On the Effectiveness of Trainable Steering Vectors in Supervised Fine-Tuning

Anonymous ACL submission

Abstract

Prior work showed that trainable steering vectors can match full-model reinforcement learning fine-tuning on mathematical reasoning benchmarks while updating only a tiny fraction of parameters. We test whether this equivalence extends to supervised fine-tuning and identify the factors that control it. Using Qwen2.5-Math-7B and LLaMa3.1-8B-Instruct trained on OpenThoughts-114k-math and evaluated on six math benchmarks, we find a consistent gap under SFT: steering vectors underperform full-model fine-tuning, unlike in the RL setting. We show that SFT-trained steering models deviate more from their base models, and that closing the gap by increasing adapter capacity requires full-rank updates (e.g., $\approx 27\%$ of parameters for Qwen2.5-Math-7B when scaling `MLP.down_proj` LoRA). Finally, we show that changing the data removes the gap: when we distill from RL-trained teachers by training on selected positive generations, low-parameter steering vectors match full-tuning, without simply reproducing RL steering directions.

1 Introduction

Prior work by (Sinii et al., 2025b) showed that applying Reinforcement Learning (RL) to trainable steering vectors can achieve the same accuracy on math reasoning tasks as full-model RL fine-tuning, despite the extreme parameter constraint. This raises the question of whether Supervised Fine-Tuning (SFT) can replicate the same outcome, and more generally, what determines when a small representational subspace is sufficient for reasoning-relevant adaptation.

SFT and RL provide qualitatively different training signals. SFT supplies token-level supervision from curated reasoning traces, whereas RL with verifiable rewards can optimize from outcome-level feedback. Motivated by recent analyses (Schulman and Lab, 2025) suggesting that such differences matter most under constrained adaptation capacity, we compare steering-vector training and full-model fine-tuning under both objectives in a matched experimental setup.

We find that the RL equivalence does not carry over to SFT: steering vectors consistently underperform full-model fine-tuning across both model families and benchmarks. We then probe two potential explanations. First, we measure deviation from the base model and observe that SFT-trained steering models produce lower-probability generations under the base distribution. Second, we vary adaptation capacity by sweeping LoRA rank on `MLP.down_proj` and find that the SFT gap closes only at full rank. Finally, we show that the gap is not inevitable: training on generations produced by RL-trained teachers (distillation) closes the steering/full-tuning gap, and the resulting steering vectors are not aligned with the RL-trained steering vectors, suggesting that distillation induces a different mechanism within the same constrained parameterization.

2 Related Work

Reinforcement Learning and Supervised Fine-Tuning for Reasoning. Both supervised fine-tuning (SFT) and reinforcement learning with verifiable rewards (RLVR) have been shown to improve reasoning performance in large language models. SFT-based approaches leverage curated reasoning traces or chain-of-thought annotations and can yield substantial gains when sufficient high-quality data is available (Li et al., 2024; Ye et al., 2025; Yao et al., 2023). More recently, RLVR has emerged as a dominant paradigm for reasoning-centric training, underpinning systems such as OpenAI *o1* (Jaech et al., 2024), DeepSeek-R1 (Guo et al., 2025), and several open reproductions (Zeng et al., 2025; Hu et al., 2025), and demonstrating that outcome-based rewards alone can induce long reasoning trajectories and strong benchmark performance. Prior work has also highlighted that the two training paradigms can lead to systematically different behaviors and capabilities (Matsutani et al., 2025; Chu et al., 2025).

Parameter-Efficient Adaptation and Steering. A large body of work has explored parameter-efficient fine-tuning (PEFT) as an alternative to full-model adaptation. Low-Rank Adaptation (LoRA) (Hu et al., 2022) introduced the idea of constraining updates to low-dimensional subspaces, initially applied to attention projections. Subsequent work demonstrated that applying LoRA more broadly, including to MLP layers, can

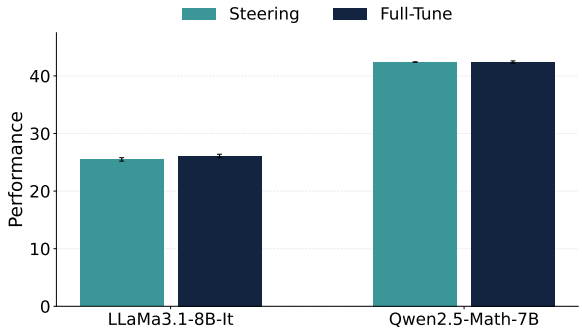


Figure 1: **Steering with Reinforcement learning.** Mean accuracy across six math benchmarks comparing trainable steering vectors with full-model fine-tuning. Across the two models, steering vectors match or closely approach the performance of full fine-tuning.

