0.95, one sample per input, and max generation length 32,768 tokens.

We define in-domain and out-of-domain (OOD) evaluation based on the training data distribution. For models fine-tuned on mathematical reasoning tasks, in-domain evaluation uses mathematical problems while OOD evaluation employs other task categories. Conversely, models trained on MMLU are evaluated in-domain on commonsense tasks and OOD on the remaining domains. We compare three training strategies: using only *positive* samples, only *negative* samples, and a balanced combination of both.

Artifact Licenses and Intended Use: The models, the evaluation benchmarks and datasets are public artifacts. We utilize them in strict accordance with their respective licenses. Our use of these artifacts for SFT and reasoning evaluation is consistent with their intended use for scientific research.

A.2 Detailed Theoretical Derivation

We provide a theoretical framework to motivate the dynamic reweighting mechanism in Eq. 4. Under idealized smoothness and stability assumptions, our derivation suggests that GLOW can improve optimization conditioning. The core intuition is that a sample's short-horizon loss reduction acts as a proxy for the alignment between its gradient and the current update direction. Consequently, assuming low-gain samples align with undercovered subspaces, upweighting them adds positive semidefinite curvature along complementary directions. This potentially increases the spectrum of the weighted Fisher proxy, improves local conditioning, and reduces algorithmic sensitivity.

A.2.1 Setup and Notation

We analyze a single update step. Let θ be the current parameters and $\ell_i(\theta)$ the per-sample loss. Define

$$g_i \triangleq \nabla_\theta \ell_i(\theta), \qquad G \triangleq \frac{1}{N} \sum_{i=1}^{N} w_i \, g_i,$$

where $w_i \geq 0$ are the weights used in the current step. The update is

$$\theta' = \theta - \eta \, G.$$

We also define the weighted surrogate objective

$$R_w(\theta) \triangleq \frac{1}{N} \sum_{i=1}^{N} w_i \, \ell_i(\theta),$$

where the weights $\{w_i\}$ are held fixed during this step.

The weighted empirical Fisher proxy at θ is

$$F_w(\theta) \triangleq \frac{1}{N} \sum_{i=1}^{N} w_i \, g_i g_i^\top,$$

and we abbreviate $F_w(\theta)$ to F_w when the dependence is clear.

A.2.2 Notation and Standing Assumptions

Assumption A.1 (Smoothness, boundedness, and curvature injection)**.**

(A1) Each $\ell_i(\theta)$ is twice differentiable and L-smooth, namely $\|\nabla^2_\theta \ell_i(\theta)\|_{\mathrm{op}} \le L$ for all i and θ.

(A2) Gradients are uniformly bounded: $\|g_i(\theta)\|_2 \le G_{\max}$.

(A3) The learning rate η is small enough so that second-order remainders in Taylor expansions are controlled by L.

(A4) (Fisher Hessian closeness for the surrogate) At the iterates where the analysis is applied, the Hessian of R_w satisfies

$$\left\|\nabla^2_\theta R_w(\theta) - F_w(\theta)\right\|_{\mathrm{op}} \le \delta.$$

(A5) (Coverage on low-curvature directions) Let U be a k-dimensional subspace with projector P_U that captures low-curvature directions of F_w (e.g., the span of the k smallest-eigenvalue eigenvectors of F_w). Intuitively, the reweighting rule upweights samples with small gain, whose gradients are weakly aligned with the current update direction and thus tend to contribute complementary curvature along undercovered directions. Suppose the rule increases weights on a set T by increments $\delta w_i \ge 0$, inducing

$$\Delta F \triangleq \frac{1}{N} \sum_{i \in T} \delta w_i \, g_i g_i^\top.$$

We assume this update provides nontrivial coverage on U:

$$P_U \Delta F P_U \succeq \frac{\gamma}{k} P_U.$$

A.2.3 From Gain to Gradient Alignment

Lemma A.1 (Loss reduction and gradient alignment)**.** *Under (A1)–(A3), after the update $\theta' = \theta - \eta G$, we have*

$$\Delta_i \triangleq \ell_i(\theta) - \ell_i(\theta') = \eta \, g_i^\top G - \frac{1}{2}\eta^2 \, G^\top H_i(\xi_i) \, G$$

for some ξ_i on the line segment between θ and θ', where $H_i(\xi_i) = \nabla^2_\theta \ell_i(\xi_i)$. Moreover,

$$\left|\Delta_i - \eta \, g_i^\top G\right| \le \frac{1}{2} L \eta^2 \|G\|_2^2.$$

Proof. A second-order Taylor expansion of $\ell_i(\theta - \eta G)$ around θ gives the stated expression, and L-smoothness bounds the remainder. □

Lemma A.1 implies that, up to a controlled second-order term, the gain Δ_i is large when g_i aligns with the update direction G, and small when g_i is weakly aligned. Therefore, using small inter-epoch gain as a signal to upweight samples is consistent with prioritizing directions that are undercovered by recent optimization.

A.2.4 PSD Augmentation of the Fisher Proxy

Lemma A.2 (Positive weight increments induce PSD augmentation)**.** *If weights change by increments $\delta w_i \ge 0$ for $i \in T$, then the induced change in the weighted Fisher is*

$$\Delta F = \frac{1}{N} \sum_{i \in T} \delta w_i \, g_i g_i^\top,$$

which is positive semidefinite. Consequently, the updated Fisher $F'_w = F_w + \Delta F$ satisfies $F'_w \succeq F_w$, and its eigenvalues are monotonically nondecreasing.

