

000 001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 TIMBER: TRAINING-FREE INSTRUCT MODEL REFINING WITH BASE MODEL VIA EFFECTIVE RANK

Anonymous authors

Paper under double-blind review

ABSTRACT

Post-training, which elicits a pretrained Base model into the corresponding Instruct model, is widely considered to be superficial. In this work, we first reinforce this hypothesis by providing novel quantitative evidence from the weight level that the effective rank (eRank) remains negligibly changed. However, this superficiality also suffers a critical trade-off, improving the exploitation capabilities at the cost of limiting its exploration. To tackle this issue, we propose Timber, a simple yet effective *training-free* method that enhances the exploration capability of the Instruct model while preserving its exploitation. The key insight is to partially revert Instruct towards the paired Base model by subtle yet targeted refinement of the weight deltas. Extensive experiments on Llama and Qwen series demonstrate that Timber consistently improves vanilla Instruct models, particularly on Pass@k performance. Our findings offer new insights into the post-training stage at the weight level and practical strategies to refine the Instruct model *without* training.

1 INTRODUCTION

Large Language Models (LLMs), such as Qwen3 (Yang et al., 2025), Llama 3 (Grattafiori et al., 2024), and Deepseek R1 (Guo et al., 2025), have achieved superior success in Natural Language Processing (NLP), especially in reasoning tasks (Huang & Chang, 2023). To train these LLMs, a Base model is first pretrained on huge amounts of data. After that, a post-training stage is applied to train an Instruct model, adopting supervised finetuning (SFT) and reinforcement learning (RL) to elicit alignment and reasoning ability (Yang et al., 2025). The post-training stage tends to be superficial, i.e., post-training only utilizes the pattern contained in the Base model acquired during pre-training (Yue et al., 2025; Zhou et al., 2023a; Ye et al., 2025; Muennighoff et al., 2025).

In this paper, we investigate the Base and Instruct models through the lens of effective rank (eRank, (Roy & Vetterli, 2007)), providing a novel weight-level perspective on the superficiality of post-training. Specifically, eRank quantifies the effective dimensionality of a weight matrix by measuring the uniformity of its singular value distribution, reflecting the intrinsic representational capacity (Schumacher, 1995; Wei et al., 2024). As shown in Figure 1, the eRanks of corresponding linear layers from the Base and Instruct models are almost identical. We can find that post-training induces only negligible changes to the effective dimensionality, offering new supporting evidence from the weight level for its superficiality (Wu et al., 2025a).

However, such superficiality of post-training also suffers a critical trade-off between exploitation and exploration. Specifically, while the Instruct model achieves higher Pass@1 in reasoning tasks, it lags behind on Pass@k for relatively large k (Wang et al., 2025a; Yue et al., 2025; Zhu et al., 2025). In summary, superficial post-training suppresses the sampling space and thereby limits the performance potential. While recent works have sought to mitigate this limitation by introducing additional training objectives (Chen et al., 2025) or external tools (Wang et al., 2025a), these methods invariably incur significant overhead during training or inference.

To this end, we propose **Timber**, a simple yet effective training-free method to enhance an Instruct model using its paired Base model at the weight level. Inspired by the model merge (Yang et al., 2024; Zhang et al., 2024), our key design is to refine the weight delta between the two models, partially reverting the Instruct model towards its Base state to improve exploration ability. Specifically, our method first decomposes the weight delta using SVD and identifies the head and tail components

of the singular values via eRank. Subsequently, the tail components are either removed or attenuated. Through such fine-grained refinement, Timber achieves a better trade-off between exploitation and exploration.

We evaluate the proposed Timber on models including Llama and Qwen series across a suite of popular benchmarks. Experimental results demonstrate that Timber consistently outperforms the vanilla Instruct models, confirming its effectiveness and robustness. Further analysis reveals that this performance gain is largely attributed to a significant improvement in Pass@k scores, which underscores the ability of Timber to enhance the exploration capabilities. Our contributions can be concluded as follows:

- We propose to revisit paired Base and Instruct models via eRank, providing a more granular understanding of post-training superficiality at the weight level.
- We propose Timber, a simple yet effective training-free method that enhances the exploration capability of the Instruct model while preserving its exploitation. The key is to partially revert the Instruct model towards its Base via refining the weight delta.
- We demonstrate the effectiveness and robustness of the proposed Timber via results on various LLMs and benchmarks.

2 PRELIMINARY AND ANALYSIS

2.1 BACKGROUND

Superficial Post-training. Typically, the training of LLM follows a two-stage pipeline (Yang et al., 2025). The first step is to build a Base model with rich knowledge by pretraining on a large amount of training data. After that, we perform post-training on Base to elicit the instruction following and reasoning abilities through the SFT and RL (Shao et al., 2024). Due to the mess of naming, we detailed paired Base and Instruct models in Table 1.

Recent work finds that such a post-training process is superficial (Zhou et al., 2023a; Ye et al., 2025; Ji et al., 2025; Wu et al., 2025a). Superficial Alignment Hypothesis claims that the model’s knowledge and capabilities are acquired almost entirely during pretraining, while alignment teaches it which sub-distribution of formats should be used when interacting with users (Zhou et al., 2023a; Ye et al., 2025). Model Elasticity finds that models tend to maintain the original distribution, i.e., resist alignment and return quickly when tuned in the opposite direction (Ji et al., 2025). Shadow-FT represents that paired Base and Instruct are highly similar in weights (Wu et al., 2025a). Compared to these previous works, we revisit Base and Instruct models from the view of effective rank, providing a novel perspective on the superficiality.

Effective Rank. Effective rank (eRank) measures the uniformity of the singular value distribution to quantify the effective dimensionality of a matrix (Roy & Vetterli, 2007). For any non-zero matrix $\mathbf{W} \in \mathbb{R}^{d_1 \times d_2}$ with singular values $\Sigma = \{\sigma_1, \sigma_2, \sigma_3, \dots, \sigma_{r-1}, \sigma_r\}$ where $r = \min\{d_1, d_2\}$. The eRank of \mathbf{W} is defined as the exponential of the Shannon entropy computed from its normalized singular value distribution, formulated as follows:

$$\text{eRank}(\mathbf{W}) = \exp \left(- \sum_{i=1}^r \frac{\sigma_i^\gamma}{\sum_{i=1}^r \sigma_i^\gamma} \log \left(\frac{\sigma_i^\gamma}{\sum_{i=1}^r \sigma_i^\gamma} \right) \right). \quad (1)$$

The γ is the scale factor and can be 1 or 2, typically. In this paper, we set $\gamma = 1$. Meanwhile, we can calculate the entropy using $\log_2(x)$ and then apply 2^x instead of $\exp(x)$ in Equation 1. We can easily prove that the eRank would be exactly the same.

