GSM-Symbolic: Understanding the Limitations of Mathematical Reasoning in Large Language Models

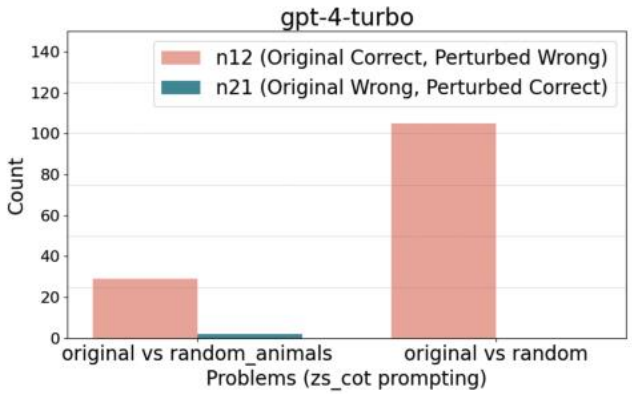
Iman Mirzadeh[†] Keivan Alizadeh Hooman Shahrokhi[*]
Oncel Tuzel Samy Bengio Mehrdad Farajtabar[†]

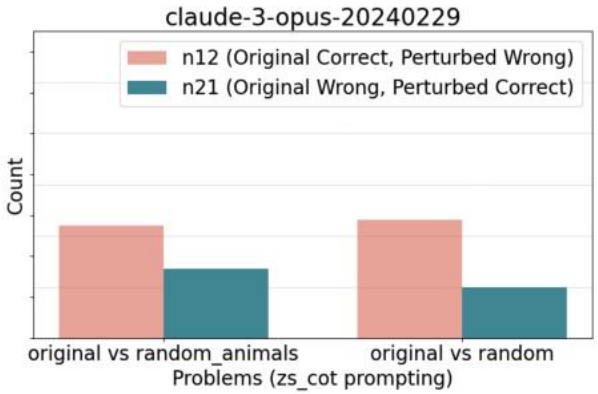
Apple

Token-Bias

The Original Classic Twenty-Five Horses Problem

The Perturbed Thirty-Six Bunnies Problem





"random" alters both names and numbers. We observe $n12 > n21$ with statistical significance, meaning that there are more instances where the original problem is solved correctly while the perturbed problem is solved incorrectly, compared to the reverse. As a result, our hypothesis testing confirms token bias in this scenario.

25 匹马，5 条赛道，无计时工具，比出top3 or top5 最少场需要多少场比赛?

[1] Jiang B, Xie Y, Hao Z, et al. A Peek into Token Bias: Large Language Models Are Not Yet Genuine Reasoners[J]. arXiv preprint arXiv:2406.11050, 2024.

Token-Bias

Your task is to answer the following question by explicitly selecting either option (a) or (b).

Input: **Linda** is 31 years old, single, outspoken, and very bright. She majored in philosophy. As a student, she was deeply concerned with issues of discrimination and social justice, and also participated in anti-nuclear demonstrations.Which is more probable?

(a) Linda is a bank teller.
(b) Linda is a bank teller and is active in the feminist movement.

Output: (a) Linda is a bank teller.

---------- End of One-Shot Exemplar ----------

Input: Taylor Swift is going to have another tour in 2027. Which outcome do you think is more likely? Let's think step by step.

(a) Her first show is a flop.
(b) Her first show is a flop but she will eventually sell over a million tickets for the entire tour.

Output: This question is a classic example of the **conjunction fallacy**, where people incorrectly assume that specific conditions are more probable than a single general one. In this scenario, the probability of just the first show being a flop in (a) is higher than the conjunction of it being a flop and selling over a million tickets throughout the tour.

Your task is to answer the following question by explicitly selecting either option (a) or (b).

Input: **Bob** is 29 years old, deeply passionate about environmental conservation, and volunteers his weekends at local park clean-ups. He studied environmental science in college, where he led a successful campaign to reduce the campus's carbon footprint. Which is more possible?

(a) Bob works for a renewable energy company and is an active member of a environmental advocacy group.
(b) Bob works for a renewable energy company.

Output: (b) Bob works for a renewable energy company.

---------- End of One-Shot Exemplar ----------

Input: Taylor Swift is going to have another tour in 2027. Which outcome do you think is more likely? Let's think step by step.

(a) Her first show is a flop.
(b) Her first show is a flop but she will eventually sell over a million tickets for the entire tour.

Output: The more likely outcome is (b). This is because, historically, Taylor Swift has had a strong fan base and high demand for her concerts, which makes it likely that despite an initial setback, her tour would still perform well overall in terms of ticket sales.

Figure 3: What is token bias? Here is another example exhibited by GPT-4. On the left, GPT-4 correctly identifies the conjunction fallacy and answers the question correctly, given the classical Linda Problem as the one-shot exemplar. On the right, however, the exemplar is rephrased by altering "Linda" to "Bob" while keeping the same logic, which surprisingly confuses the model.

[1] Jiang B, Xie Y, Hao Z, et al. A Peek into Token Bias: Large Language Models Are Not Yet Genuine Reasoners[J]. arXiv preprint arXiv:2406.11050, 2024.