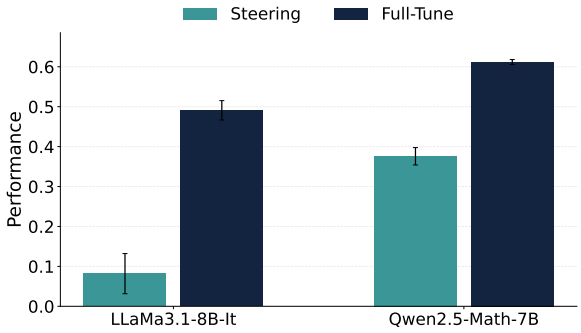


Figure 2: **Steering with Supervised Fine-Tuning.** Mean accuracy across six math benchmarks comparing trainable steering vectors with full-model fine-tuning. Scores are normalized between the base and RL models. Unlike in the RL setting, steering vectors consistently underperform full-model fine-tuning across both architectures.

further improve performance, highlighting the importance of where adaptation capacity is allocated within the model (Zaken et al., 2022). In parallel, bias-only fine-tuning methods such as BitFit showed that even extremely restricted update regimes can be effective across a range of NLP tasks (Zaken et al., 2022).

Steering vectors build on these ideas by operating directly at the level of hidden representations, inducing behavioral changes through low-dimensional directions rather than widespread parameter updates. Prior work on activation engineering and contrastive steering demonstrated that such directions can reliably elicit target behaviors at inference time (Turner et al., 2023; Panickssery and Bowman, 2023). More recent studies have shown that steering vectors trained with reinforcement learning can match full-model RL fine-tuning on reasoning benchmarks despite modifying only a tiny fraction of parameters (Sinii et al., 2025b,a). Together, these results suggest that reasoning-relevant adaptations may be concentrated in small representational subspaces, making adaptation capacity a critical factor independent of the training objective.

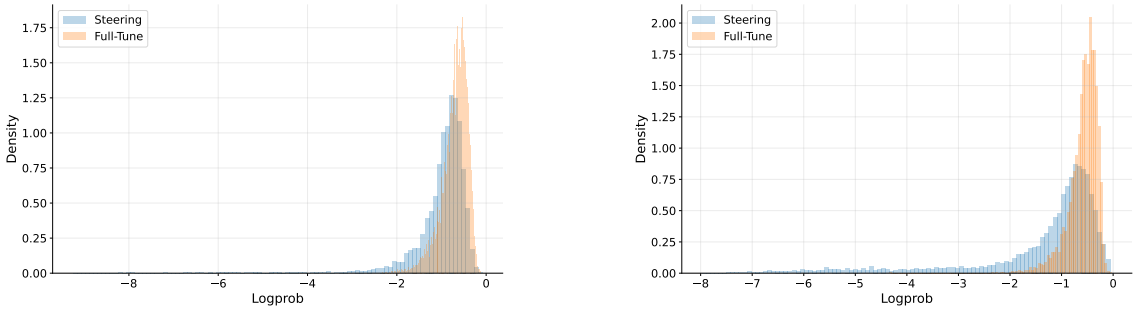
3 Steering Vectors Fail to Match Full Fine-Tuning under SFT

Setup To enable a direct comparison with the RL results of Sinii et al. (2025b), we adopted the same set of benchmarks, training setup, and steering vector implementation. We used the OpenThoughts-114k dataset (Open Thoughts Team, 2025) since it allowed both SFT and RL training. For SFT dataset, we extracted tasks within the mathematics domain and constructed each example by combining the original problem statement with the corresponding DeepSeek-R1 solution. We considered two base models in our study: Qwen2.5-Math-7B (Yang et al., 2024) and LLaMa3.1-8B-Instruct (Grattafiori et al., 2024). We trained these models for seven epochs, evaluated once per epoch and reported the best score. Our evaluation covered six mathematical reasoning benchmarks: AIME24/25, AMC23, MATH500 (Hendrycks et al., 2021), MinervaMath (Lewkowycz et al., 2022), and OlympiadBench (He et al., 2024). We report the mean score across these benchmarks in the main text and specify the raw scores in Section C. For MATH500, MinervaMath, and OlympiadBench we report PASS@1; for AIME24/25 and AMC23 we report AVG@32 due to their smaller sizes. All metrics are reported with mean and standard deviation evaluated over three seeds. For details, see Section A

We use reinforcement learning setup, based on online sampling with a binary correctness reward and optimization via the RLOO objective; full details are given in Section A.