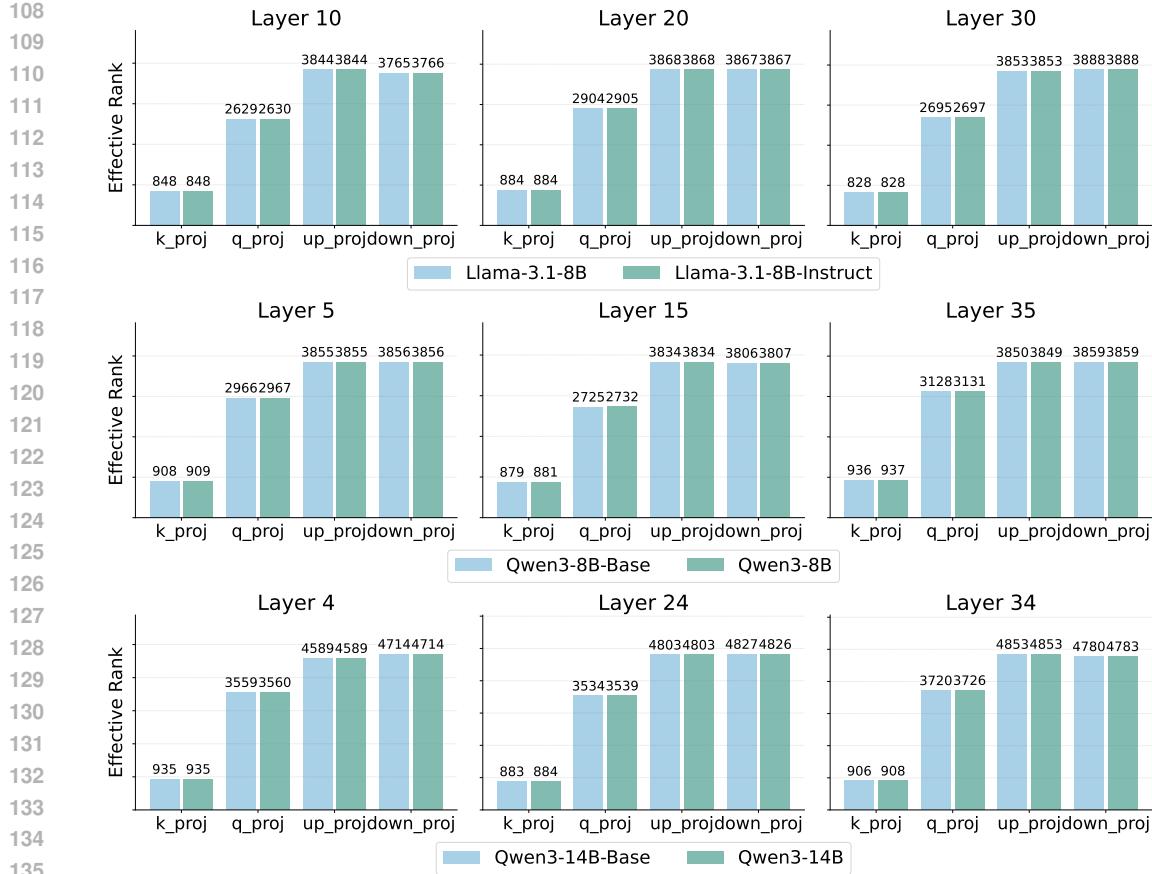


Figure 1: Effective ranks (eRank) of the linear weights from various paired Base and Instruct models. We randomly select three layers, and k_proj is relatively small due to the Grouped-Query Attention (GQA) mechanism. We can find that eRanks from paired Base and Instruct models are almost the same.

Effective rank plays an important role in measuring the information among hidden states. Diff-eRank assesses LLMs by analyzing hidden representations and measuring how efficiently LLMs eliminate redundant information during training (Wei et al., 2024). Li et al. (2025) employs the eRank of gradients to assess the quality of training data. To the best of our knowledge, we are the first to directly analyze the eRank of weights.

2.2 REVISIT BASE AND INSTRUCT VIA EFFECTIVE RANK

To investigate the effects of post-training, we examine the weights of Base and Instruct models through the lens of effective rank (eRank). Our analysis covers several mainstream LLMs, including the Llama and Qwen3 series. Without loss of generality, we randomly select three representative linear layers from the bottom, middle, and top. Due to space constraints, we report the ceiling of the eRank values.

Figure 1 shows the paired eRank distributions. The core finding is that the eRank values for corresponding linear layers in the Base and Instruct models are nearly identical. For instance, the k_proj matrix in Layer 4 of Qwen3-14B has an eRank of 935 in both Base and Instruct versions. This striking similarity holds across all models and layers tested, demonstrating the robustness of this phenomenon.

Given that eRank quantifies the effective dimensionality of a weight matrix, our results indicate that this dimensionality remains almost unchanged after post-training. Therefore, we can conclude that

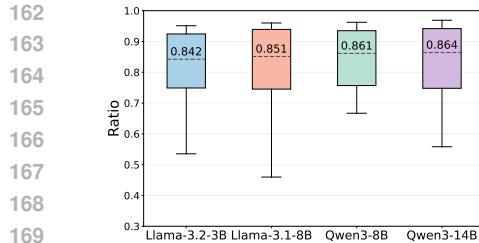


Figure 2: The distribution of eRank-to-Rank ratios for all linear layers in various LLMs.

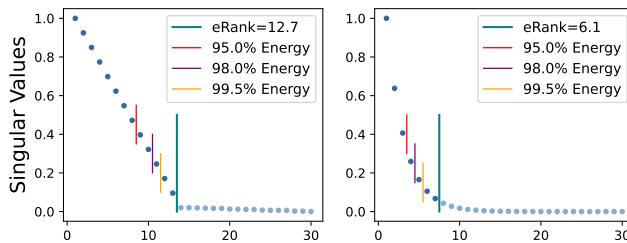


Figure 3: Toy examples of singular value distributions and thresholds using eRank or energy. eRank works well as an adaptive threshold.

the knowledge acquired during pre-training is retained, reinforcing the hypothesis that post-training is a superficial process.

We further analyze the distribution of the eRank-to-Rank ratio, defined as $e\text{Rank}/r$. By definition (Equation 1, $1 \leq e\text{Rank} \leq r$), this ratio is constrained to the interval $(0,1]$. As illustrated in Figure 2, the ratios are highly concentrated. For all evaluated models, the mean ratio remains stable at approximately 0.85, and the interquartile range consistently falls between 0.75 and 0.95. This suggests that eRank is consistently a high fraction of the total rank.

3 METHODOLOGY

3.1 MOTIVATION

Challenge. The post-training phase also introduces a critical trade-off between exploitation and exploration. During this stage, the Instruct model is optimized to maximize rewards by focusing on the most effective reasoning paths, thereby sharpening its exploitative capabilities. However, this intense focus comes at the cost of its ability to explore a diverse range of solutions. This trade-off is empirically evident in model performance. Specifically, Instruct models significantly outperform their base counterparts on Pass@1, but tend to underperform on Pass@k for larger values of k (Yue et al., 2025; Wang et al., 2025a; Zhu et al., 2025). Therefore, how to enhance the exploration without compromising its exploitation remains a challenge.

Refine Instruct with Base. Recent works have sought to mitigate this limitation by introducing additional training objectives (Chen et al., 2025) or external tools (Wang et al., 2025a). These methods, however, invariably incur significant computational overhead during training or inference. In this paper, we focus on the training-free method to circumvent this issue.

Given that the post-training process is superficial at the weight level (Wu et al., 2025a), one intuitive idea is to enhance Instruct with the weights from the Base model. Such a training-free strategy has been validated on related tasks such as model merge (Yang et al., 2024; Zhang et al., 2024; Wu et al., 2025b). The Base model contains almost all the knowledge, while the Instruct model only elicits part of the high-reward thinking patterns. **As the post-training shrinks the generative horizon (Yang & Holtzman, 2025), our solution is to partially revert the Instruct model towards its Base state.** Therefore, our next goal is to refine the weight deltas between the Instruct and Base models.

3.2 TIMBER

To refine the weight deltas, one naive way is to scale them linearly. However, the modifications from post-training are known to be fragile (Ji et al., 2025), and this simple scaling often fails (see Section 5.2). Fortunately, eRank measures the effective dimension of the matrix and **suitable for indicating the majority of the singular values**. For instance, eRank would be 1 for singular values $\Sigma = \{1, 0, \dots, 0, 0\}$ and be r for $\Sigma = \{1, 1, \dots, 1, 1\}$. As illustrated in the toy examples in Figure 3, eRank serves as an effective threshold for isolating the principal components of the singular value spectrum.

Motivated by this property, we propose a simple yet effective training-free method named **Timber**. The core idea is to enhance the weight delta via partially reverting the Instruct model towards its Base state. Specifically, we employ eRank as a threshold to partition the singular values (i.e., the matrix spectrum) of the weight delta into head and tail parts, and then either remove or attenuate the tail.