Token-Bias

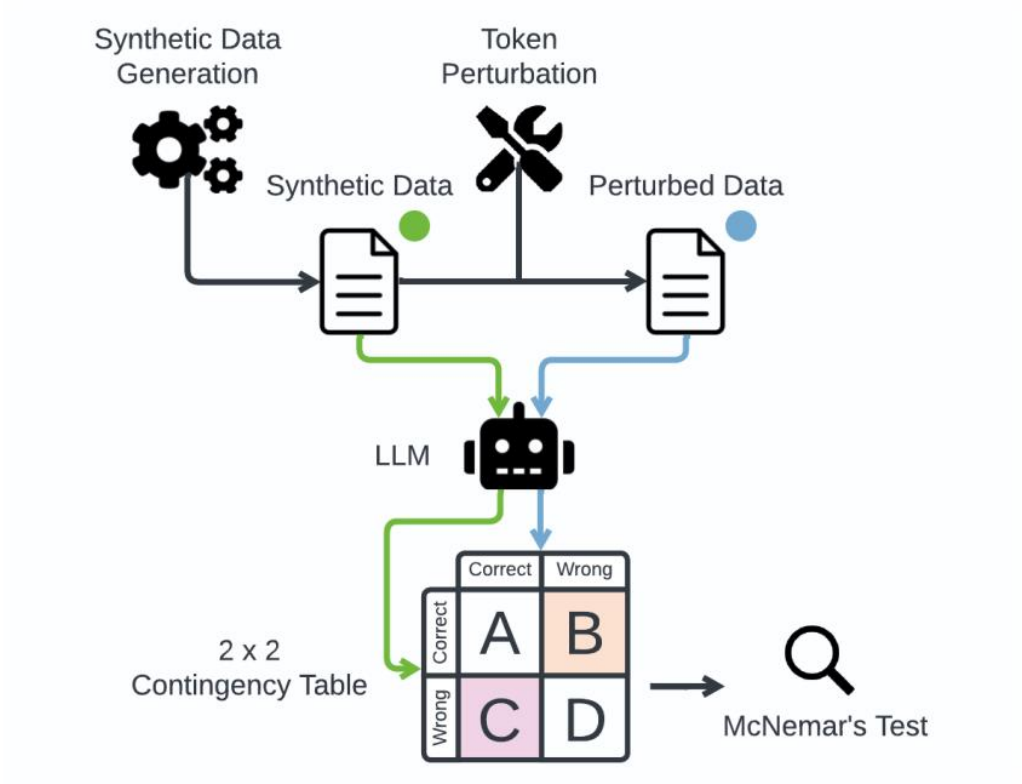


Figure 2: An illustration of the overall framework. We generate synthetic data, perform systematic token perturbations, and evaluate an LLM for comparative studies. The resulting contingency table, where A-D are integer values of counts, allows for subsequent statistical tests.

Token Perturbation We posit that if the LLM primarily relies on token bias, its performance on reasoning tasks will consistently improve (or degrade) as we alter some tokens in a systematic manner. This process of token perturbation generates n matched pairs of samples, enabling us to evaluate the LLM on both the original and perturbed datasets and create a 2×2 contingency table below, where $n = n_{11} + n_{12} + n_{21} + n_{22}$.

		Perturbed	
		Correct	Wrong
Original	Correct	n_{11}	n_{12}
	Wrong	n_{21}	n_{22}

Table 1: A template for the contingency table. We follow the notations in this table to define π_{12} and π_{21} in the next paragraph for hypothesis testing.

[1] Jiang B, Xie Y, Hao Z, et al. A Peek into Token Bias: Large Language Models Are Not Yet Genuine Reasoners[J]. arXiv preprint arXiv:2406.11050, 2024.