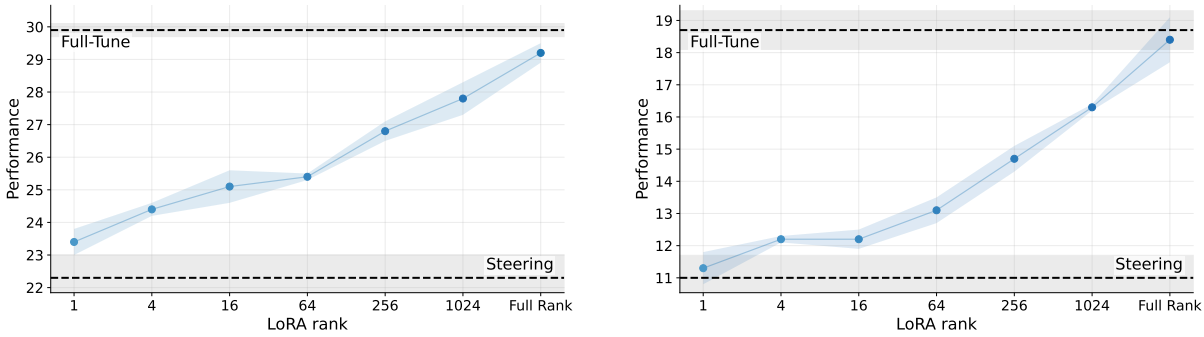
Results We first evaluated RL training on the OpenThoughts-114k dataset. Consistent with the findings of Sinii et al. (2025b), we observed that steering-based training achieved performance comparable to full fine-tuning across all evaluated benchmarks. This equivalence held for both Qwen2.5-Math-7B and LLaMa3.1-8B-Instruct, indicating that the result was robust to differences in model architecture. Figure 1 summarizes these RL results.

In contrast, this equivalence did not extend to the SFT setting. Figure 2 shows that trainable steering consistently underperformed full-model fine-tuning for both architectures. Figure 3 further shows that generations from steering models received lower likelihood under the base model, indicating a larger deviation from the base distribution. We conjecture that, because SFT weights all tokens equally, steering vectors are forced to fit many irrelevant tokens; this couples updates across the sequence and limits fine-grained, high-impact changes. Under RL, by contrast, steering and full-tuned models exhibit comparable deviation from the base model (Section B).



LLaMa3.1-8B-It Qwen2.5-Math-7B

Figure 3: Generations under steering vectors are lower-probability under the base model, when trained in SFT setup. Since SFT weights all tokens equally, the training constrains low-parameter steering vector from fine-grained updates and prohibits high accuracy. In contrast, full-tuning stays closer to the base model and does not disrupt generations.



Qwen2.5-Math-7B LLaMa3.1-8B-It

Figure 4: **Capacity scaling of MLP down-projection adapters under SFT.** Mean accuracy across all evaluation benchmarks as a function of LoRA rank when applied exclusively to the MLP down-projection. Performance increases monotonically with adapter rank, eventually almost matching the performance of full-tuning. This demonstrates that, under supervised fine-tuning, improvements in reasoning accuracy depend critically on adaptation capacity even when restricted to a fixed representational subspace.

4 Capacity Scaling in Supervised Fine-Tuning

To pinpoint the source of this discrepancy, we asked whether simply increasing the number of trainable parameters would close the gap – and, if so, how much capacity would be required. Rank-1 LoRA applied to the `MLP.down_proj` layer can be viewed as a gradual increase in the complexity of a steering vector, since it allows a token-specific magnitude. We therefore swept the LoRA rank at this location from 1 up to full rank, searching for the capacity threshold at which the gap disappears. Figure 4 shows that the gap closes only under full-rank training, which corresponds to training $\approx 27\%$ of parameters for Qwen2.5-Math-7B.

This result aligns with the analysis of Schulman and Lab (2025), who argue that SFT requires substantially more information (bits) to be learned, which can limit the effectiveness of small-parameter methods such as low-rank LoRA (Hu et al., 2022). They motivate the success of rank-1 LoRA in RL by contrasting trainable capacity with the amount of learning signal available: in RL, each generation produces a single scalar reward, so the signal scales as (dataset_size $\times$ num_generations), whereas in SFT it is spread across all tokens that contribute to the loss. In the next section, however, we show that with an appropriate data source, low-parameter steering vectors can still match full-tuning.