Proof. Each $g_i g_i^\top$ is symmetric and positive semidefinite. With $\delta w_i \ge 0$, every term $\delta w_i g_i g_i^\top$ is positive semidefinite, hence so is their average ΔF. Thus $F'_w = F_w + \Delta F \succeq F_w$, and eigenvalue monotonicity follows from standard Weyl-type inequalities. □

A.2.5 Improving Low-Curvature Directions

Lemma A.2 guarantees a PSD augmentation $F'_w = F_w + \Delta F$, but PSD alone does not ensure improved curvature along the bottleneck directions: ΔF could concentrate on already well-conditioned directions and leave the low-curvature subspace unchanged. Assumption (A5) rules out this degeneracy by requiring the reweighting update to provide nontrivial coverage on U, which is consistent with upweighting small-gain samples whose gradients are weakly aligned with the current update and tend to contribute complementary directions.

Lemma A.3 (Improvement on a k-dimensional subspace)**.** *Let U be a k-dimensional subspace with projector P_U. Under (A5),*

$$\lambda_{\min}(F'_w|_U) \ge \lambda_{\min}(F_w|_U) + \frac{\gamma}{k},$$

where $F_w|_U$ denotes the restriction of F_w to U.

Proof. By (A5), $\Delta F|_U \succeq (\gamma/k)\, P_U$, so $\lambda_{\min}(\Delta F|_U) \geq \gamma/k$. Since $F'_w|_U = F_w|_U + \Delta F|_U$ and both are symmetric,

$$\begin{aligned}\lambda_{\min}(F'_w|_U) &\geq \lambda_{\min}(F_w|_U) + \lambda_{\min}(\Delta F|_U) \\ &\geq \lambda_{\min}(F_w|_U) + \frac{\gamma}{k}.\end{aligned}$$

□

The same coverage condition (A5) also implies an average-curvature increase on U:

$$\begin{aligned}\frac{1}{k}\operatorname{tr}(P_U F'_w) &= \frac{1}{k}\operatorname{tr}(P_U F_w) + \frac{1}{k}\operatorname{tr}(P_U \Delta F) \\ &\geq \frac{1}{k}\operatorname{tr}(P_U F_w) + \frac{\gamma}{k}.\end{aligned}$$

A.2.6 Transferring Improvement from Fisher to Hessian

Lemma A.4 (Fisher Hessian transfer on U)**.** *Let $H(\theta) = \nabla^2_\theta R_w(\theta)$ and $H'(\theta) = \nabla^2_\theta R_{w'}(\theta)$ be the Hessians of the surrogate objectives associated with F_w and F'_w, respectively. Under (A4),*

$$\lambda_{\min}(H'|_U) \geq \lambda_{\min}(H|_U) + \frac{\gamma}{k} - 2\delta.$$

Proof. For any symmetric matrices A, B, $|\lambda_{\min}(A) - \lambda_{\min}(B)| \leq \|A - B\|_{\mathrm{op}}$. Applying this to (H, F_w) and (H', F'_w) and using (A4) yields

$$\begin{aligned}\lambda_{\min}(H'|_U) &\geq \lambda_{\min}(F'_w|_U) - \delta \\ &\geq \lambda_{\min}(F_w|_U) + \frac{\gamma}{k} - \delta \\ &\geq \lambda_{\min}(H|_U) + \frac{\gamma}{k} - 2\delta.\end{aligned}$$

where the middle inequality uses Lemma A.3. □

A.2.7 Conditioning and Stability-Based Generalization

Lemma A.5 (Improved conditioning reduces parameter sensitivity)**.** *Assume a restricted strong convexity condition on U: $\lambda_{\min}(H|_U) \geq \mu$, and standard Lipschitz conditions for gradients hold. Then the algorithmic stability scale is inversely proportional to μ. Consequently, increasing $\lambda_{\min}(H|_U)$ to $\mu' = \mu + \gamma/k - 2\delta$ reduces sensitivity to data perturbations and yields a smaller stability-based generalization bound; see Bousquet and Elisseeff (2002); Hardt et al. (2016).*

Proposition A.6 (Conditioning and generalization improvement)**.** *Under (A1)–(A5), the reweighting rule induces a PSD Fisher augmentation and improves curvature on the low-curvature subspace U. In particular:*

1. *(Curvature on U) The Fisher proxy satisfies*

$$\begin{aligned}\frac{1}{k}\operatorname{tr}(P_U F'_w P_U) &\geq \frac{1}{k}\operatorname{tr}(P_U F_w P_U) + \frac{\gamma}{k}, \\ \lambda_{\min}(F'_w|_U) &\geq \lambda_{\min}(F_w|_U) + \frac{\gamma}{k}.\end{aligned}$$

2. *(Hessian transfer) The Hessian of the surrogate objective satisfies*

$$\lambda_{\min}(H'|_U) \geq \lambda_{\min}(H|_U) + \frac{\gamma}{k} - 2\delta.$$

3. *(Stability and generalization) If $\lambda_{\min}(H|_U) \geq \mu$, then after reweighting, the effective curvature lower bound increases to $\mu' = \mu + \gamma/k - 2\delta$, which improves stability-based generalization bounds.*

Proof. Item 1 follows from Lemma A.2, assumption (A5), and Lemma A.3. Item 2 follows from Lemma A.4. Item 3 follows from Lemma A.5. □

In summary, gain-based reweighting uses small gain as a signal of weak alignment with recent updates, upweights such samples, injects curvature into undercovered directions through PSD Fisher augmentation, and improves local conditioning on low-curvature subspaces, which supports stronger stability-based generalization guarantees.