For the weight matrices $\mathbf{W}_B \in \mathbb{R}^{m \times n}$ and $\mathbf{W}_I \in \mathbb{R}^{m \times n}$ from the same linear layer of Base and Instruct models, we first compute the weight delta:

$$\mathbf{W}_\Delta = \mathbf{W}_I - \mathbf{W}_B. \quad (2)$$

The weight delta \mathbf{W}_Δ is typically full-rank and rank $r = \min\{m, n\}$. We then calculate the singular values via SVD decomposition:

$$\text{SVD}(\mathbf{W}_\Delta) \rightarrow \mathbf{U}\Sigma\mathbf{V}^T, \quad (3)$$

where \mathbf{U} and \mathbf{V}^T are two orthogonal matrices and $\Sigma = \text{diag}(\sigma_0, \sigma_1, \dots, \sigma_{r-1}, \sigma_r)$ contains the singular values in non-decreasing order. Our goal is to create a refined weight matrix \mathbf{W}_I^+ by modifying these singular values:

$$\mathbf{W}_I^+ = \mathbf{W}_B + \text{refine}(\mathbf{W}_\Delta) = \mathbf{W}_B + \mathbf{U} \text{refine}(\Sigma) \mathbf{V}^T, \quad (4)$$

where $\text{refine}(\cdot)$ is the enhancement operation to the singular values in Σ .

In Timber, we define this refinement process as follows. First, we set a threshold K based on the ceiling of the eRank:

$$K := \lceil \text{eRank}(\mathbf{W}_\Delta) \rceil. \quad (5)$$

One strategy is to remove the tail part by zeroing out singular values beyond the K -th position:

$$\text{refine}(\Sigma) = \text{diag}\{\underbrace{\sigma_1, \sigma_2, \dots, \sigma_K}_{\text{Top-K,preserve}}, \underbrace{0, \dots, 0, 0}_{\text{discard}}\}. \quad (6)$$

Since this operation lowers the rank of the weight delta, we name this strategy **Timber-L**. As shown in Figure 2, the eRanks are around the 85th percentile of full rank.

Another strategy is to attenuate the tail part rather than discard it entirely:

$$\text{refine}(\Sigma) = \text{diag}\{\underbrace{\sigma_1, \sigma_2, \dots, \sigma_K}_{\text{Top-K,preserve}}, \underbrace{\lambda \cdot \sigma_{K+1}, \dots, \lambda \cdot \sigma_{r-1}, \lambda \cdot \sigma_r}_{\text{attenuate}}\}, \quad (7)$$

where $0 < \lambda < 1$ is an attenuation factor. This full-rank strategy is referred to as Timber. Note that Timber-L is a special case of this approach where $\lambda = 0$. When applying Timber, we only modify the weights of linear layers, leaving bias terms and normalization layers unchanged.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Models. We conduct experiments on mainstream LLMs, specifically the Llama 3 and Qwen3 series, with model sizes ranging from 0.6B to 30B. In particular, we also include the MoE-style Qwen3-30B-A3B for a more comprehensive setting. All model checkpoints are downloaded from the official HuggingFace repository. Detailed model information is provided in Table 1. For Timber, we search for the best attenuation factor in $\{0.2, 0.5, 0.8\}$ based on the performance on AIME’24. This search incurs a minimal computational cost as Timber is a **training-free** method. Please refer to Appendix A.4 for the detailed score with different λ .

Evaluation. We evaluate the models on a suite of mainstream benchmarks spanning various tasks: IFEval (Zhou et al., 2023b) for the instruction following, MATH (Hendrycks et al., 2021) and MATH-500 (Lightman et al., 2023) for mathematical reasoning, GPQA-Diamond (GPQA-D, (Rein et al., 2024)) for scientific question answering, and HellaSwag (Zellers et al., 2019) for common-sense reasoning. To assess the Qwen3 series in Thinking mode, we also utilize the challenging AIME’24¹ and HumanEval (Chen et al., 2021) for coding task. For all models, we use the officially recommended hyperparameters for inference. Further details on the benchmarks and evaluation settings can be found in Appendix A.3.

¹<https://huggingface.co/datasets/AI-MO/aimo-validation-aime>

270
 271 Table 2: Performance of vanilla Instruct, proposed Timber-L, and Timber on mainstreaming benchmarks
 272 regarding Llama and Qwen3 series. The Qwen3 models are evaluated in *Non-thinking* mode.
 273 Under all the settings, Timber-L and Timber outperform the baseline **without any training**.

Model	Setting	IFEval	MATH-500	MATH	GPQA-D	HellaSwag	Avg.	Δ
Llama-3.2-1B	Instruct	48.34	14.85	16.88	22.98	34.94	27.60	
	Timber-L	49.17	15.20	16.77	25.88	35.77	28.56	+0.96
	Timber	49.58	14.95	17.08	24.62	35.35	28.32	+0.72
Llama-3.2-3B	Instruct	69.64	41.35	34.22	24.75	61.01	46.19	
	Timber-L	68.81	41.70	34.48	25.76	61.29	46.41	+0.22
	Timber	69.59	42.05	34.23	25.51	61.46	46.57	+0.38
Llama-3.1-8B	Instruct	74.25	49.60	39.46	28.62	75.65	53.52	
	Timber-L	73.81	51.13	39.21	28.96	81.36	54.89	+1.37
	Timber	75.66	50.67	39.35	28.11	82.37	55.23	+1.71
Qwen3-0.6B	Instruct	56.75	52.00	41.92	23.74	43.42	43.57	
	Timber-L	56.75	52.60	42.66	23.40	43.27	43.74	+0.17
	Timber	57.36	51.60	43.11	28.79	43.30	44.83	+1.26
Qwen3-8B	Instruct	82.75	84.20	73.27	45.62	84.62	74.09	
	Timber-L	83.46	84.00	73.21	46.63	84.67	74.39	+0.30
	Timber	83.09	84.73	73.23	48.48	84.68	74.84	+0.75
Qwen3-14B	Instruct	85.15	87.47	75.74	47.81	88.13	76.86	
	Timber-L	85.77	86.60	75.97	51.68	87.97	77.60	+0.74
	Timber	85.46	88.00	75.77	51.01	88.04	77.66	+0.80
Qwen3-30B-A3B	Instruct	84.17	88.00	76.99	44.95	89.25	76.67	
	Timber-L	85.71	88.33	77.00	45.12	89.43	77.12	+0.45
	Timber	85.09	88.80	76.85	45.96	89.27	77.19	+0.52

296 **Metric.** To evaluate the exploration capability, we use the popular Pass@k, which is defined as
 297 the fraction of problems for which at least one correct response is produced in k independent trials.
 298 However, directly computing Pass@k using only k rollouts per problem often suffers from high
 299 variance. Therefore, we employ the unbiased estimator (Chen et al., 2021). Specifically, we roll out
 300 for n times ($n \geq k$), and calculate Pass@k as follows:

$$\text{Pass}@k := \mathbb{E}_{x \sim \mathcal{D}} \left[1 - \frac{\binom{n-c}{k}}{\binom{n}{k}} \right], \quad (8)$$

304 where x is the input prompt from dataset D , and c is the count of correct solutions. In addition to
 305 Pass@k, we also report Mean@k, defined as the average accuracy across k independent trials. We
 306 repeat 4 times for Llama-3.2-1B, and 3 times for the rest larger models.

4.2 MAIN RESULTS

309 As shown in Table 2, we report the Mean@k results on 6 benchmarks and their average score. **We**
 310 **repeat 4 trials for Llama-3.2-1B (k=4) and 3 for others (k=3).** Some findings can be concluded as
 311 follows:

- 313 • *Our proposed Timber consistently and comprehensively outperforms the vanilla Instruct
 314 model.* Across all tested models, both Timber-L and Timber significantly outperform the
 315 vanilla Instruct model. For instance, Timber achieves an average score of 55.23 for Llama-
 316 3.1-8B, which is 1.71 higher than baseline.
- 317 • *The attenuation strategy of Timber is generally superior to directly dropping in Timber-L.* When
 318 comparing the two proposed variants, the standard Timber method demonstrates a
 319 greater performance gain (i.e., Δ) than Timber-L in 6 out of 7 cases. For instance, on
 320 Qwen3-0.6B, Timber gets an average of 44.83, significantly higher than 43.74 of Timber-L.
 321 This suggests that Timber strikes a better balance between optimization and knowledge
 322 preservation.
- 323 • *Timber is a robust and broadly applicable training-free plug-in.* The performance benefits
 324 of Timber are not limited to a specific model family, size, or architecture. The method

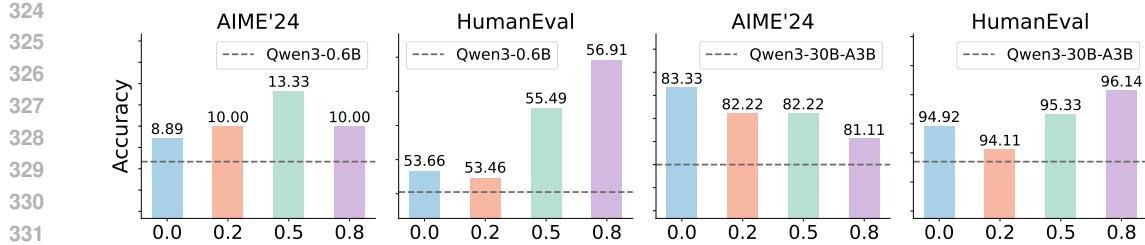


Figure 4: Accuracy on AIME’24 and HumanEval benchmarks for Timber with various λ . The vanilla scores for Instruct models are 6.67, 53.05, 80.0, and 93.7, respectively. For both models, we sample the results under *Thinking* mode. Timber shows strong robustness regarding λ .