5 Changing Data Removes the Gap

In previous sections, we showed that trainable steering vectors fail to match full-model fine-tuning when trained with SFT. We also found that increasing training capacity can reduce this gap, but only at the cost of training an impractically large number of parameters. Here we show that, with appropriately chosen data, the capacity of steering vectors is in fact sufficient.

For each model, we construct a distillation dataset using its RL-trained counterpart: we sample 16 generations per prompt and select a single positive generation. We then train on these selected generations, distilling

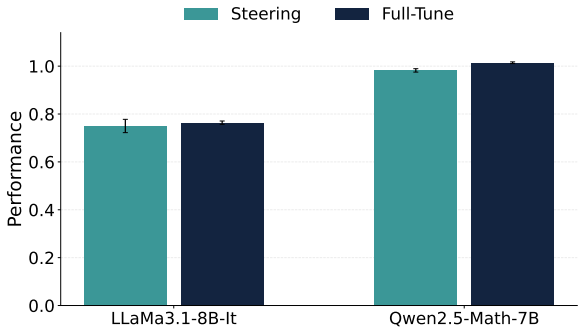


Figure 5: **Steering with Distillation.** When models are trained on generations produced by their respective RL-trained teachers, the performance gap between steering and full-model fine-tuning largely disappears. With an appropriate data source, low-parameter steering vectors have sufficient capacity to match full-tuning. The values are normalized between the base and RL models.

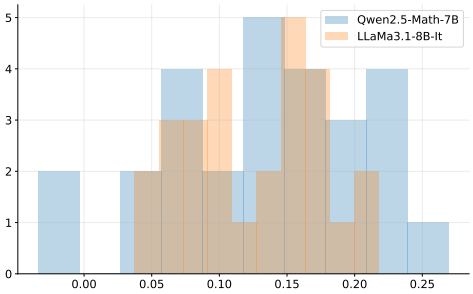


Figure 6: **Cosine similarity between steering vectors in RL and distilled models.** The steering vectors learned under RL and under distillation are poorly aligned, indicating that distillation does not simply copy the donor model's steering directions. Instead, it learns a different mechanism.

the knowledge encoded by RL into the student models.

Figure 5 shows that under this distillation setup the gap between steering and full-tuning closes. This indicates that steering vectors are not fundamentally capacity-limited; rather, their effectiveness depends on the training data. Importantly, distillation does not simply reproduce the RL-trained steering vectors. Figure 6 reports cosine similarities between steering vectors from the RL-trained and distilled models: across all layers, the vectors are poorly aligned, suggesting that distillation learns a different mechanism.

6 Out-of-Distribution Evaluation

Finally, we evaluated the models on out-of-distribution (OOD) benchmarks. We considered three tasks: GPQA (Rein et al., 2024), SynLogic (Liu et al., 2025), and IFEval (Zhou et al., 2023), and report the average score (with per-benchmark results in Section D). Figure 7 shows that full-model fine-tuning largely retains its OOD performance under both RL and SFT. In contrast, trainable steering vectors suffer a substantial drop. Taken together with our in-distribution results, this pattern suggests that steering vectors are less effective and less robust under SFT, exhibiting weaker generalization to distribution shifts.

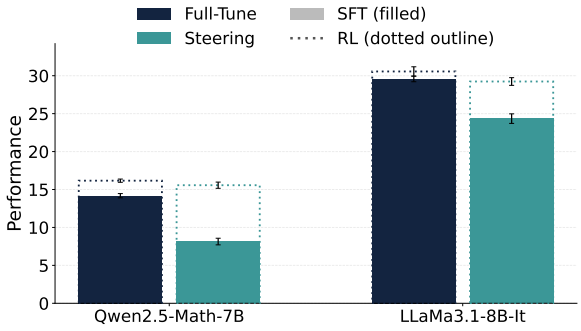


Figure 7: **Out-of-Distribution Evaluation.** Mean accuracy across three OOD benchmarks comparing trainable steering vectors with full-model fine-tuning. Under SFT, steering vectors exhibit a larger performance drop, indicating weaker generalization than full-model fine-tuning.

7 Limitations

Our experiments cover a narrow slice of online-training settings. Broader sweeps – across settings, tasks, and model families and sizes – would test generality of our observations.

8 Conclusion

We studied when trainable steering vectors can serve as a parameter-efficient alternative to full-model fine-tuning for reasoning. Our results show that their effectiveness depends strongly on the training signal and the data: in some settings, constrained steering updates are sufficient to reach full-tuning performance, while in others they fall short unless substantially more capacity is introduced. Overall, this work clarifies the conditions under which steering vectors are a viable substitute for full-model adaptation and highlights the central role of supervision quality and objective choice in making low-parameter training competitive.