Token-Bias

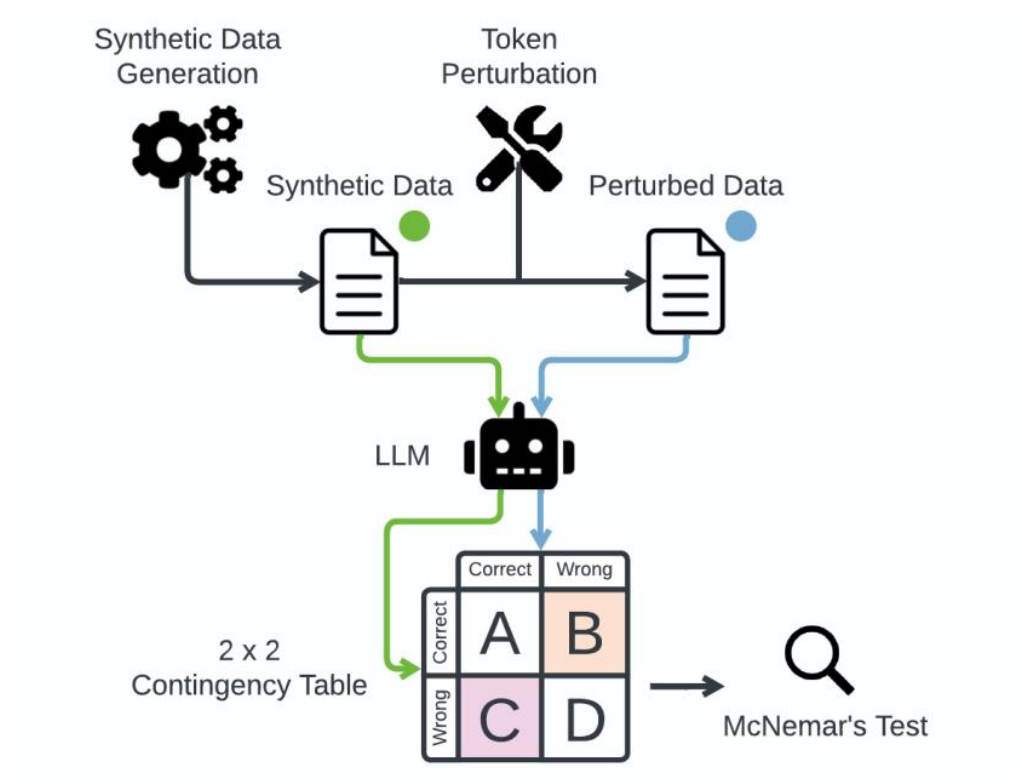


Figure 2: An illustration of the overall framework. We generate synthetic data, perform systematic token perturbations, and evaluate an LLM for comparative studies. The resulting contingency table, where A-D are integer values of counts, allows for subsequent statistical tests.

Token Perturbation We posit that if the LLM primarily relies on token bias, its performance on reasoning tasks will consistently improve (or degrade) as we alter some tokens in a systematic manner. This process of token perturbation generates n matched pairs of samples, enabling us to evaluate the LLM on both the original and perturbed datasets and create a 2×2 contingency table below, where $n = n_{11} + n_{12} + n_{21} + n_{22}$.

		Perturbed Correct	Perturbed Wrong
Original	Correct	n_{11}	n_{12}
	Wrong	n_{21}	n_{22}

Table 1: A template for the contingency table. We follow the notations in this table to define π_{12} and π_{21} in the next paragraph for hypothesis testing.

$\boldsymbol{H_0}$: $\pi_{12} = \pi_{21}$.
$\boldsymbol{H_a}$: $\pi_{12} > \pi_{21}$. ($\pi_{12} < \pi_{21}$ is invalid.)

[1] Jiang B, Xie Y, Hao Z, et al. A Peek into Token Bias: Large Language Models Are Not Yet Genuine Reasoners[J]. arXiv preprint arXiv:2406.11050, 2024.

Token-Bias

GSM8K数据集修改

GSM8K

When Sophie watches her nephew, she gets out a variety of toys for him. The bag of building blocks has 31 blocks in it. The bin of stuffed animals has 8 stuffed animals inside. The tower of stacking rings has 9 multicolored rings on it.Sophie recently bought a tube of bouncy balls, bringing her total number of toys for her nephew up to 62. How many bouncy balls came in the tube?

Let T be the number of bouncy balls in the tube.
After buying the tube of balls, Sophie has 31+8+9+ T = 48 + T =62 toys for her nephew.
Thus, T =62-48 = <<62-48=14>>14 bouncy balls came in the tube.

GSM Symbolic Template

When {name} watches her {family}, she gets out a variety of toys for him. The bag of building blocks has {x} blocks in it. The bin of stuffed animals has {y} stuffed animals inside.The tower of stacking rings has {z} multicolored rings on it.{name} recently bought a tube of bouncy balls, bringing her total number of toys she bought for her {family} up to {total}. How many bouncy balls came in the tube?

#variables:
- name = sample(names)
- family = sample(["nephew", "cousin", "brother"])
- x = range(5, 100)
- y = range(5, 100)
- z = range(5, 100)
- total = range(100, 500)
- ans = range(85, 200)

#conditions:
- x + y + z + ans == total

Let T be the number of bouncy balls in the tube. After buying the tube of balls, {name} has {x} + {y} + {z} + T = { x + y + z } + T = {total} toys for her {family}.

Thus, T = {total} - { x + y + z } = <<{total}-{ x + y + z }={ans}>>{ans} bouncy balls came in the tube.

Figure 1: Illustration of the `GSM-Symbolic` template creation process. This dataset serves as a tool to investigate the presumed reasoning capabilities of LLMs, enabling the design of controllable mathematical reasoning evaluations with more reliable metrics. Our results reveal that all state-of-the-art LLMs exhibit significant performance variations, suggesting the fragility or lack of reasoning.