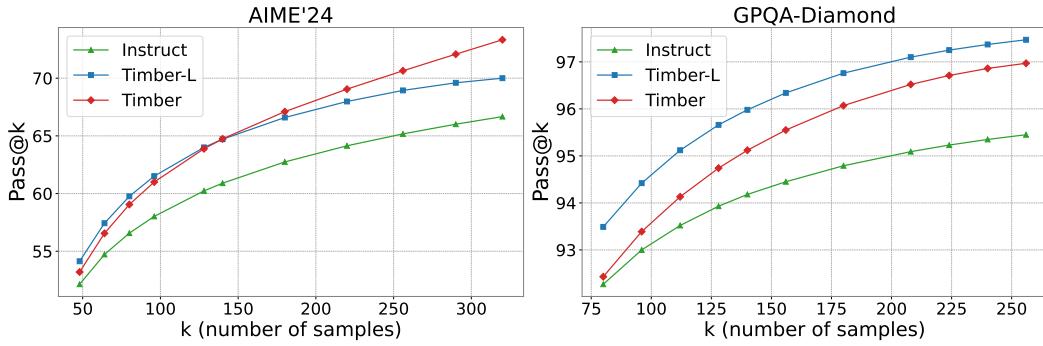


Figure 5: Pass@k results on AIME’24 and GPQA-Diamond benchmarks for Qwen3-0.6B under Thinking mode. Both Timber and Timber-L improve the exploration significantly. [Please refer to Appendix A.4 for more detailed results.](#)

proves effective on both the Llama 3 and Qwen3 series and scales from small 0.6B models to the large 30B Mixture-of-Experts (MoE) architecture.

The Qwen3 series supports the hybrid thinking, allowing them to generate outputs in either Thinking or Non-Thinking mode. In the Thinking mode, the LLM will output a longer reasoning process and typically performs better thanks to the test-time scaling. To assess their advanced reasoning capabilities, we further evaluate the thinking capability of Qwen3-0.6B and Qwen3-30B-A3B on the AIME’24 and HumanEval tasks. As shown in Figure 4, our proposed method, Timber, consistently outperforms the vanilla Instruct model across various attenuation factors λ , demonstrating both superior effectiveness and robustness. For instance, when applied to Qwen3-30B-A3B, Timber achieves a score of 96.14 on HumanEval, surpassing the vanilla Instruct model by 2.44 points.

The core principle of Timber is to partially revert the Instruct model towards its Base state, a process designed to enhance exploration without compromising exploitation. To validate this hypothesis, we conducted further experiments on Qwen3-0.6B, evaluating its exploration performance using the Pass@k metric. During inference, we configured the model to rollout in Thinking mode, generating 320 candidate samples for AIME’24 and 256 for GPQA-Diamond. The results, shown in Figure 5, indicate that both Timber and Timber-L achieve significantly higher Pass@k scores than the vanilla Instruct model.

Critically, the performance gap between the Timber methods and the Instruct baseline widens as k increases. This trend provides strong evidence that Timber is fundamentally more effective at exploring the solution space and generating a diverse set of high-quality candidates. In summary, our proposed *training-free* method, Timber, successfully enhances exploration without compromising exploitation, a conclusion supported by the comprehensive results in Table 2 and Figure 4.

378 Table 3: Performance for Truncated SVD methods and proposed Timber on Qwen3-8B. Truncate-R
 379 denotes the strategy on the ratio of full rank, and Truncated-E on the ratio of energy.
 380

Method	Energy	IFEval	MATH-500	MATH	GPQA-D	HellaSwag	Avg.
Instruct	100.00%	82.75	84.20	73.27	45.62	84.62	74.09
Truncate-R	99.60%	83.30	85.07	73.21	44.95	84.74	74.25
	99.04%	83.18	83.87	73.08	47.47	84.69	74.46
	98.30%	82.50	85.00	72.99	44.11	84.62	73.84
	97.39%	83.43	84.60	73.56	46.63	84.70	74.58
	99.50%	83.30	84.27	72.99	44.28	84.73	73.91
Truncate-E	99.00%	82.62	84.33	72.73	45.96	84.72	74.07
	98.00%	84.47	84.47	72.93	44.95	84.76	74.32
	95.00%	83.49	83.87	73.23	44.95	84.87	74.08
	98.82%	83.09	84.73	73.23	48.48	84.68	74.84

5 EXTENSIVE ANALYSIS

5.1 COMPARISON WITH TRUNCATED SVD

399 Truncated SVD is a widely employed technique for compressing Large Language Models
 400 (LLMs) (Wang et al., 2025b; Qinsi et al., 2025). While our method is conceptually different, Timber-
 401 L can be interpreted as a special case of SVD applied to the weight deltas, where truncation occurs
 402 at the effective rank (eRank). However, Timber differs fundamentally: instead of discarding the tail
 403 singular values, it attenuates them with a scaling factor.

404 For further comparison, despite these theoretical differences, we design two SVD truncation base-
 405 lines: Truncate-R and Truncate-E. Truncate-R discards singular values based on a fixed ratio of the
 406 full rank, while Truncate-E based on a target energy preservation ratio. The energy of the singular
 407 value distribution is a vital metric representing the amount of preserved information. Therefore,
 408 we set the ratios in Truncate-R and Truncate-E to be comparable with our proposed method, Tim-
 409 ber. Specifically, Timber preserves 98.82% of the total energy on Qwen3-8B. For Truncate-R, we
 410 thus set the rank ratios to 0.95, 0.9, 0.85, and 0.8, which correspond to preserving 99.60%, 99.04%,
 411 98.30%, and 97.39% of the total energy, respectively. For Truncate-E, we set the energy preservation
 412 thresholds directly to 99.50%, 99.00%, 98.00%, and 95.00%.

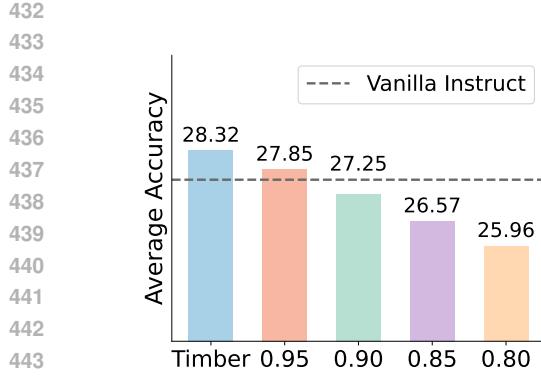
413 Table 3 details the performance of these methods on benchmarks. We can find that our proposed
 414 Timber method achieves the highest average score (74.84), outperforming all variants of the Trun-
 415 cated SVD baselines. While both Truncate-R and Truncate-E can outperform the Instruct model,
 416 the performance is unstable and highly sensitive to the truncation threshold. In contrast, Timber
 417 provides a more substantial and robust performance gain, suggesting that eRank serves as a solid
 418 threshold and attenuating the tail singular values is more effective than discarding.