References

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*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schel-

ten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

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*.

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

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. 2025. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *arXiv preprint arXiv:2503.24290*.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

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.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, and 1 others. 2024. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*, 13(9):9.

Junteng Liu, Yuanxiang Fan, Zhuo Jiang, Han Ding, Yongyi Hu, Chi Zhang, Yiqi Shi, Shitong Weng, Aili Chen, Shiqi Chen, and 1 others. 2025. Synlogic: Synthesizing verifiable reasoning data at scale for learning logical reasoning and beyond. *arXiv preprint arXiv:2505.19641*.

Kohsei Matsutani, Shota Takashiro, Gouki Minegishi, Takeshi Kojima, Yusuke Iwasawa, and Yutaka Matsuo. 2025. Rl squeezes, sft expands: A comparative study of reasoning llms. *arXiv preprint arXiv:2509.21128*.

Open Thoughts Team. 2025. Openthoughts-114k-math: A math-filtered subset of the openthoughts-114k reasoning dataset. https://huggingface.co/datasets/open-r1/OpenThoughts-114k-math. Filtered and verified math subset (≈ 89,120 examples) of the OpenThoughts-114K dataset.

Rahul Panickssery and Samuel R. Bowman. 2023. Steering llama 2 via contrastive activation addition. *arXiv preprint arXiv:2310.01405*.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

John Schulman and Thinking Machines Lab. 2025. Lora without regret. *Thinking Machines Lab: Connectionism*. Https://thinkingmachines.ai/blog/lora/.

Viacheslav Sinii, Nikita Balagansky, Gleb Gerasimov, Daniil Laptev, Yaroslav Aksenov, Vadim Kurochkin, Alexey Gorbatovski, Boris Shaposhnikov, and Daniil Gavrilov. 2025a. Small vectors, big effects: A mechanistic study of rl-induced reasoning via steering vectors. *arXiv preprint arXiv:2509.06608*.

Viacheslav Sinii, Alexey Gorbatovski, Artem Cherepanov, Boris Shaposhnikov, Nikita Balagansky, and Daniil Gavrilov. 2025b. Steering llm reasoning through bias-only adaptation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Alex Turner, Lukas Thiergart, Madeleine Udell, and 1 others. 2023. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. 2024. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. 2025. Limo: Less is more for reasoning. *arXiv preprint arXiv:2502.03387*.

Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. 2022. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. 2025. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *arXiv preprint arXiv:2503.18892*.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.

A SFT and RL Setups; Steering Vectors implementation

In RL training setup, for each prompt, we sample multiple candidate solutions from the current policy and assign a binary reward based on whether the generated answer is correct and enclosed in a `\boxed{...}` template. Optimization is performed using the RLOO objective.

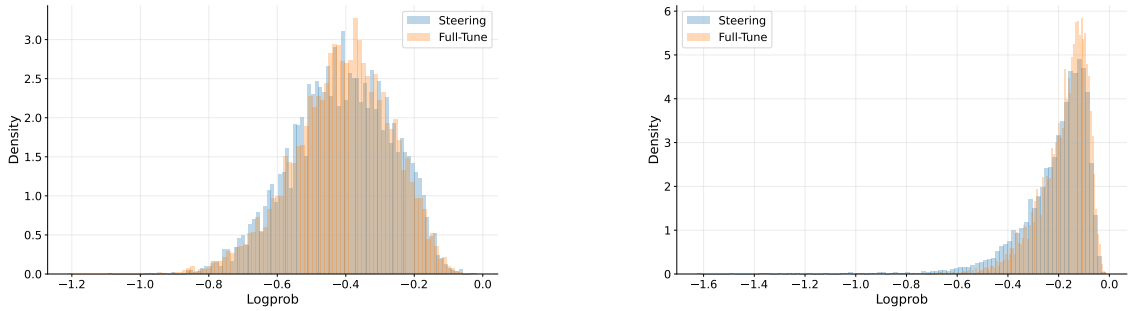
$$\nabla_\theta J = \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)}\big[\, a(x,y)\, \nabla_\theta \log \pi_\theta(y \mid x)\,\big],$$

All hyperparameters and evaluation procedures are identical to those used in the referenced work.

Steering vectors are implemented by enabling a learnable bias term in the MLP down-projection linear layer at the selected transformer block. Concretely, this corresponds to setting `bias=True` for the down-projection module and training only this bias parameter. All other model parameters are frozen throughout training. This parameterization exactly matches the steering vector setup of Sinii et al. (2025b) and ensures that all observed effects arise from a single additive direction in the residual stream.