A.3 IRM View of Diverse Negative Trajectories

This section formalizes our interpretation through Invariant Risk Minimization (IRM) (Arjovsky et al., 2019) in an autoregressive language modeling setting. Let $\mathcal{E}$ denote the set of environments induced by error categories of *negative* trajectories. Each environment $e \in \mathcal{E}$ corresponds to a distribution D^e over sequences (x, y), where x is the input and y is the target reasoning trajectory.

We decompose the language model into a shared representation map Φ and a shared next-token predictor w, where Φ denotes the model body and w denotes the vocabulary projection head. IRM seeks a representation Φ and a predictor w such that the same w is optimal across all environments when paired with Φ:

$$\begin{cases}\min\limits_{\Phi,\, w} \sum\limits_{e\in\mathcal{E}} R^e(w \circ \Phi), \\ \text{s.t. } \ w \in \arg\min_{w'} R^e(w' \circ \Phi), \ \forall e \in \mathcal{E}.\end{cases} \tag{6}$$

The per-environment autoregressive risk is

$$R^e(w \circ \Phi) = \mathbb{E}_{(x,y)\sim D^e}\Bigg[\sum_{t=1}^{|y|} \ell\Big(w\big(\Phi(x, y_{<t})\big),\, y_t\Big)\Bigg]. \tag{7}$$

where ℓ denotes the cross-entropy loss.

Because w is shared across all $e \in \mathcal{E}$, the shared-optimality constraint encourages Φ to encode reasoning features that remain predictive across heterogeneous error environments. Under our interpretation, *negative* trajectories enlarge $\mathcal{E}$ by covering many error categories, which explains why diversity in *negatives*, rather than any single error type, can improve robustness and OOD generalization.

A.4 Detailed Taxonomy of Negative Training Samples

We provide statistics on the detailed categorization of *negative* samples in our training dataset. As shown in Figure 3a and Figure 3b, the error types of samples from OpenMathReasoning and MMLU that are not selected by reject sampling can be grouped into nine major categories and twenty-four subcategories. Although the distribution across categories is imbalanced, the errors still exhibit a broad coverage, ensuring a comprehensive representation of error types.

A.5 Hyperparameter Sensitivity of GLOW

As shown in Figure 4, GLOW yields modest improvements over the full-SFT reference in most configurations. Varying α between 0.8 and 1.5 leads to small changes, and $\beta = 12$ is generally stronger than $\beta = 10$ or $\beta = 18$ at matched α. These results suggest incremental gains with moderate hyperparameter choices in our setup.

A.6 Training Loss on OpenMathReasoning and MMLU

Figure 5 compares training losses for all models on OpenMathReasoning and MMLU under *positive* and *negative* settings.

A.7 Model Performance Evolution Across Epochs

Table 10 compares intermediate checkpoints (epochs 5–20) for Qwen2.5-7B and 32B. Across settings, *negative-trajectory* SFT consistently outperforms the base model, yielding gains comparable to its *positive* counterpart while often matching or exceeding it on OOD benchmarks. This confirms that *negatives* provide structured supervision rather than noise.

A.8 Negatives Are Frequently Upweighted by GLOW

Figure 6 reports the fraction of *negatives* among the most upweighted examples during GLOW training. We fine-tune Qwen2.5-3B on Math and MMLU using a shuffled mixture of *positives* and *negatives*, where responses are distilled from Qwen3-8B and labels are determined by final-answer matching. At each optimization step, we select the example with the largest upweighting signal and compute, within each epoch, the proportion of *negatives* among these selections. The fraction stays above 50% for most epochs, peaks around 75%–80% early in training, and then gradually approaches 50%. This aligns with the design of GLOW, which emphasizes samples with small inter-epoch loss reduction, a behavior more common among *negatives*.

A.9 Pass@k under OOD Evaluation

We evaluate pass@k ($k \in \{4, 8, 16, 32\}$) averaged over three OOD benchmarks per setting (OpenMath: BBH, ACPBench, HeadQA; MMLU: Olympia, ACPBench, HeadQA). As shown in Figures 7 and 8, *negative*-trained models consistently achieve higher pass@k across all k. This superior multi-sample efficiency confirms that *negatives* promote broader reasoning exploration and provide a stronger base policy for subsequent RL.

A.10 Reasoning Length Under On-Policy Decoding

A natural confound in chain-of-thought evaluation is response length. To assess this possibility, we measure on-policy output lengths under the same decoding setup for the positive-trained and negative-trained Qwen2.5-7B models. Table 11 reports benchmark-level mean lengths for the `<think>` span, the answer span, and the total output.

The results do not support a pure verbosity account. For the Qwen2.5-7B model trained on OpenMathReasoning, the mean `<think>` span increases from 2283.64 to 2399.55 tokens (+5.08%), whereas the mean answer span decreases from 567.74 to 557.15 tokens (-1.87%). The corresponding total output increases only from 2851.38 to 2956.70 tokens (+3.69%). For the Qwen2.5-7B model trained on MMLU, the mean `<think>` span increases from

(a) Qwen2.5-7B is fine-tuned on the **math reasoning** dataset using ***positive*** distilled trajectories.

Epoch	Math500	Minerva	Olympia	AMC	MMLU	MMLU-Pro	BBH
Base	58.40	26.84	26.07	52.50	55.80	26.56	51.10
5epoch	72.80	37.13	37.19	45.00	60.95	30.34	54.69
10epoch	75.80	38.24	40.59	65.00	64.06	32.50	61.62
15epoch	77.20	36.76	41.93	55.00	60.81	32.15	59.69
20epoch	78.00	36.76	41.78	57.50	61.03	32.70	60.58

(b) Qwen2.5-7B is fine-tuned on the **math reasoning** dataset using ***negative*** distilled trajectories.