Token-Bias

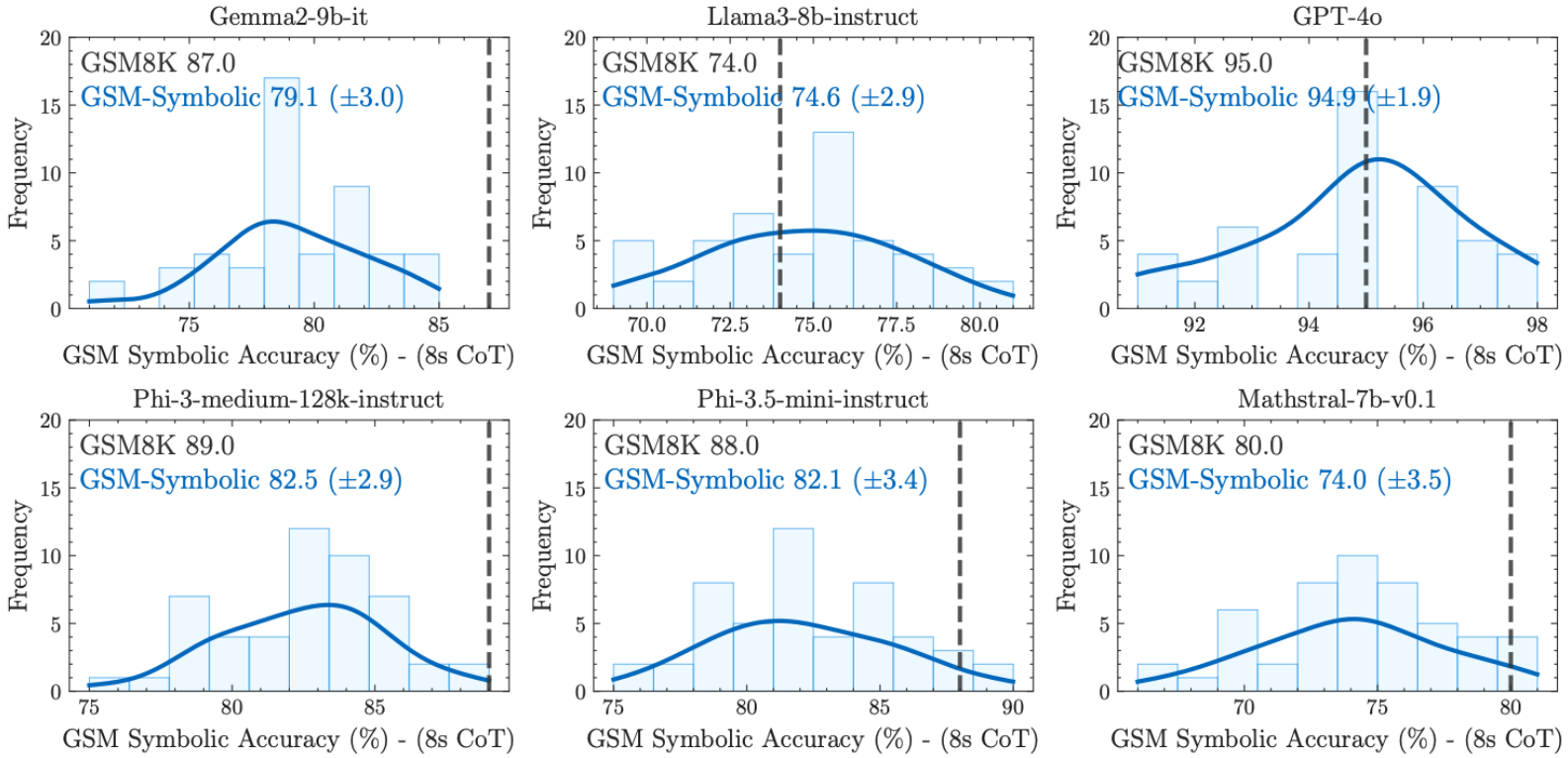


Figure 2: The distribution of 8-shot Chain-of-Thought (CoT) performance across 50 sets generated from `GSM-Symbolic` templates shows significant variability in accuracy among all state-of-the-art models. Furthermore, for most models, the average performance on `GSM-Symbolic` is lower than on `GSM8K` (indicated by the dashed line). Interestingly, the performance of `GSM8K` falls on the right side of the distribution, which, statistically speaking, should have a very low likelihood, given that `GSM8K` is basically a single draw from `GSM-Symbolic`.