5.2 COMPARISON WITH MODEL MERGE

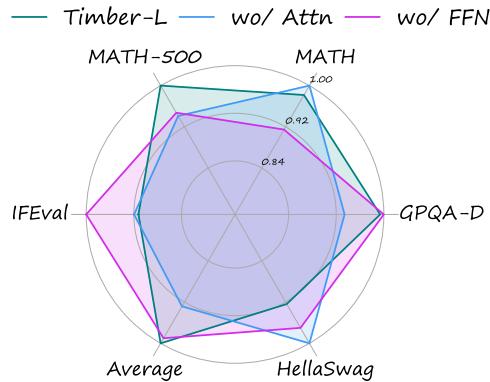
421 Model merging is a training-free paradigm that combines the weights of specialized models to create
 422 a single, more capable one (Yang et al., 2024). Our method, Timber, utilizes a similar principle to
 423 refine the weights of Instruct model by incorporating weights from paired Base model. To bench-
 424 mark our approach, we compare Timber against a straightforward model merging strategy: direct
 425 weighted averaging of the model parameters. Specifically, we merge the weights of layers using a
 426 linear interpolation:

$$\mathbf{W}_{\text{merge}} = \mu \mathbf{W}_I + (1 - \mu) \mathbf{W}_B = \mathbf{W}_B + \mu \mathbf{W}_\Delta, \quad (9)$$

427 where \mathbf{W}_Δ denotes the weight difference and μ is a global scaling factor. From this perspective,
 428 simple model merging is a special case that applies a uniform linear scale to the entire weight delta.
 429 In contrast, Timber employs a more sophisticated, fine-grained paradigm that scales the weight delta
 430 based on its eRank.



445 Figure 6: Performance of Timber and model
446 merge strategies with various μ on Llama-
447 3.2-1B. Vanilla Instruct gets a score of 27.60.



445 Figure 7: Normalized scores of Timber and its ab-
446 lation on the module to apply. *Attn* and *FFN* denote
447 the attention and FFN layers.

448
449 Figure 6 shows the results on Llama-3-1B. The simple merging strategy can slightly improve perfor-
450 mance over the Vanilla Instruct baseline when μ is 0.95. As the scaling factor μ decreases further,
451 performance degrades sharply, quickly falling below the baseline. This highlights the fragility of
452 applying a uniform scaling factor. In contrast, Timber, a fine-grained refinement based on eRank,
453 shows a more robust and significant performance enhancement, demonstrating the superiority of its
454 more nuanced merging strategy.

456 5.3 MODULES TO APPLY TIMBER

457 Prior research indicates that in Transformer, FFN layers primarily store factual and commonsense
458 knowledge, while attention layers are responsible for mixing information between tokens within
459 the context (Geva et al., 2021; Dai et al., 2022; Meng et al., 2023). Given these distinct roles, we
460 conducted an ablation study to isolate which module benefits most from the Timber method.

462 Figure 7 presents the results of applying Timber-L without attention layers (wo/ Attn) or without
463 FFN layers (wo/ FFN) on Llama-3.1-8B. The scores are normalized for better visualization. We
464 can find that applying Timber-L to both modules yields the best overall performance, particularly
465 on [knowledge-intensive tasks](#) like MATH and GPQA-D. Meanwhile, reverting the attention module
466 only (wo/ FFN) performs better at IFEval, while reverting the FFN module only (wo/ Attn) benefits
467 the math reasoning tasks. [This suggests that the complex logical flows required for reasoning rely](#)
468 [heavily on the attention weights, whereas the knowledge-intensive capabilities are predominantly](#)
469 [stored in the FFN layers.](#) Such observation is consistent with the conclusion that FFN modules
470 primarily store factual knowledge (Geva et al., 2021; Dai et al., 2022), while attention modules are
471 responsible for information mixing.

472 5.4 CASE STUDY

474 We further analyze the cases of generated responses. Please refer to Appendix A.6 for detailed ex-
475 amples and analysis on Qwen3-14B. Timber outperforms Instruct model with more comprehensive
476 thinking trajectories. In short, Timber can effectively refine Instruct with Base and thus achieve a
477 better trade-off between exploration and exploitation, which is consistent with the conclusions in
478 Section 4.2.

479 6 CONCLUSION

482 In this work, we first carefully compare the Base and Instruct models in terms of effective rank,
483 reinforcing the hypothesis that post-training is superficial. To tackle the issue that the exploration
484 capability of the Instruct model is limited, we further propose a simple yet effective training-free
485 method, Timber, to refine the weight delta. The key is to partially revert the Instruct model towards
its Base state. Specifically, we first employ eRank as a threshold to split the singular values of weight

486 deltas, followed by an enhancement strategy that either removes or attenuates the tail. Extensive ex-
 487 periments show that Timber successfully enhances exploration without compromising exploitation.
 488 We leave it for future work to explore more strategies to enhance the weight deltas. [Also, applying](#)
 489 [Timber to other model merge scenarios is another interesting topic.](#)
 490

491 REPRODUCIBILITY STATEMENT

494 We are committed to the reproducibility of our work. The full source code required to reproduce
 495 our main findings is included in the *supplementary material*. Corresponding hyperparameters and
 496 detailed configuration files for all experiments are documented in Appendix A.3.2. All experiments
 497 were conducted on publicly available benchmarks, and the details are provided in Appendix A.3.
 498 All the models will be made public in the future.

500 REFERENCES

501 Paul Albert, Frederic Z Zhang, Hemanth Saratchandran, Anton van den Hengel, and Ehsan Abbas-
 502 nejad. Towards higher effective rank in parameter-efficient fine-tuning using khatri-rao product.
 503 In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1292–1302,
 504 2025.

506 Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared
 507 Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large
 508 language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

510 Zhipeng Chen, Xiaobo Qin, Youbin Wu, Yue Ling, Qinghao Ye, Wayne Xin Zhao, and Guang Shi.
 511 Pass@ k training for adaptively balancing exploration and exploitation of large reasoning models.
 512 *arXiv preprint arXiv:2508.10751*, 2025.

513 OpenCompass Contributors. Opencompass: A universal evaluation platform for foundation models.
 514 <https://github.com/open-compass/opencompass>, 2023.

516 Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons
 517 in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for*
 518 *Computational Linguistics (Volume 1: Long Papers)*, pp. 8493–8502, 2022.

520 Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are
 521 key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural*
 522 *Language Processing*, pp. 5484–5495, 2021.

523 Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad
 524 Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd
 525 of models. *arXiv preprint arXiv:2407.21783*, 2024.

527 Xiangming Gu, Tianyu Pang, Chao Du, Qian Liu, Fengzhuo Zhang, Cunxiao Du, Ye Wang, and
 528 Min Lin. When attention sink emerges in language models: An empirical view. In *The Thirteenth*
 529 *International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=78Nn4QJTEN>.

531 Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu
 532 Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforce-
 533 ment learning. *Nature*, 645(8081):633–638, 2025.

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

539 Jie Huang and Kevin Chen-Chuan Chang. Towards reasoning in large language models: A survey.
 540 In *Findings of the association for computational linguistics: ACL 2023*, pp. 1049–1065, 2023.

540 Jiaming Ji, Kaile Wang, Tianyi Alex Qiu, Boyuan Chen, Jiayi Zhou, Changye Li, Hantao Lou, Josef
 541 Dai, Yunhuai Liu, and Yaodong Yang. Language models resist alignment: Evidence from data
 542 compression. In *Proceedings of the 63rd Annual Meeting of the Association for Computational
 543 Linguistics (Volume 1: Long Papers)*, pp. 23411–23432, 2025.

544 Ming Li, Yanhong Li, Ziyue Li, and Tianyi Zhou. How instruction and reasoning data shape post-
 545 training: Data quality through the lens of layer-wise gradients. *arXiv preprint arXiv:2504.10766*,
 546 2025.

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

550 Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. Mass-editing
 551 memory in a transformer. In *The Eleventh International Conference on Learning Representations*,
 552 2023. URL <https://openreview.net/forum?id=MkbcAH1YgyS>.

553 Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke
 554 Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori B Hashimoto. s1: Simple test-time
 555 scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language
 556 Processing*, pp. 20286–20332, 2025.