Epoch	Math500	Minerva	Olympia	AMC	MMLU	MMLU-Pro	BBH
Base	58.40	26.84	26.07	52.50	55.80	26.56	51.10
5epoch	71.20	31.99	31.56	47.50	62.58	44.04	56.28
10epoch	77.20	34.93	39.26	50.00	71.39	52.14	69.49
15epoch	78.60	39.71	38.37	52.50	72.10	52.24	71.09
20epoch	77.60	40.44	38.37	57.50	73.11	53.74	71.73

(c) Qwen2.5-7B is fine-tuned on the **general reasoning** dataset using ***positive*** distilled trajectories.

Epoch	Math500	Minerva	Olympia	AMC	MMLU	MMLU-Pro	BBH
Base	58.40	26.84	26.07	52.50	55.80	26.56	51.10
5epoch	72.00	36.76	37.33	47.50	73.62	50.61	64.05
10epoch	74.60	37.50	41.48	55.00	73.79	53.32	69.73
15epoch	72.00	37.50	39.26	50.00	74.11	53.91	68.34
20epoch	74.40	37.50	39.85	50.00	73.42	53.22	68.23

(d) Qwen2.5-7B is fine-tuned on the **general reasoning** dataset using ***negative*** distilled trajectories.

Epoch	Math500	Minerva	Olympia	AMC	MMLU	MMLU-Pro	BBH
Base	58.40	26.84	26.07	52.50	55.80	26.56	51.10
5epoch	76.80	36.76	37.78	47.50	71.09	43.99	66.00
10epoch	76.80	37.87	40.30	52.50	71.43	45.87	68.84
15epoch	76.80	37.13	41.48	55.00	71.30	44.62	69.30
20epoch	77.00	37.13	42.07	60.00	71.23	45.79	69.46

(e) Qwen2.5-32B is fine-tuned on the **math reasoning** dataset using ***positive*** distilled trajectories.

Epoch	Math500	Minerva	Olympia	AMC	MMLU	MMLU-Pro	BBH
Base	63.20	34.19	26.52	35.00	68.34	39.80	58.65
5epoch	90.20	49.63	59.11	85.00	76.53	46.77	78.04
10epoch	92.60	50.00	60.44	85.00	78.63	51.67	79.01
15epoch	93.00	48.53	62.07	90.00	78.72	51.99	80.57
20epoch	91.40	50.74	60.89	85.00	79.01	54.31	80.61

(f) Qwen2.5-32B is fine-tuned on the **math reasoning** dataset using ***negative*** distilled trajectories.

Epoch	Math500	Minerva	Olympia	AMC	MMLU	MMLU-Pro	BBH
Base	63.20	34.19	26.52	35.00	68.34	39.80	58.65
5epoch	88.40	45.22	52.30	85.00	83.07	68.23	83.55
10epoch	92.20	51.10	57.93	85.00	85.14	73.75	84.22
15epoch	91.20	50.74	57.33	90.00	85.02	73.48	84.62
20epoch	92.20	50.74	58.37	95.00	85.47	73.53	84.51

(g) Qwen2.5-32B is fine-tuned on the **general reasoning** dataset using ***positive*** distilled trajectories.

Epoch	Math500	Minerva	Olympia	AMC	MMLU	MMLU-Pro	BBH
Base	63.20	34.19	26.52	35.00	68.34	39.80	58.65
5epoch	84.60	44.85	52.00	62.50	82.10	66.54	80.03
10epoch	86.60	46.69	55.70	75.00	81.14	67.01	80.69
15epoch	85.00	47.06	56.59	75.00	81.73	68.33	81.73
20epoch	85.20	46.69	56.15	75.00	81.97	68.54	81.60

(h) Qwen2.5-32B is fine-tuned on the **math reasoning** dataset using ***negative*** distilled trajectories.

Epoch	Math500	Minerva	Olympia	AMC	MMLU	MMLU-Pro	BBH
Base	63.20	34.19	26.52	35.00	68.34	39.80	58.65
5epoch	85.00	44.49	51.26	77.50	78.74	57.48	79.09
10epoch	87.20	46.30	54.52	75.00	79.01	60.43	80.88
15epoch	86.40	47.79	55.70	65.00	77.77	57.14	79.97
20epoch	86.40	47.06	56.89	72.50	77.99	58.34	80.71

Table 10: **Checkpoint evaluation across SFT epochs with distilled reasoning trajectories.** We report performance at 5, 10, 15, and 20 epochs. Each row corresponds to a model size and training dataset, and each row contains two subtables that compare training on *positive* (left) versus *negative* (right) distilled trajectories. Columns in each subtable correspond to benchmarks. Rows correspond to training epochs, with 'Base' denoting the model before SFT.

1619.58 to 1858.28 tokens (+14.74%), while the mean answer span decreases from 684.84 to 651.37 tokens (-4.89%), yielding a total increase from 2304.42 to 2509.64 tokens (+8.91%). Across the 18 benchmark-level comparisons in Table 11, the `<think>` span grows by 10.63% on average.

Thus, negative-trained models tend to allocate additional tokens inside the reasoning span while keeping the final answer span comparable or slightly shorter. This pattern is inconsistent with a pure verbosity effect, which would be expected to lengthen the final answer as well, and instead aligns with the interpretation that *negatives* encourage broader exploration during reasoning while preserving sharper final commitment.

A.11 Prompt for Categorize Negative Samples

We design a structured prompt to categorize each erroneous reasoning trajectory into a fine-grained error class. The classification framework contains 9 primary categories and 22 sub-categories. The full classification schema and the prompt used for categorization are shown in Figure 9.