During both training and evaluation in both RL and SFT setups, model generations were truncated to a maximum length of 4096 tokens for Qwen-based models and 8192 tokens for LLaMA-based models, respectively.

B Deviation from Base for RL-Trained Models



LLaMa3.1-8B-It Qwen2.5-Math-7B

Figure 8: RL-trained models exhibit similar deviation from their respective base models.

C Full Evaluation Results

In this section we provide all unnormalized accuracies for all evaluated models on each benchmark. All numbers are averaged over three evaluation seeds.

Table 1 reports SFT results on the OpenThoughts dataset, comparing steering vectors, LoRA adaptations, and full-model fine-tuning across all benchmarks.

Table 4 reports RL results on the OpenThoughts and DeepScaleR datasets, as well as base-model results that serve as the reference for scaling and analyzing SFT performance.

D Out of Domain evaluation Results

In this section, we report unnormalized accuracies for all evaluated models on each out-of-distribution (OOD) benchmark. All values correspond to AVG@8 scores, averaged over three evaluation seeds.

Table 3 reports SFT and RL results on the OOD benchmarks, comparing steering vectors and full-model fine-tuning.

Table 1: Full SFT OpenThoughts-114k per-benchmark results. AVG@32 denotes accuracy averaged over 32 sampled generations. All values are mean ± standard deviation across three seeds.

Model	Setup	AIME25@32	AIME24@32	AMC@32	MATH500	MinervaMath	Olympiad-Bench	Avg.
LLaMa3.1-8B-It	Full-Tune	0.5 ± 0.0	3.7 ± 0.0	22.9 ± 0.2	48.1 ± 2.8	21.7 ± 1.5	15.5 ± 0.5	18.7 ± 0.6
	Steering	0.3 ± 0.0	0.9 ± 0.0	12.6 ± 0.1	32.1 ± 2.5	12.0 ± 1.4	8.3 ± 0.8	11.0 ± 0.7
	MLP	0.6 ± 0.0	2.4 ± 0.1	25.2 ± 0.3	47.2 ± 2.3	20.6 ± 1.0	16.4 ± 0.7	18.7 ± 0.5
	MLP.down_proj	1.1 ± 0.0	3.1 ± 0.1	22.4 ± 0.1	46.5 ± 2.0	21.8 ± 1.4	15.2 ± 0.9	18.4 ± 0.7
	LoRA-1	0.3 ± 0.0	1.1 ± 0.1	11.3 ± 0.1	32.6 ± 1.9	14.7 ± 0.6	8.0 ± 1.2	11.3 ± 0.5
	LoRA-4	0.8 ± 0.0	1.3 ± 0.0	14.8 ± 0.0	32.7 ± 0.7	14.2 ± 1.4	9.1 ± 0.3	12.2 ± 0.1
	LoRA-16	0.3 ± 0.0	1.2 ± 0.0	15.3 ± 0.1	35.1 ± 1.1	12.7 ± 0.5	8.8 ± 0.3	12.2 ± 0.3
	LoRA-64	0.4 ± 0.1	1.6 ± 0.0	15.5 ± 0.4	37.0 ± 2.0	15.2 ± 0.2	8.9 ± 0.2	13.1 ± 0.4
	LoRA-256	0.6 ± 0.0	1.8 ± 0.0	17.8 ± 0.1	41.3 ± 0.6	16.2 ± 1.3	10.5 ± 1.0	14.7 ± 0.4
	LoRA-1024	0.6 ± 0.0	2.6 ± 0.1	19.4 ± 0.5	43.5 ± 1.6	17.9 ± 1.5	13.8 ± 0.5	16.3 ± 0.1
Qwen2.5-Math-7B	Full-Tune	3.5 ± 0.0	6.2 ± 0.1	44.5 ± 0.2	67.1 ± 1.0	30.4 ± 1.0	27.7 ± 1.7	29.9 ± 0.2
	Steering	2.5 ± 0.0	3.8 ± 0.0	32.9 ± 0.4	52.8 ± 1.5	21.3 ± 0.8	20.3 ± 2.1	22.3 ± 0.7
	MLP	5.5 ± 0.2	7.2 ± 0.2	43.9 ± 0.1	66.6 ± 0.7	29.4 ± 2.4	27.8 ± 0.9	30.1 ± 0.7
	MLP.down_proj	4.5 ± 0.1	5.7 ± 0.0	39.8 ± 0.3	55.5 ± 2.2	25.0 ± 0.8	25.7 ± 0.8	26.0 ± 0.2
	LoRA-1	3.7 ± 0.1	4.3 ± 0.1	31.1 ± 0.2	57.1 ± 1.5	21.3 ± 1.4	22.7 ± 0.7	23.4 ± 0.4
	LoRA-4	3.6 ± 0.0	6.8 ± 0.1	33.7 ± 0.2	56.9 ± 2.1	21.9 ± 0.8	23.5 ± 1.6	24.4 ± 0.2
	LoRA-16	4.1 ± 0.1	4.3 ± 0.1	34.8 ± 0.3	58.7 ± 1.6	26.8 ± 0.9	22.0 ± 0.9	25.1 ± 0.5
	LoRA-64	3.8 ± 0.2	5.0 ± 0.2	36.1 ± 0.3	59.2 ± 1.2	25.1 ± 1.5	23.3 ± 0.7	25.4 ± 0.1
	LoRA-256	4.3 ± 0.1	5.6 ± 0.1	37.8 ± 0.2	61.3 ± 0.5	27.1 ± 0.8	24.5 ± 1.2	26.8 ± 0.3
	LoRA-1024	5.4 ± 0.0	5.9 ± 0.1	40.6 ± 0.2	60.8 ± 1.1	28.6 ± 0.5	25.6 ± 2.0	27.8 ± 0.5