557 Sagnik Mukherjee, Lifan Yuan, Dilek Hakkani-Tur, and Hao Peng. Reinforcement learning finetunes
 558 small subnetworks in large language models. *arXiv preprint arXiv:2505.11711*, 2025.

559 June Young Park, Minjae Kang, Seongbae Lee, Haegang Lee, Seongwan Kim, and Jaeho Lee. Rie-
 560 mannian optimization for LoRA on the stiefel manifold. In Christos Christodoulopoulos, Tanmoy
 561 Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational
 562 Linguistics: EMNLP 2025*, pp. 20971–20985, Suzhou, China, November 2025. Association for
 563 Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.
 564 1143. URL <https://aclanthology.org/2025.findings-emnlp.1143/>.

565 Wang Qinsi, Jinghan Ke, Masayoshi Tomizuka, Kurt Keutzer, and Chenfeng Xu. Dobi-SVD: Differ-
 566 entiable SVD for LLM compression and some new perspectives. In *The Thirteenth International
 567 Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=kws76i5XB8>.

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

571 Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *2007
 572 15th European signal processing conference*, pp. 606–610. IEEE, 2007.

573 Benjamin Schumacher. Quantum coding. *Physical Review A*, 51(4):2738, 1995.

574 Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang,
 575 Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathemati-
 576 cal reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

577 Evan Z Wang, Federico Cassano, Catherine Wu, Yunfeng Bai, William Song, Vaskar Nath, Ziwen
 578 Han, Sean M. Hendryx, Summer Yue, and Hugh Zhang. Planning in natural language improves
 579 LLM search for code generation. In *The Thirteenth International Conference on Learning Rep-
 580 resentations*, 2025a. URL <https://openreview.net/forum?id=48WAZhwhHw>.

581 Xin Wang, Yu Zheng, Zhongwei Wan, and Mi Zhang. SVD-LLM: Truncation-aware singular value
 582 decomposition for large language model compression. In *The Thirteenth International Confer-
 583 ence on Learning Representations*, 2025b. URL <https://openreview.net/forum?id=LNYIUouhdt>.

584 Lai Wei, Zhiquan Tan, Chenghai Li, Jindong Wang, and Weiran Huang. Diff-erank: A novel rank-
 585 based metric for evaluating large language models. *Advances in Neural Information Processing
 586 Systems*, 37:39501–39521, 2024.

594 Taiqiang Wu, Runming Yang, Jiayi Li, Pengfei Hu, Ngai Wong, and Yujiu Yang. Shadow-ft: Tuning
 595 instruct via base. *arXiv preprint arXiv:2505.12716*, 2025a.
 596

597 Taiqiang Wu, Runming Yang, Tao Liu, Jiahao Wang, and Ngai Wong. Revisiting model interpolation
 598 for efficient reasoning. *arXiv preprint arXiv:2510.10977*, 2025b.
 599

600 Tian Xie, Zitian Gao, Qingnan Ren, Haoming Luo, Yuqian Hong, Bryan Dai, Joey Zhou, Kai Qiu,
 601 Zhirong Wu, and Chong Luo. Logic-rl: Unleashing llm reasoning with rule-based reinforcement
 602 learning. *arXiv preprint arXiv:2502.14768*, 2025.
 603

604 An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu,
 605 Chang Gao, Chengan Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint
 606 arXiv:2505.09388*, 2025.
 607

608 Chenghao Yang and Ari Holtzman. How alignment shrinks the generative horizon. *arXiv preprint
 609 arXiv:2506.17871*, 2025.
 610

611 Enneng Yang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, Jie Zhang, and Dacheng Tao.
 612 Model merging in llms, mllms, and beyond: Methods, theories, applications and opportunities.
 613 *arXiv preprint arXiv:2408.07666*, 2024.
 614

615 Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. Limo: Less is more
 616 for reasoning. *arXiv preprint arXiv:2502.03387*, 2025.
 617

618 Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. Does re-
 619 enforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv
 620 preprint arXiv:2504.13837*, 2025.
 621

622 Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a ma-
 623 chine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association
 624 for Computational Linguistics*, 2019.

625 Yiming Zhang, Baoyi He, Shengyu Zhang, Yuhao Fu, Qi Zhou, Zhijie Sang, Zijin Hong, Kejing
 626 Yang, Wenjun Wang, Jianbo Yuan, et al. Unconstrained model merging for enhanced llm reason-
 627 ing. *arXiv preprint arXiv:2410.13699*, 2024.
 628

629 Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia
 630 Efrat, Ping Yu, Lili Yu, et al. Lima: Less is more for alignment. *Advances in Neural Information
 631 Processing Systems*, 36:55006–55021, 2023a.
 632

633 Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny
 634 Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint
 635 arXiv:2311.07911*, 2023b.
 636

637 Xinyu Zhu, Mengzhou Xia, Zhepei Wei, Wei-Lin Chen, Danqi Chen, and Yu Meng. The surprising
 638 effectiveness of negative reinforcement in llm reasoning. *arXiv preprint arXiv:2506.01347*, 2025.
 639

640

641

642

643

644

645

646

647

648 **A APPENDIX**
649650 **A.1 LLM USAGE**
651652 We utilized a large language model (LLM) to assist in proofreading and refining the language of our
653 manuscript. The use was limited to improving clarity, grammar, and style for both the main text and
654 figure captions. We authors are fully responsible for all scientific claims and the final content of this
655 paper.
656657 **A.2 MORE RELATED WORK**
658659 **Effective Rank.** The effective rank (eRank) is a metric used to quantify the flatness of a singular
660 value distribution, offering insights into the intrinsic dimensionality of a representation. It has found
661 diverse applications in analyzing large language models (LLMs). For instance, Diff-eRank utilizes
662 the eRank of hidden representations to measure how efficiently LLMs prune redundant information
663 during training (Wei et al., 2024). Beyond analysis, eRank has been adapted to measure data quality
664 by examining gradients (Li et al., 2025) and to guide network design, as seen in the KRApapter
665 fine-tuning method (Albert et al., 2025) and the Stiefel optimizer (Park et al., 2025). In our work,
666 we employ eRank to provide further evidence on the nature of post-training adjustments in LLMs.
667668 **Weight-Level Similarity of Base and Instruct Models.** Instruct models are derived from Base
669 models via post-training and thus share an identical architecture. Research has shown that their
670 similarity extends to the weight level, where the differences are often minimal. For example, RL
671 updates only a small subnetwork, leaving most parameters unchanged (Mukherjee et al., 2025).
672 Corroborating this, Wu et al. (2025a) demonstrated that the weight difference between a Base model
673 and its paired Instruct model can be less than 5%. Furthermore, Base and Instruct models have
674 been observed to exhibit similar emergent behaviors, such as similar training dynamics during RL
675 training (Xie et al., 2025) and similar attention sink phenomena (Gu et al., 2025). Building on these
676 findings, our paper, to the best of our knowledge, is the first to analyze the similarity between Base
677 and Instruct models from the perspective of effective rank.
678679 **A.3 EVALUATION DETAILS**
680681 **A.3.1 BENCHMARK**
682683 We conduct evaluation on the wonderful framework OpenCompass (Contributors, 2023). More
684 details about the evaluated benchmark are as follows:
685686

- 687 • IFEval (Zhou et al., 2023b): evaluating instruction-following language models, focusing
688 on their ability to understand and respond to various prompts. It includes 25 types of those
689 verifiable instructions and is constructed around 500 prompts, with each prompt containing
690 one or more verifiable instructions. We report the prompt_level strict accuracy under a
691 zero-shot setting.
- 692 • MATH (Hendrycks et al., 2021): evaluating the mathematical reasoning abilities of AI
693 models through a variety of problem types, including arithmetic, algebra, geometry, and
694 more. There are 7,500 training examples and 5000 test samples. We report the accuracy
695 under a 4-shot setting.
- 696 • MATH-500 (Lightman et al., 2023): 500 uniformly selected test problems from MATH.
697 We report the accuracy under a 4-shot setting.
- 698 • GPQA-Diamond (GPQA-D, (Rein et al., 2024)): evaluating the reasoning ability of large
699 language models (LLMs) on challenging multiple-choice questions written by domain ex-
700 perts in biology, physics, and chemistry. It contains 198 selected questions that require
701 step-by-step reasoning to arrive at the correct answer. We report the accuracy under the
702 0-shot setting.
- 703 • HellaSwag (Zellers et al., 2019): evaluating the ability on commonsense reasoning tasks.
704 It consists of multiple-choice questions where the model must select the most plausible
705 continuation of a given context. We report the accuracy under the 0-shot setting.