A.12 Case Study of Negative Samples

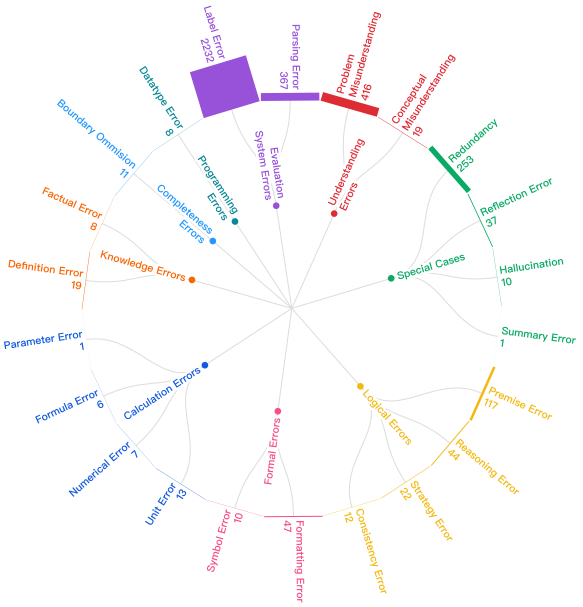
As discussed in Section 4.3, *negative* trajectories exhibit higher entropy than positives ones on certain reasoning tokens and transition words. For illustration, we select one case and highlight the high-entropy segments. The results in Figure 10 show that *negatives* contain substantially more such reasoning-related high-entropy fragments than *positives*.

Qwen2.5-7B trained on OpenMathReasoning											
Span	**Train**	**Math500**	**Minerva**	**Olympia**	**AMC**	**MMLU**	**MMLU-Pro**	**BBH**	**ACPBench**	**HeadQA**	**Avg.**
<think> span	Positive	2467.08	2671.41	2790.35	2749.69	1591.75	2010.23	2133.37	2602.78	1536.12	2283.64
	Negative	2505.56	2793.06	2818.95	2598.36	1733.71	2247.95	2310.42	2828.62	1759.32	2399.55
Answer span	Positive	502.17	551.13	642.06	645.77	506.70	591.06	447.96	700.03	522.78	567.74
	Negative	506.43	505.86	682.55	624.45	518.49	606.05	440.56	583.61	546.34	557.15
Total output	Positive	2969.25	3222.54	3432.41	3395.46	2098.45	2601.29	2581.33	3302.81	2058.90	2851.38
	Negative	3011.99	3298.92	3501.50	3222.81	2252.20	2854.00	2750.98	3412.23	2305.66	2956.70
Qwen2.5-7B trained on MMLU											
Span	**Train**	**Math500**	**Minerva**	**Olympia**	**AMC**	**MMLU**	**MMLU-Pro**	**BBH**	**ACPBench**	**HeadQA**	**Avg.**
<think> span	Positive	1258.80	1514.07	1699.26	1587.57	1422.85	1960.56	1484.31	2186.37	1462.45	1619.58
	Negative	1435.82	1781.16	1810.39	1944.38	1652.44	2153.02	1741.03	2393.92	1812.33	1858.28
Answer span	Positive	582.73	797.56	953.76	808.10	501.59	653.65	505.08	821.57	539.51	684.84
	Negative	567.01	770.61	915.97	645.25	520.17	650.87	500.61	737.58	554.22	651.37
Total output	Positive	1841.53	2311.63	2653.02	2395.67	1924.44	2614.21	1989.39	3007.94	2001.96	2304.42
	Negative	2002.83	2551.77	2726.36	2589.63	2172.61	2803.89	2241.64	3131.50	2366.55	2509.64

Table 11: On-policy output length statistics under identical decoding settings. Lengths are measured in tokens. "Positive" and "Negative" denote models fine-tuned on positive-only and negative-only data, respectively. "Avg." is the average over the nine evaluation benchmarks.

A.13 Case Study of Samples Generated by Various Models

To qualitatively evaluate the differences in reasoning behavior, we provide a comparative case study in Figure 11, contrasting trajectories from M_{pos} and M_{neg}. M_{neg} tends to exhibit more frequent use of discourse and hesitation tokens(e.g. "wait" ,"but"), particularly when encountering complex reasoning steps. These qualitative observations align with the token distribution analysis in Figure 2, confirming that M_{neg} allocates a larger portion of its generation budget to connective exploration. By maintaining multiple plausible continuations instead of committing prematurely to a single path, M_{neg} demonstrates a more exhaustive search of the reasoning space before finalizing its response.



(a) Error distribution in OpenMathReasoning.

(b) Error distribution in MMLU.

Figure 3: Detailed categorization of *negative* samples in OpenMathReasoning and MMLU.