Table 2: Full RL OpenThoughts-114k and DeepScaler per-benchmark results. AVG@32 denotes accuracy averaged over 32 sampled generations. All values are mean ± standard deviation across three seeds.

Model	Dataset	Setup	AIME25@32	AIME24@32	AMC@32	MATH500	MinervaMath	Olympiad-Bench	Avg.
LLaMa3.1-8B-It	DeepScaler	Steering	1.5 ± 0.2	10.3 ± 0.2	32.4 ± 0.7	58.7 ± 0.8	29.3 ± 1.5	24.6 ± 1.3	26.1 ± 0.6
	DeepScaler	Full-Tune	1.6 ± 0.2	10.8 ± 0.2	35.3 ± 0.4	58.0 ± 0.7	29.4 ± 0.6	21.8 ± 0.8	26.1 ± 0.1
	OpenThoughts	Steering	1.4 ± 0.1	8.0 ± 0.0	32.4 ± 0.4	58.9 ± 0.2	28.8 ± 2.6	23.6 ± 0.5	25.5 ± 0.3
	OpenThoughts	Full-Tune	3.3 ± 0.1	6.2 ± 0.2	34.8 ± 0.3	58.1 ± 0.7	29.8 ± 1.1	24.2 ± 0.5	26.1 ± 0.3
		Base, $\tau = 1$	0.0 ± 0.0	0.0 ± 0.0	11.7 ± 2.4	34.6 ± 0.7	12.3 ± 1.5	9.6 ± 0.6	11.7 ± 0.5
Qwen2.5-Math-7B	DeepScaler	Steering	12.6 ± 0.2	24.4 ± 0.2	62.5 ± 0.2	79.9 ± 1.2	36.0 ± 2.8	44.1 ± 0.8	43.3 ± 0.3
	DeepScaler	Full-Tune	14.0 ± 0.1	25.7 ± 0.1	64.2 ± 0.1	79.3 ± 0.6	36.9 ± 2.0	41.1 ± 0.9	43.5 ± 0.4
	OpenThoughts	Steering	13.3 ± 0.1	21.8 ± 0.2	62.2 ± 0.1	79.3 ± 0.7	35.5 ± 1.0	42.2 ± 0.4	42.4 ± 0.0
	OpenThoughts	Full-Tune	11.1 ± 0.2	23.9 ± 0.1	63.3 ± 0.3	79.1 ± 1.1	35.3 ± 0.9	41.9 ± 0.8	42.4 ± 0.2
		Base, $\tau = 1$	2.2 ± 1.6	10.0 ± 4.7	25.0 ± 8.2	37.7 ± 6.1	8.3 ± 1.5	10.2 ± 2.9	14.3 ± 1.7

Table 3: Evaluation results across IFEval, GPQA, and SynLogic.