702 • AIME’24 ²: evaluating the ability to solve challenging mathematics problems from the
 703 American Invitational Mathematics Examination, a prestigious high school mathematics
 704 competition. We report the accuracy under the 0-shot setting.
 705 • HumanEval (Chen et al., 2021): evaluating the ability of code generation models to write
 706 Python functions based on given specifications. It includes 164 programming problems
 707 with a function signature, docstring, body, and several unit tests.
 708

709 A.3.2 HYPERPARAMETER FOR GENERATION
 710

711 We follow the official recommended hyperparameters for inference. The details as shown in Table
 712 4. We turn on the sampling strategy for more diversity.
 713

714 Table 4: Hyperparameter during generation for different models.
 715

716 Model	717 Temperature	718 Top_p	719 Top_k	720 Max_token
717 Llama-3.2-1B	0.6	0.9	-	4096
718 Llama-3.2-3B	0.6	0.9	-	8192
719 Llama-3.1-8B	0.6	0.9	-	8192
720 Qwen3 Series (Non-Thinking)	0.7	0.8	20	8192
721 Qwen3 Series (Thinking)	0.6	0.95	20	38912

722
 723
 724 A.4 PERFORMANCE WITH DIFFERENT λ
 725

726 Table 7 shows the results of Timber with different λ . We can find that Timber consistently out-
 727 performs the vanilla baseline, demonstrating the robustness. Also, $\lambda = 0.2$ is a sweet point. We
 728 recommend setting λ to 0.2 for the latest released models.
 729

730
 731 A.5 DETAILED RESULTS UNDER THINKING MODE
 732

733 Table 5: Detailed Pass@k performance on AIME24.
 734

736 Model	737 Pass@1	738 Pass@4	739 Pass@8	740 Pass@32	741 Pass@64	742 Pass@128	743 Pass@256	744 Pass@320
Instruct	10.79%	23.14%	31.20%	48.08%	54.72%	60.23%	65.17%	66.67%
Timber-L	11.04%	23.12%	30.75%	49.02%	57.43%	64.00%	68.94%	70.00%
Timber	11.39%	23.78%	31.50%	48.36%	56.55%	63.88%	70.64%	73.33%

745 Table 6: Detailed Pass@k performance on GPQA-Diamond.
 746

744 Model	745 Pass@1	746 Pass@4	747 Pass@8	748 Pass@16	749 Pass@32	750 Pass@64	751 Pass@128	752 Pass@256
Instruct	26.02%	53.60%	67.21%	78.28%	86.25%	91.21%	93.93%	95.45%
Timber-L	25.78%	54.15%	67.83%	78.71%	86.73%	92.18%	95.66%	97.47%
Timber	26.04%	54.38%	68.03%	78.72%	86.17%	91.15%	94.74%	96.97%

753 As shown in Table 5 and 6, for almost all the k values, Timber/Timber-L outperforms the Instruct
 754 model from Pass@1 to Pass@320/256. For AIME24 (Pass@1), Timber improves from 10.79%
 755 (Instruct) to 11.39%. For GPQA (Pass@1), Timber maintains performance (26.04% vs 26.02%).
 756 The performance gap widens significantly as k increases. In conclusion, Timber-L and Timber
 757 demonstrate consistent performance improvements, particularly in preserving exploitation at lower
 758 k values while enhancing exploration at higher k values.
 759

²<https://huggingface.co/datasets/AI-MO/aimo-validation-aime>

756
757

A.6 DETAILED CASES

758
759
760

We showcase specific questions from GPQA-Diamond and corresponding answers from Qwen3-14B. In particular, we employ the Gemini 2.5 pro to simplify the answers for better visualization. The prompt for generation is:

761
762
763
764
765

Answer the following multiple-choice question. The last line of your response should be of the following format: 'ANSWER: LETTER' (without quotes), where LETTER is one of ABCD. Think step by step before answering.

766
767
768

The proposed Timber outperforms vanilla Instruct model regarding the more comprehensive thinking trajectory (shown in Table 8, Table 10) and more extensive domain knowledge (shown in Table 9).

769
770

Table 7: The average of 5 benchmarks for vanilla Instruct and Timber with different λ .

771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809

Model	Vanilla	Timber			
		$\lambda = 0.0$	$\lambda = 0.2$	$\lambda = 0.5$	$\lambda = 0.8$
Llama-3.2-1B	27.60	28.56	28.32	27.91	27.69
Llama-3.2-3B	46.19	46.41	46.23	46.57	46.49
Llama-3.1-8B	53.52	54.89	55.23	53.84	53.16
Qwen3-0.6B	43.57	43.74	43.86	43.92	44.83
Qwen3-8B	74.09	74.39	74.84	74.46	74.12
Qwen3-14B	76.86	77.60	77.66	77.31	77.43
Qwen3-30B-A3B	76.67	77.12	76.82	77.19	77.19

810
811
812
813
814
815
816
817
818

819 Table 8: One case from GPQA-Diamond benchmark. The responses from Timber-L and Timber are
820 more comprehensive via traversal each answer.

821	Question	You have prepared a di-substituted 6-membered aromatic ring compound. The 822 FTIR spectrum of this compound shows absorption peaks indicating the pres- 823 ence of an ester group. The ^1H NMR spectrum shows six signals: two signals 824 corresponding to aromatic-H, two signals corresponding to vinyl-H (one doublet 825 and one doublet of quartets), and two signals corresponding to $-\text{CH}_3$ groups. 826 There are no signals corresponding to $-\text{CH}_2$ groups. Identify the chemical for- 827 mula of this unknown compound as either $\text{C}_{11}\text{H}_{12}\text{O}_2$, $\text{C}_{11}\text{H}_{14}\text{O}_2$, $\text{C}_{12}\text{H}_{12}\text{O}_2$, 828 or $\text{C}_{12}\text{H}_{14}\text{O}_2$. A) $\text{C}_{11}\text{H}_{14}\text{O}_2$ B) $\text{C}_{12}\text{H}_{12}\text{O}_2$ C) $\text{C}_{12}\text{H}_{14}\text{O}_2$ D) $\text{C}_{11}\text{H}_{12}\text{O}_2$
829	Answer	D
830	Instruct	Based on the spectroscopic data, the compound is a di-substituted aromatic ring 831 containing an ester group, a vinyl group, and two methyl groups. The atom count 832 is determined as follows: Carbon: 6 (aromatic ring) + 2 (vinyl) + 2 (two methyls) 833 + 2 (ester) = 12 C Hydrogen: 4 (aromatic) + 2 (vinyl) + 6 (two methyls) = 12 834 H Oxygen: 2 (ester) = 2 O This leads to the chemical formula $\text{C}_{12}\text{H}_{12}\text{O}_2$. ANSWER: B
835	Timber-L	Based on the spectral data, the compound has a di-substituted aromatic ring, an 836 ester group, a vinyl group, and two methyl groups, with no methylene ($-\text{CH}_2$) 837 groups. A plausible structure fitting these clues is a benzene ring substituted with 838 a methyl ester group ($-\text{COOCH}_3$) and a propenyl group ($-\text{CH}=\text{CHCH}_3$). Let's 839 count the atoms for this structure: Carbon (C): 6 (from the ring) + 1 (from $\text{C}=\text{O}$) 840 + 1 (from $-\text{OCH}_3$) + 3 (from the propenyl group) = 11 C Hydrogen (H): 4 (from 841 the ring) + 3 (from $-\text{OCH}_3$) + 5 (from the propenyl group) = 12 H Oxygen (O): 2 842 (from the ester group) = 2 O The resulting chemical formula is $\text{C}_{11}\text{H}_{12}\text{O}_2$. which 843 matches option D. ANSWER: D
844	Timber	Based on the structural information, the degree of unsaturation (DoU) is calculated 845 first. An aromatic ring contributes 4 DoU. An ester group ($\text{C}=\text{O}$) contributes 1 846 DoU. A vinyl group ($\text{C}=\text{C}$) contributes 1 DoU. Total required DoU = $4 + 1 + 1$ 847 = 6. Next, we calculate the DoU for each option: A) $\text{C}_{11}\text{H}_{14}\text{O}_2$: DoU = 5 B) 848 $\text{C}_{12}\text{H}_{12}\text{O}_2$: DoU = 7 C) $\text{C}_{12}\text{H}_{14}\text{O}_2$: DoU = 6 D) $\text{C}_{11}\text{H}_{12}\text{O}_2$: DoU = 6 Only 849 options C and D have the correct DoU of 6. To decide between them, we analyze 850 the ^1H NMR data. A di-substituted benzene ring has 4 H, the vinyl group has 2 851 H, and two methyl groups have 6 H (2×3 H). Total H = 4 (aromatic) + 2 (vinyl) 852 + 6 (methyls) = 12 H. Comparing this to the remaining options, only $\text{C}_{11}\text{H}_{12}\text{O}_2$ 853 has 12 hydrogens. A possible isomer that fits this data is methyl 4-(prop-1-en-1- 854 yl)benzoate. ANSWER: D
855		
856		
857		
858		
859		
860		
861		
862		
863		