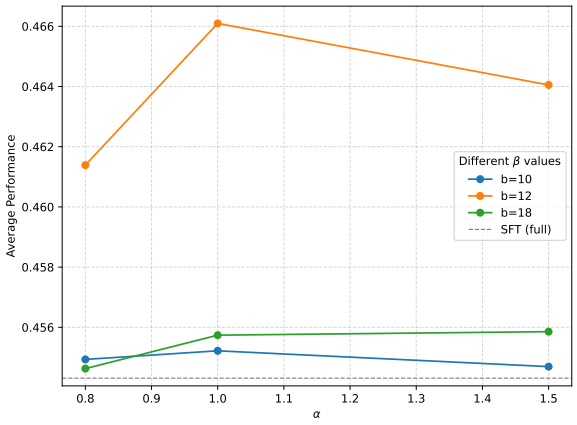
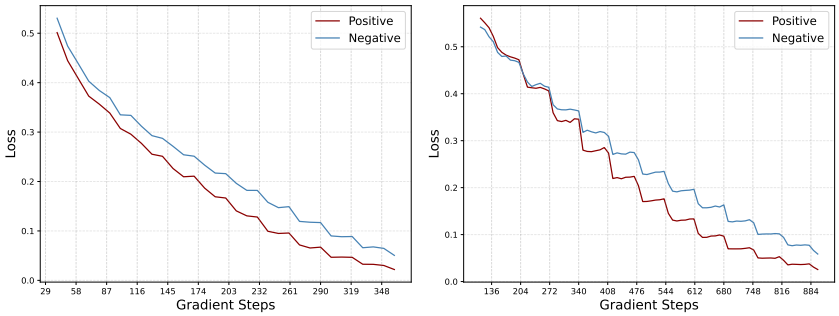
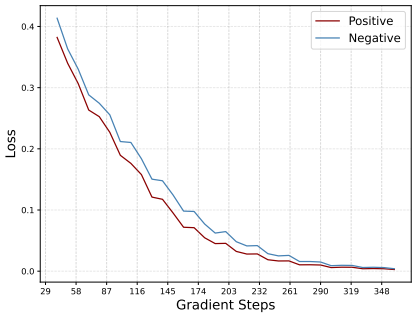


Figure 4: Ablation study on the hyperparameters α and β. GLOW exhibits stable performance across different settings, demonstrating the robustness of the reweighting formulation.

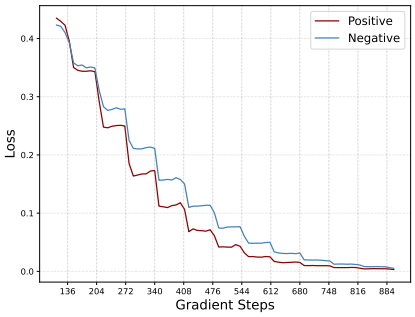


(a) Qwen2.5-3B on OpenMathReasoning

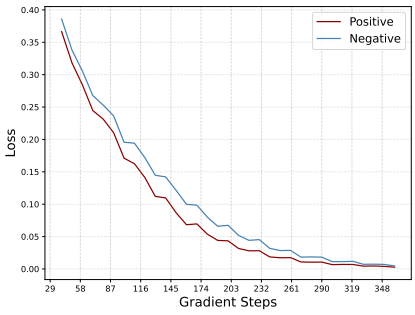
(b) Qwen2.5-3B on MMLU



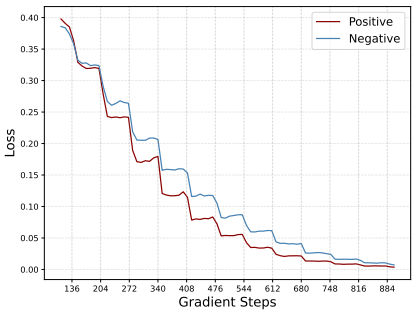
(c) Qwen2.5-7B on OpenMathReasoning



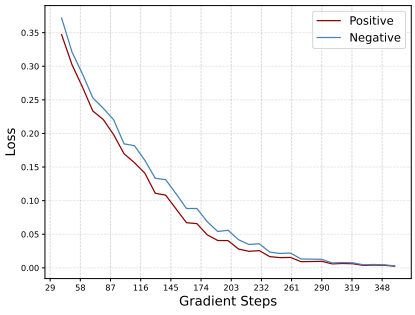
(d) Qwen2.5-7B on MMLU



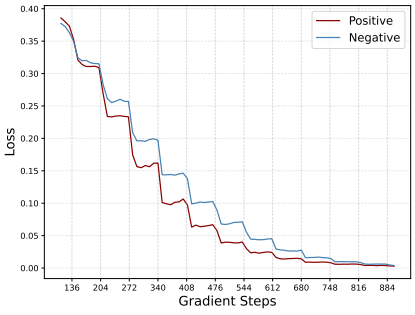
(e) Qwen2.5-14B on OpenMathReasoning



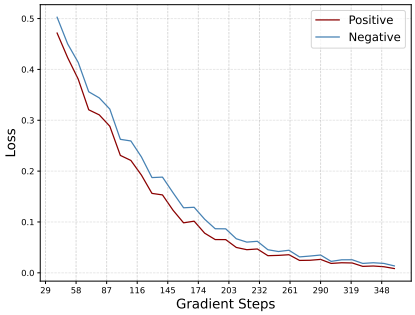
(f) Qwen2.5-14B on MMLU



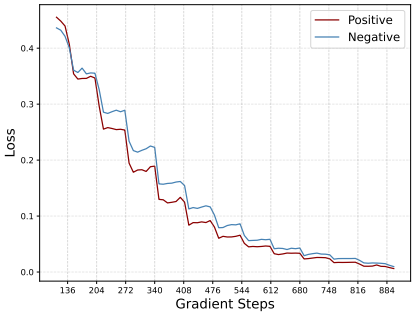
(g) Qwen2.5-32B on OpenMathReasoning



(h) Qwen2.5-32B on MMLU



(i) Llama3.1-8B on OpenMathReasoning



(j) Llama3.1-8B on MMLU

Figure 5: Training loss of Qwen2.5 models and Llama3.1-8B on OpenMathReasoning (left) and MMLU (right). Losses drop across epochs, with the *positive* setting converging faster than the *negative*.