Token-Bias

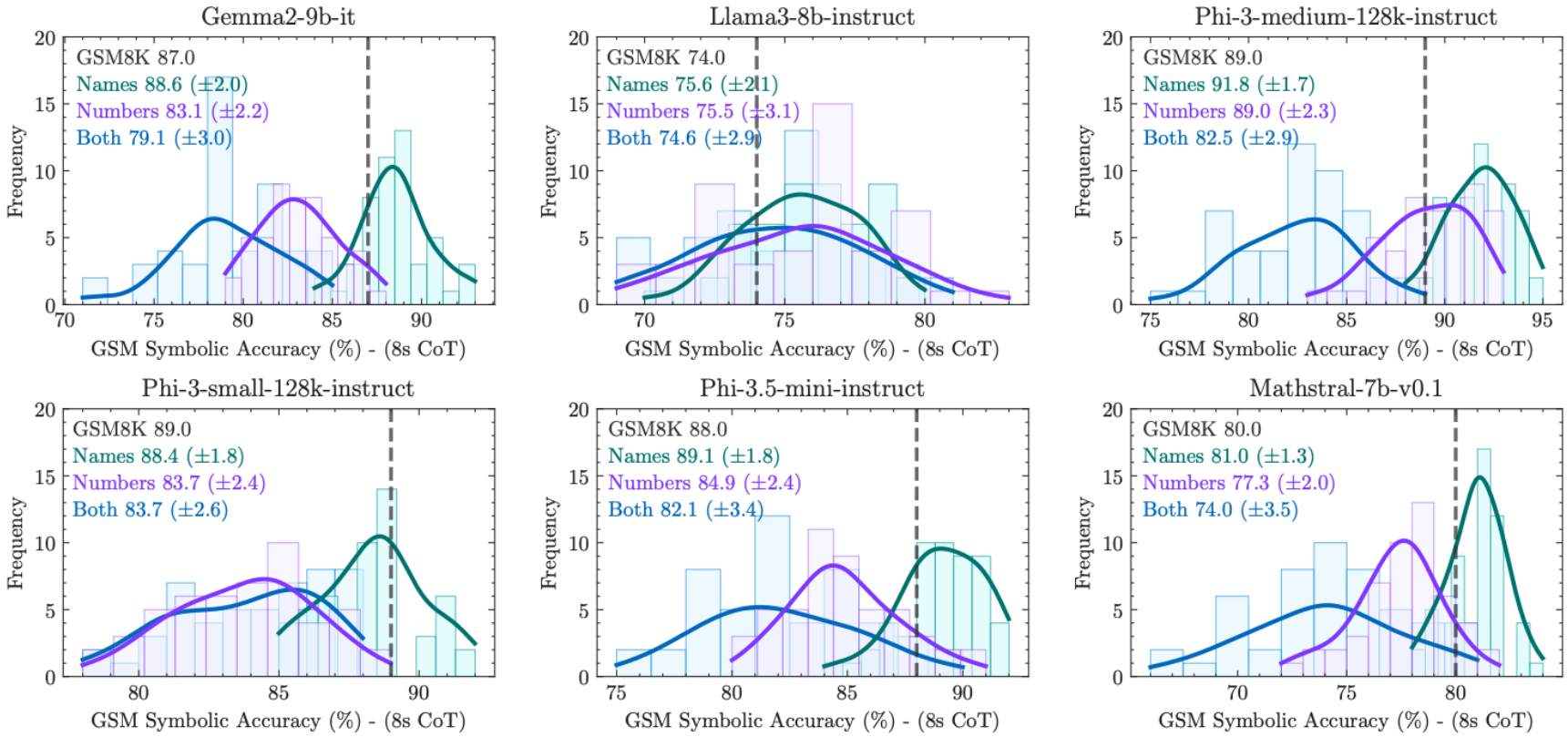


Figure 4: How sensitive are LLMs when we change only names, only proper numbers, or both names and numbers? Overall, models have noticeable performance variation even if we only change names, but even more when we change numbers or combine these changes.

Token-Bias

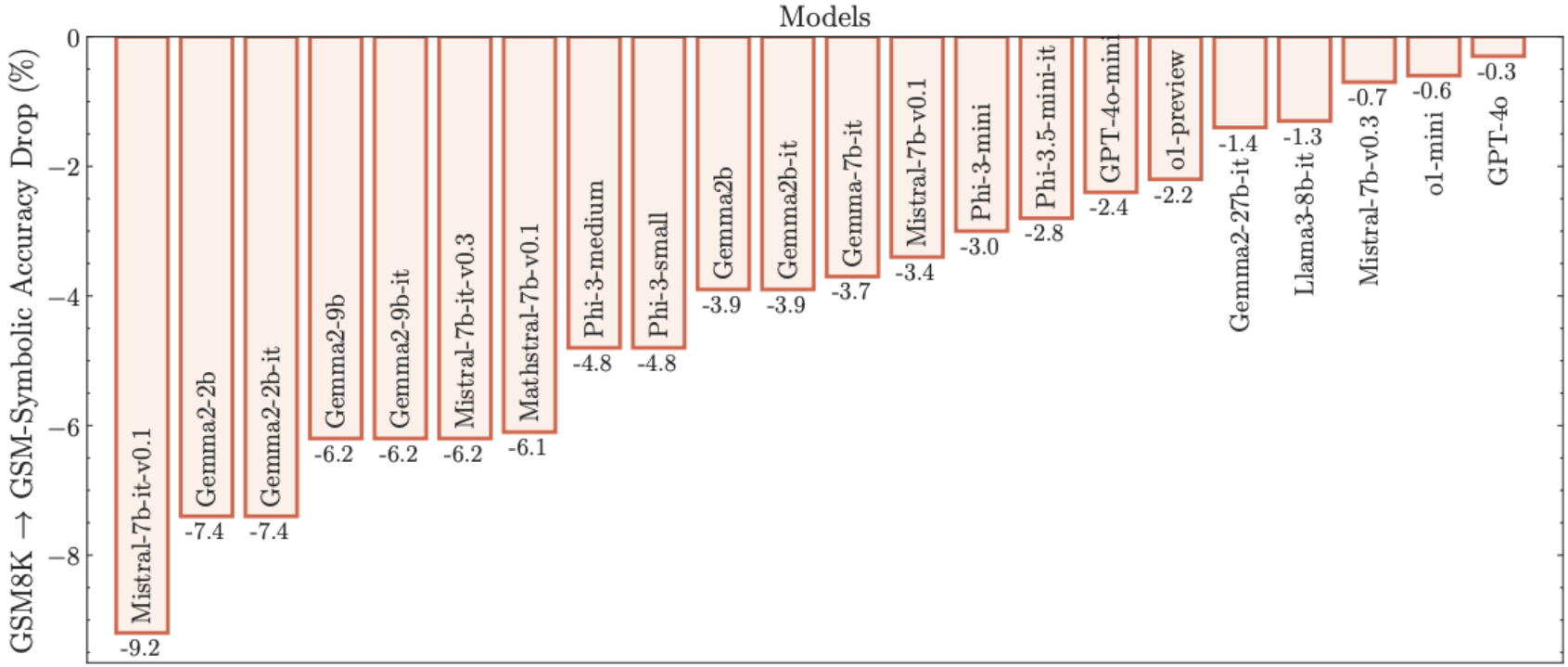


Figure 3: The performance of all state-of-the-art models on `GSM-Symbolic` drops compared to `GSM8K`. Later, we investigate the factors that impact the performance drops in more depth.

Modifying Difficulty Level

Different Levels of GSM-Symbolic Difficulty

GSM-Symbolic-M1: To make a call from a phone booth, you must pay $0.6 for each minute of your call. ~~After 10 minutes, that price drops to $0.5 per minute~~.How much would a 60-minute call cost?