Model	Method	Setup	IFEval	GPQA	Synlogic	Avg.
Qwen2.5-Math-7B	RL	Full-Tune	16.2 ± 0.04	25.0 ± 0.01	7.3 ± 0.02	16.2 ± 0.02
Qwen2.5-Math-7B	RL	Steering	17.1 ± 0.02	21.4 ± 0.08	8.2 ± 0.03	15.6 ± 0.04
LLaMa3.1-8B-It	RL	Full-Tune	59.9 ± 0.05	23.5 ± 0.10	8.3 ± 0.03	30.6 ± 0.06
LLaMa3.1-8B-It	RL	Steering	60.3 ± 0.04	19.8 ± 0.10	7.7 ± 0.02	29.2 ± 0.05
Qwen2.5-Math-7B	SFT	Full-Tune	11.7 ± 0.02	26.7 ± 0.05	4.2 ± 0.02	14.2 ± 0.03
Qwen2.5-Math-7B	SFT	Steering	10.6 ± 0.02	10.8 ± 0.10	2.9 ± 0.01	8.1 ± 0.04
LLaMa3.1-8B-It	SFT	Full-Tune	54.4 ± 0.03	30.3 ± 0.06	4.0 ± 0.03	29.6 ± 0.04
LLaMa3.1-8B-It	SFT	Steering	48.8 ± 0.05	21.7 ± 0.11	2.6 ± 0.03	24.3 ± 0.06
Qwen2.5-Math-7B	-	Base, $\tau = 1$	11.5 ± 0.02	7.2 ± 0.07	2.5 ± 0.03	7.1 ± 0.04
LLaMa3.1-8B-It	-	Base, $\tau = 1$	59.1 ± 0.10	4.8 ± 0.03	4.7 ± 0.02	22.9 ± 0.05
Qwen2.5-Math-7B	-	Base, $\tau = 0$	17.2 ± 0.00	9.1 ± 0.00	4.4 ± 0.00	10.3 ± 0.00
LLaMa3.1-8B-It	-	Base, $\tau = 0$	66.7 ± 0.00	7.8 ± 0.00	6.0 ± 0.00	26.8 ± 0.00

Table 4: Full RL OpenThoughts-114k and DeepScaler per-benchmark results. AVG@32 denotes accuracy averaged over 32 sampled generations. All values are mean $\pm$ standard deviation across three seeds.

Model	Dataset	Setup	AIME25@32	AIME24@32	AMC@32	MATH500	MinervaMath	Olympiad-Bench	Avg.
LLaMa3.1-8B-It	DeepScaler	Steering	1.5 ± 0.2	10.3 ± 0.2	32.4 ± 0.7	58.7 ± 0.8	29.3 ± 1.5	24.6 ± 1.3	26.1 ± 0.6
	DeepScaler	Full-Tune	1.6 ± 0.2	10.8 ± 0.2	35.3 ± 0.4	58.0 ± 0.7	29.4 ± 0.6	21.8 ± 0.8	26.1 ± 0.1
	OpenThoughts	Steering	1.4 ± 0.1	8.0 ± 0.0	32.4 ± 0.4	58.9 ± 0.2	28.8 ± 2.6	23.6 ± 0.5	25.5 ± 0.3
	OpenThoughts	Full-Tune	3.3 ± 0.1	6.2 ± 0.2	34.8 ± 0.3	58.1 ± 0.7	29.8 ± 1.1	24.2 ± 0.5	26.1 ± 0.3
		Base, $\tau = 1$	0.0 ± 0.0	0.0 ± 0.0	11.7 ± 2.4	34.6 ± 0.7	12.3 ± 1.5	9.6 ± 0.6	11.7 ± 0.5
Qwen2.5-Math-7B	DeepScaler	Steering	12.6 ± 0.2	24.4 ± 0.2	62.5 ± 0.2	79.9 ± 1.2	36.0 ± 2.8	44.1 ± 0.8	43.3 ± 0.3
	DeepScaler	Full-Tune	14.0 ± 0.1	25.7 ± 0.1	64.2 ± 0.1	79.3 ± 0.6	36.9 ± 2.0	41.1 ± 0.9	43.5 ± 0.4
	OpenThoughts	Steering	13.3 ± 0.1	21.8 ± 0.2	62.2 ± 0.1	79.3 ± 0.7	35.5 ± 1.0	42.2 ± 0.4	42.4 ± 0.0
	OpenThoughts	Full-Tune	11.1 ± 0.2	23.9 ± 0.1	63.3 ± 0.3	79.1 ± 1.1	35.3 ± 0.9	41.9 ± 0.8	42.4 ± 0.2
		Base, $\tau = 1$	2.2 ± 1.6	10.0 ± 4.7	25.0 ± 8.2	37.7 ± 6.1	8.3 ± 1.5	10.2 ± 2.9	14.3 ± 1.7