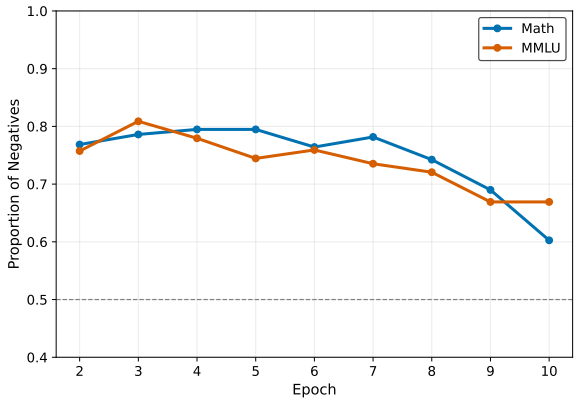


Figure 6: Fraction of *negatives* among stepwise highest-weight samples across epochs for Math and MMLU training.

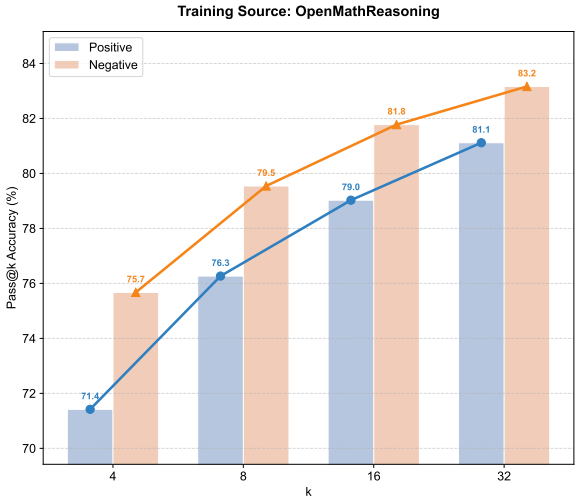


Figure 7: OOD pass@k for models trained on OpenMathReasoning under *positive-only* vs. *negative-only* SFT. Results are averaged over BBH, ACPBench, and HeadQA.

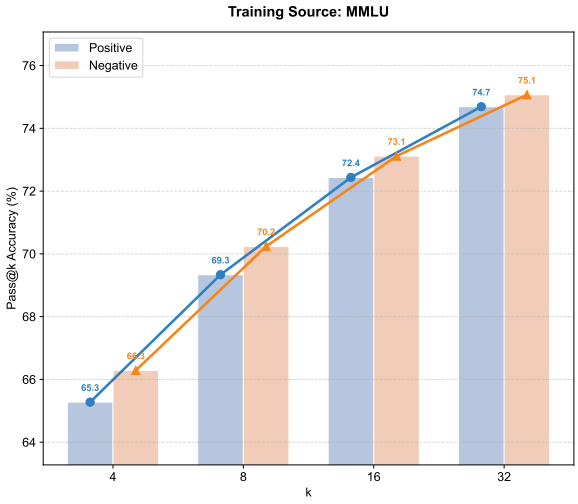


Figure 8: OOD pass@k for models trained on MMLU under *positive-only* vs. *negative-only* SFT. Results are averaged over Olympia, ACPBench, and HeadQA.

Prompt for Categorizing Negative Samples

```
You are an expert AI assistant tasked with identifying the single,
most specific error category from the list below.

Error Category List:
- Primary_category: Understanding Errors
  - sub_category: Problem Misunderstanding, Conceptual Misunderstanding
- Primary_category: Knowledge Errors
  - sub_category: Factual Error, Theorem Error, Definition Error
- Primary_category: Logical Errors
  - sub_category: Strategy Error, Reasoning Error, Premise Error, Consistency Error
- Primary_category: Calculation Errors
  - sub_category: Numerical Error, Formula Error, Parameter Error, Unit Error
- Primary_category: Programming Errors
  - sub_category: Syntax Error, Function Error, Data Type Error
- Primary_category: Formal Errors
  - sub_category: Symbol Error, Formatting Error
- Primary_category: Completeness Errors
  - sub_category: Boundary Omission
- Primary_category: Special Cases
  - sub_category: Reflection Error, Summary Error, Hallucination, Redundancy
- Primary_category: Evaluation System Errors
  - sub_category: Incorrect Ground Truth, Correct Answer Parsing Error

Data for Analysis:
- Question: {question}
- Ground Truth Answer: {groundtruth}
- Model's Reasoning Process (to be analyzed): {model_reasoning}

CRITICAL INSTRUCTION:
Analyze the provided reasoning process. Your response MUST be ONLY a single,
raw JSON object with the keys "sub_category" and "analysis". Do not include any
other text, explanations, apologies, or markdown formatting.

Example of a perfect response:
{
  "sub_category": "Premise Error",
  "analysis": "The model incorrectly assumed that all bicycles use plastic
              squares for identification, which is a flawed premise not
              supported by the question's context."
}
```

Figure 9: Prompt used for categorizing negative reasoning samples into predefined error subcategories.

Problem

The graph shows the total distance Sam drove from 6 a.m. to 11 a.m. How many miles per hour is the car's average speed for the period from 6 a.m. to 11 a.m.?

unitsize(0.2inch); draw((0,0)–(5.5,0)); draw((0,0)–(0,8.5)); draw((1,0)–(1,8.5)); draw((2,0)–(2,8.5)); draw((3,0)–(3,8.5)); draw((4,0)–(4,8.5)); draw((5,0)–(5,8.5)); draw((0,1)–(5.5,1)); draw((0,8)–(5.5,8)); draw((0,7)–(5.5,7)); draw((0,6)–(5.5,6)); draw((0,5)–(5.5,5)); draw((0,4)–(5.5,4)); draw((0,3)–(5.5,3)); draw((0,2)–(5.5,2)); draw((0,0)–(1,2)–(2,3)–(3,5)–(4,6)–(5,8)); dot((0,0)); dot((1,2)); dot((2,3)); dot((3,5)); dot((4,6)); dot((5,8)); label("6",(0,-0.5),S); label("7",(1,-0.5),S); label("8",(2,-0.5),S); label("9",(3,-0.5),S); label("10",(4,-0.5),S); label("11",(5,-0.5),S); label("0",(-0.5,0),W); label("40",(-0.5,2),W); label("80",(-0.5,4),W); label("120",(-0.5,6),W); label("160",(-0.5,8),W); label("Time of Day (a.m.)",(2.7,-2),S); label("Total distance",(-0.5,9),N);