864
865
866
867
868
869
870
871
872
873

874 Table 9: One case from GPQA-Diamond benchmark. The vanilla Instruct model lacks the relevant
875 domain knowledge

877	Question	ChIP-seq on a PFA-fixed sample with an antibody to the IKAROS transcription 878 factor in human B cells followed by next-generation sequencing and standard qual- 879 ity control, alignment and peak-calling steps produced ChIP peaks that disappeared 880 when PFA+DSG fixation was used. Where are we most likely to find such dis- 881 appearing peaks? A) At repeats B) In the introns of large genes C) At active promoters 882 and enhancers D) At random locations in the genome
883	Answer	C
884	Instruct	The key observation is that IKAROS ChIP-seq peaks, detected with standard PFA 885 fixation, disappear when a second, more stringent crosslinker, DSG, is added. This 886 suggests the binding captured by PFA alone was not stable or specific. The dis- 887 appearance of these peaks under the more rigorous PFA+DSG condition indicates 888 they were likely weak, transient, or artifactual interactions. Such non-specific bind- 889 ing is not characteristic of functional regulatory elements like active promoters or 890 enhancers but is more likely to occur at random locations across the genome. AN- 891 SWER: D
892	Timber-L	The key observation is that ChIP-seq peaks for the transcription factor IKAROS, 893 found using standard PFA fixation, disappeared when the protocol was altered to 894 use PFA+DSG. This suggests the detected protein-DNA interactions are sensitive 895 to the specific crosslinking chemistry. IKAROS is known to functionally bind at 896 active promoters and enhancers to regulate genes. The conclusion is that these 897 disappearing peaks are located precisely at these functional sites. The dynamic 898 chromatin environment at these active regions may be more susceptible to disrup- 899 tion or alteration by the different crosslinking properties of DSG, leading to the loss 900 of the signal. ANSWER: C
901	Timber	The disappearance of IKAROS ChIP-seq peaks when switching from PFA to the 902 more extensive PFA+DSG fixation method indicates that the detection of this bind- 903 ing is highly sensitive to the crosslinking chemistry. IKAROS is a transcription 904 factor known to function at specific, dynamic regulatory regions. The conclusion is 905 that these disappearing peaks are located at its expected sites of action: active pro- 906 moters and enhancers. These functionally important areas are likely more suscep- 907 tible to having their epitopes masked or their interactions disrupted by the harsher, 908 more extensive crosslinking of PFA+DSG. ANSWER: C
909		
910		
911		
912		
913		
914		
915		
916		
917		

918

919

920

921 Table 10: One case from GPQA-Diamond benchmark. Both Timber-L and Timber reason with
922 mathematical formula while vanilla Insturct model does not.

923

924 Question	An intelligent civilization in the Large Magellanic Cloud has engineered an ex- 925 traordinary spacecraft capable of traveling at a substantial fraction of the speed of 926 light. The average lifetime of these aliens is roughly 150 solar years. Now, having 927 Earth as their destination in mind, they are determined to travel with this spacecraft 928 at a constant speed of $0.99999987 c$, where c is the speed of light. Approximately, 929 how long will it take for their 22 years old astronaut (from the point of view of the 930 astronaut) to reach the Earth using this incredibly fast spacecraft? A) 81 years B) 931 72 years C) The astronaut will die before reaching to the Earth. D) 77 years
932 Answer	A
933 Instruct	According to the theory of special relativity, due to time dilation, time passes more 934 slowly for the traveling astronaut than for an observer on Earth. The core of the 935 problem is to determine if the astronaut survives the trip. Time experienced by as- 936 tronaut: 22 years. Astronaut's initial age: 22 years. Astronaut's age upon arrival: 937 $22 + 22 = 44$ years . Average lifespan of the alien species: 150 years. Since the 938 astronaut's age upon arrival (44) is well within the average lifespan (150), the astro- 939 naut will survive and reach Earth. This conclusion directly contradicts option (C). 940 Furthermore, the question asks for the travel time from the astronaut's perspective, 941 which is given as 22 years, but this is not an option. The time elapsed on Earth is 942 calculated to be over 43,000 years, also not an option. The premises of the question 943 and the provided answer choices are inconsistent. Based on the physical analysis, 944 the statement in option (C) is factually incorrect. ANSWER: C
945 Timber-L	This problem is solved using the principles of time dilation from special relativity. 946 A critical piece of information, the distance of the journey, is missing. 947 The solution assumes the trip is from the Large Magellanic Cloud to Earth, ap- 948 proximately 163,000 light-years . Calculate Earth Time (t): First, we determine 949 the travel time from an Earth observer's perspective. $t = \text{distance}/\text{velocity} \approx$ 950 $163,000 \text{ light-years}/c \approx 163,000 \text{ years}$. Calculate the Time Dilation Factor (γ): 951 Next, we find the Lorentz factor for the spacecraft's velocity ($v = 0.99999987c$). 952 $\gamma = 1/\sqrt{1 - v^2/c^2} \approx 1961$. Calculate Astronaut's Time (t_0): Finally, we find 953 the time experienced by the astronaut by dividing the Earth time by the dilation 954 factor. $t_0 = t/\gamma \approx 163,000 \text{ years}/1961 \approx 83 \text{ years}$. The astronaut's age upon 955 arrival would be $22 + 83 = 105$ years, which is within their 150-year lifespan, so 956 they survive. The calculated journey time of approximately 83 years is closest to 957 option A. ANSWER: A
958 Timber	This problem is solved using special relativity , specifically length contraction . To 959 find the travel time from the astronaut's point of view, we must first calculate how 960 the distance to Earth appears to shrink for them. The solution assumes a travel 961 distance of 160,000 light-years . Calculate the Lorentz Factor (γ) : For a speed 962 of $v = 0.99999987c$, the time dilation and length contraction factor is calculated. 963 $\gamma = 1/\sqrt{1 - v^2/c^2} \approx 1961$ Calculate the Contracted Distance (L') : From the 964 astronaut's perspective, the immense distance is contracted by the Lorentz factor. 965 $L' = \frac{\text{Proper Distance}}{\gamma} \approx \frac{160,000 \text{ light-years}}{1961} \approx 81.6 \text{ light-years}$. Calculate the Astronaut's 966 Travel Time (t') : The time for the astronaut is the contracted distance divided by 967 their velocity (which is nearly the speed of light, c). $t' = \frac{L'}{v} \approx \frac{81.6 \text{ light-years}}{c} \approx$ 968 81.6 years The journey takes about 81.6 years for the astronaut. Since they have 969 128 years of life remaining (150 - 22), they will easily survive. This result is closest 970 to option A. ANSWER: A

971