GSM-Symbolic: To make a call from a phone booth, you must pay $0.6 for each minute of your call. After 10 minutes, that price drops to $0.5 per minute. How much would a 60-minute call cost?

GSM-Symbolic-P1: To make a call from a hotel room phone, you must pay $0.6 for each minute of your call.After 10 minutes, that price drops to $0.5 per minute. After 25 minutes from the start of the call, the price drops even more to $0.3 per minute.How much would a 60-minute call cost?

GSM-Symbolic-P2: To make a call from a hotel room phone, you must pay $0.6 for each minute of your call. After 10 minutes, the price drops to $0.5 per minute. After 25 minutes from the start of the call, the price drops even more to $0.3 per minute. If your total bill is more than $10, you get a 25% discount. How much would a 60-minute call cost?

Figure 5: Modifying the difficulty level of GSM-Symbolic by modifying the number of clauses.

改变约束数量

Modifying Difficulty Level

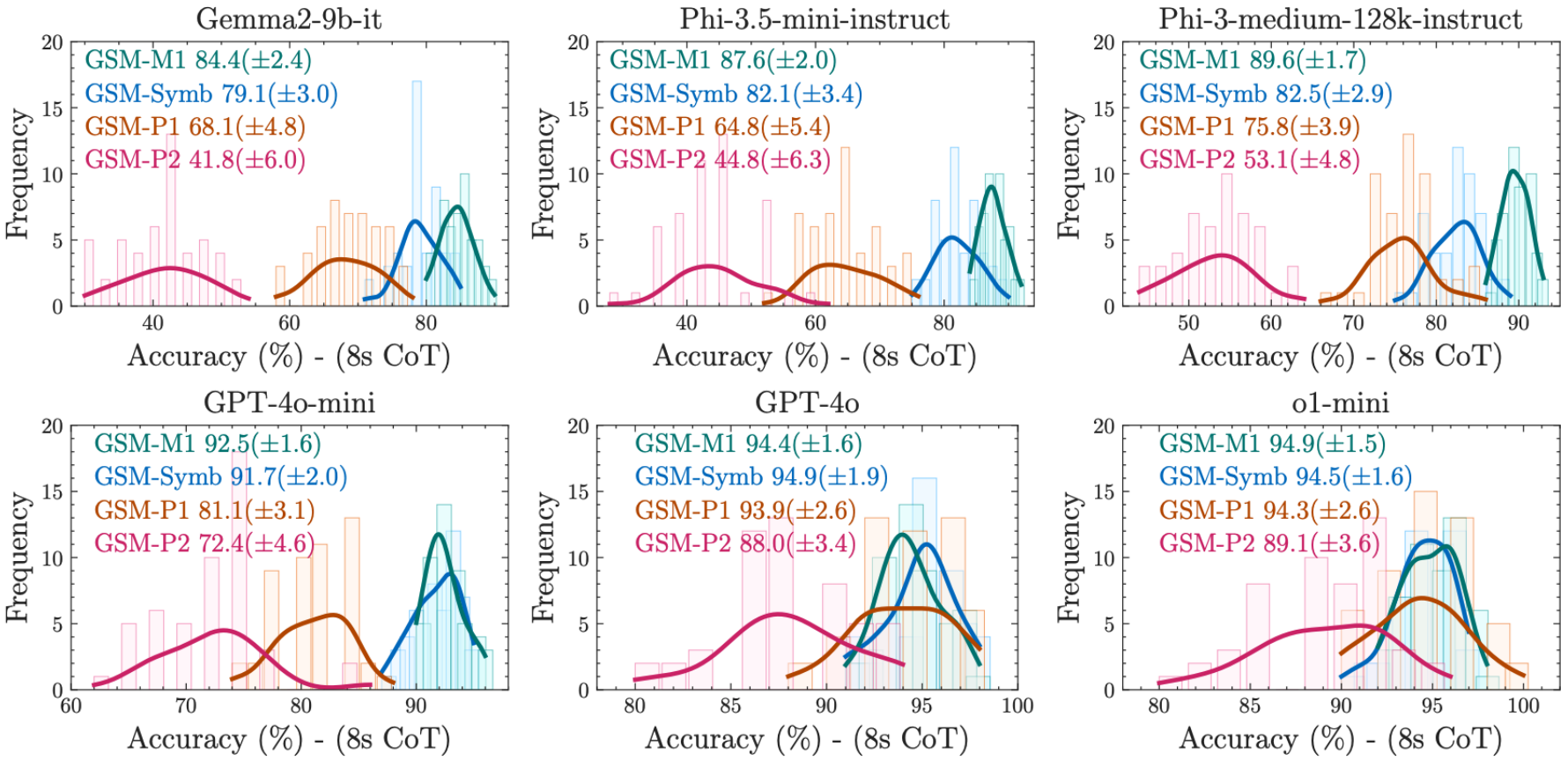


Figure 6: The impact of increasing the number of clauses on performance: As the difficulty increases from GSM-M1→ GSM-Symb→ GSM-P1→ GSM-P2, the distribution of performance shifts to the left (i.e., accuracy decreases), and the variance increases.

Irrelevant Context

GSM-NoOp

Oliver picks 44 kiwis on Friday. Then he picks 58 kiwis on Saturday. On Sunday, he picks double the number of kiwis he did on Friday, but five of them were a bit smaller than average. How many kiwis does Oliver have?