Model's Thinking Trajectory and Answer

Let me start with the total time. The period is from 6 a.m. to 11 a.m. Let me count the hours. From 6 to 7 is 1 hour, 7 to 8 is another, 8 to 9, 9 to 10, and 10 to 11. That's 5 hours total. Wait, is that right? Let me check again. From 6 to 11 is 5 hours? Wait, 11 minus 6 is 5, yes. So total time is 5 hours. Okay, that seems straightforward.
Now, the total distance. The graph shows the total distance driven at different times. The Asymptote code draws a graph with time on the x-axis and total distance on the y-axis. The points given are at time 6 a.m. (which is the first point, x = 0), then at 7 a.m. (x = 1), 8 a.m. (x = 2), etc., up to 11 a.m. (x = 5). The corresponding distances are given as y-values. Let me check the Asymptote code again to see the coordinates.
Looking at the Asymptote code, the graph is drawn from $(0, 0)$ to $(1, 2)$ to $(2, 3)$ to $(3, 5)$ to $(4, 6)$ to $(5, 8)$. The labels on the y-axis are 0, 40, 80, 120, 160. So each unit on the y-axis corresponds to 40 miles? Let me confirm. The labels are at $y = 0, 2, 4, 6, 8$. The labels next to them are 0, 40, 80, 120, 160. So, each vertical unit is 40 miles. Therefore, the y-coordinate multiplied by 40 gives the total distance in miles.
So, for example:

- At 6 a.m. (x = 0), the total distance is 0 miles.
- At 7 a.m. (x = 1), the total distance is 2 units, which is $2 \times 40 = 80$ miles.
- At 8 a.m. (x = 2), it's 3 units, so $3 \times 40 = 120$ miles.
- At 9 a.m. (x = 3), 5 units, which is $5 \times 40 = 200$ miles.
- At 10 a.m. (x = 4), 6 units, which is $6 \times 40 = 240$ miles.
- At 11 a.m. (x = 5), 8 units, which is $8 \times 40 = 320$ miles.

The question is about the total distance from 6 a.m. to 11 a.m. The graph is showing cumulative distance. So at 11 a.m., the total distance is 320 miles. Therefore, the total distance driven is 320 miles.
If the graph is cumulative, then the total distance at 11 a.m. is the total distance driven from 6 a.m. to 11 a.m., so that would be 320 miles. Therefore, total distance is 320 miles, total time is 5 hours. Therefore, average speed is $320 \div 5 = 64$ miles per hour.
Let me check if the total distance is indeed 320. The last point is at 11 a.m., which is $x = 5, y = 8$. Since each y unit is 40 miles, $8 \times 40 = 320$. Yes. So that seems correct.
Alternatively, maybe the question is trying to trick me? Suppose someone might think that the total distance is the sum of all the segments, but because the graph is cumulative, the total distance at 11 a.m. is already the total distance driven from 6 a.m. to 11 a.m.
We can also confirm by looking at increments: From 6–7: 80 miles, From 7–8: $120 - 80 = 40$ miles. From 8–9: $200 - 120 = 80$ miles. From 9–10: $240 - 200 = 40$ miles. From 10–11: $320 - 240 = 80$ miles Summing gives $80 + 40 + 80 + 40 + 80 = 320$ miles, consistent with the final value.
So average speed is total distance divided by total time. Total time is 5 hours. $320 \div 5 = 64$, so 64 miles per hour.
The time from 6 a.m. to 11 a.m. is indeed 5 hours. The x-axis is labeled from 6 a.m. to 11 a.m., with each unit representing one hour, so that checks out.
Therefore, I think my answer is correct.
Final Answer: the car's average speed is $\boxed{64}$ miles per hour.

Figure 10: Case study of a negative trajectory from the OpenMathReasoning training dataset. The model misinterprets the problem, but its subsequent step-by-step reasoning and formula derivations remain structurally correct.

An Example from MATH-500

Problem: Denali and Nate work for a dog walking business and are paid for each dog they walk. Denali is responsible for 16 dogs and Nate is responsible for 12 dogs. Under the company's new policy, they will be assigned or unassigned new dogs in groups of x dogs. The ratio of Denali's pay to Nate's pay would be the same if Denali started walking $4x$ more dogs and Nate stayed at 12 dogs or if x of Nate's dogs were reassigned to Denali. Find x if $x \neq 0$. **Answer**: $\boxed{5}$

M_{pos} Thinking Trajectories

Okay, let me try to figure out this problem...
So, setting up the equation...
Let me compute both sides...

...would be the same under two different scenarios. Let me parse that...

So, solution is x = 0...

But the problem says ...

Let me check again...

x = 0 is correct...
The final answer is $\boxed{0}$.

M_{neg} Thinking Trajectories

Okay, let me try to figure out this problem...
The key is that...
Now, we can solve this...
..., so the answer is x = 3.
Wait, let me check...
it is wrong...

Alternatively, maybe there's a different way to interpret the problem?...
Maybe I missed something...

Alternatively, maybe there's a different interpretation? Let me check...
Wait, no. The problem says...
..., so the answer is 5.
But, Let me check ...
Therefore, x = 5 is correct...
The final answer is $\boxed{5}$.

Figure 11: Case study of thinking trajectories for M_{pos} and M_{neg} on the same question.