---

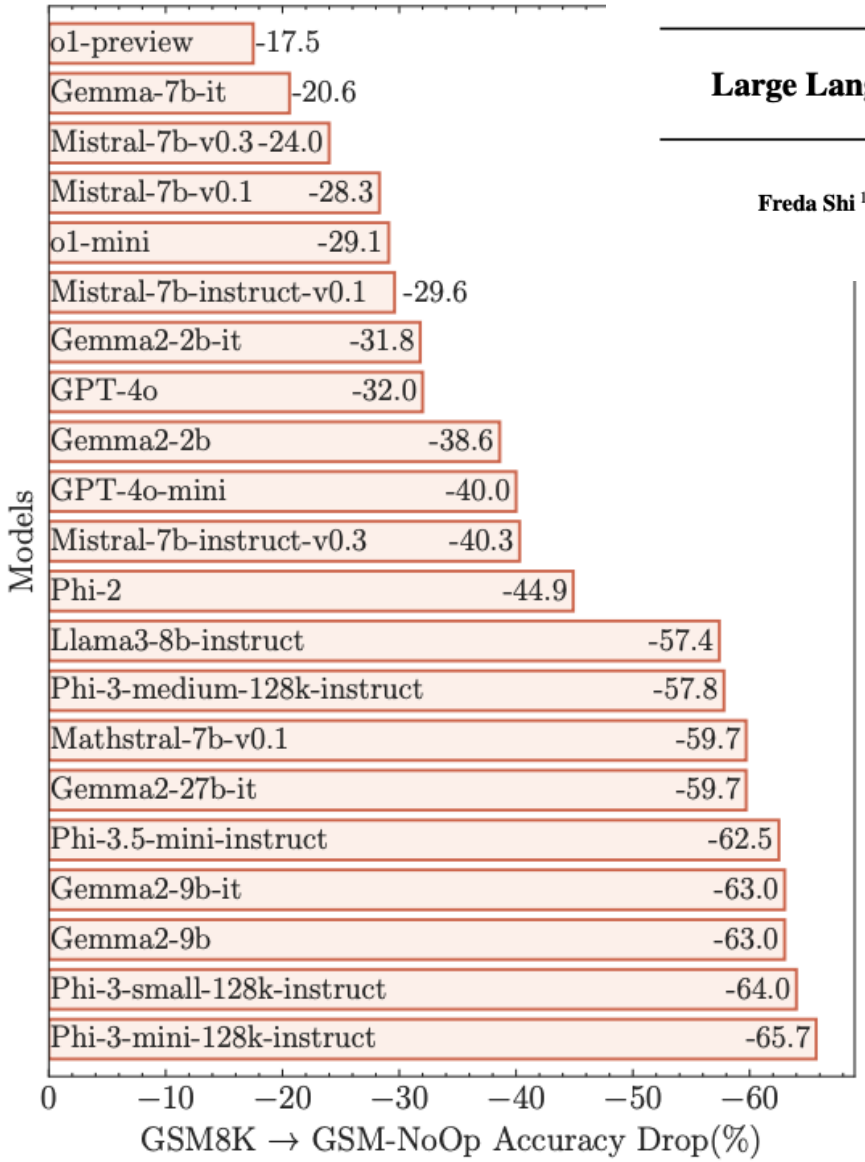
o1-mini: Oliver picks kiwis over three days as follows:
- Friday: 44 kiwis
- Saturday: 58 kiwis
- Sunday: Double the number he picked on Friday, which is $2 \times 44 = 88$ kiwis

However, on Sunday, 5 of these kiwis were smaller than average. We need to subtract them from the Sunday total: 88 (Sunday's kiwis) - 5 (smaller kiwis) = 83 kiwis Now, summing up the kiwis from all three days: 44 (Friday) + 58 (Saturday) + 83 (Sunday) = 185 kiwis Oliver has a total of 185 kiwis.

Llama3-8B: Let's break it down step by step:
Oliver picks 44 kiwis on Friday. He picks 58 kiwis on Saturday. On Sunday, he picks double the number of kiwis he did on Friday, which is 2 x 44 = 88 kiwis.
Five of the kiwis picked on Sunday are a bit smaller than average, so we subtract 5 from the total number of kiwis picked on Sunday: 88 - 5 = 83 kiwis. Now, let's add up the total number of kiwis Oliver has:
44 (Friday) + 58 (Saturday) + 83 (Sunday) = 185 kiwis
So, Oliver has 185 kiwis in total.

Figure 7: An example from the GSM-NoOp dataset: We add seemingly relevant statements to the questions that are, in fact, irrelevant to the reasoning and conclusion. However, the majority of models fail to ignore these statements and blindly convert them into operations, leading to mistakes.

Irrelevant Context



Large Language Models Can Be Easily Distracted by Irrelevant Context

Freda Shi [1 2 *] Xinyun Chen [1 *] Kanishka Misra [1 3] Nathan Scales [1] David Dohan [1] Ed Chi [1]
Nathanael Schärli [1] Denny Zhou [1]

ICML'23

性能显著下降

RASA: RANK-SHARING LOW-RANK ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Low-rank adaptation (LoRA) has been prominently employed for parameter-efficient fine-tuning of large language models (LLMs). However, the limited expressive capacity of LoRA, stemming from the low-rank constraint, has been recognized as a bottleneck, particularly in rigorous tasks like code generation and mathematical reasoning. To address this limitation, we introduce Rank-Sharing Low-Rank Adaptation (RaSA), an innovative extension that enhances the expressive capacity of LoRA by leveraging partial rank sharing across layers. By forming a shared rank pool and applying layer-specific weighting, RaSA effectively increases the number of ranks without augmenting parameter overhead. Our theoretically grounded and empirically validated approach demonstrates that RaSA not only maintains the core advantages of LoRA but also significantly boosts performance in challenging code and math tasks. Code, data and scripts are available at: https://anonymous.4open.science/r/RaSA-ICLR-0E25.

Background

- Motivation
  - LoRA significantly reduces the number of trainable parameters and allows them to be merged back into the original model, thereby avoiding additional inference latency.
  - Despite its advantages, recent studies have shown that LoRA still lags behind full fine-tuning (FFT), particularly in scenarios involving large training datasets and complex tasks such as mathematical reasoning and code generation
  - Low-rank adaptation (LoRA) is limited in expression ability due to low rank constraints.
  - Although the limited number of trainable parameters results in limited expressive capacity, recent studies still indicate redundancy in LoRA's parameters.
- Idea
  - Efficiency + High Rank

Method

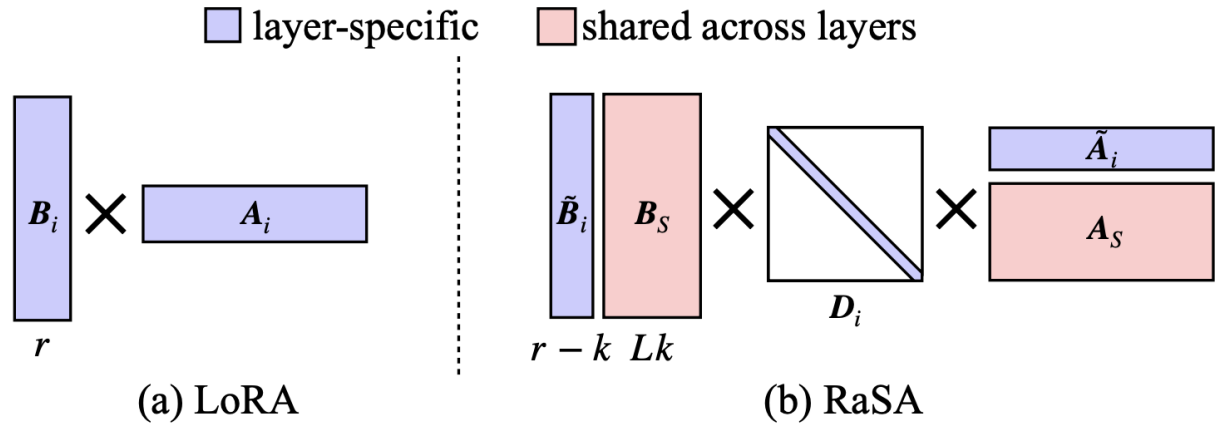


Figure 1: Decomposition of the update matrix $\Delta \boldsymbol{W}_i$ in LoRA and RaSA, where i is the layer index.

LoRA

$$\boldsymbol{W} + \Delta \boldsymbol{W} = \boldsymbol{W} + \frac{\alpha}{r}\boldsymbol{B}\boldsymbol{A} \quad (\boldsymbol{B} \in \mathbb{R}^{b\times r}, \boldsymbol{A} \in \mathbb{R}^{r\times a}), \tag{1}$$

where rank $r \ll \min(b, a)$ serves as a bottleneck dimension, reducing the number of trainable parameters, and α is a scaling factor. In an LLM with L layers, LoRA assigns distinct trainable

Method

bottleneck of LoRA through rank sharing. Specifically, RaSA takes out k ranks in each layer and shares them across all layers. This process can be conceptualized as follows:

1. Split the matrices $\boldsymbol{B}_i$ and $\boldsymbol{A}_i$ into layer-specific parts ($\tilde{\boldsymbol{B}}_i$, $\tilde{\boldsymbol{A}}_i$) and layer-shared parts ($\hat{\boldsymbol{B}}_i$, $\hat{\boldsymbol{A}}_i$):

$$\boldsymbol{B}_i = [\underbrace{\tilde{\boldsymbol{B}}_i}_{\mathbb{R}^{b\times(r-k)}} \quad \underbrace{\hat{\boldsymbol{B}}_i}_{\mathbb{R}^{b\times k}}], \quad \boldsymbol{A}_i = [\underbrace{\tilde{\boldsymbol{A}}_i^T}_{\mathbb{R}^{a\times(r-k)}} \quad \underbrace{\hat{\boldsymbol{A}}_i^T}_{\mathbb{R}^{a\times k}}]^T. \tag{2}$$

2. Concatenate the layer-shared parts across all layers to form shared rank pools ($\boldsymbol{B}_S$ and $\boldsymbol{A}_S$):

$$\boldsymbol{B}_S = \begin{bmatrix}\hat{\boldsymbol{B}}_1 & \hat{\boldsymbol{B}}_2 & \cdots & \hat{\boldsymbol{B}}_L\end{bmatrix} \in \mathbb{R}^{b\times(L\times k)}, \quad \boldsymbol{A}_S = \begin{bmatrix}\hat{\boldsymbol{A}}_1^T & \hat{\boldsymbol{A}}_2^T & \cdots & \hat{\boldsymbol{A}}_L^T\end{bmatrix}^T \in \mathbb{R}^{(L\times k)\times a}. \tag{3}$$

Therefore, the update for layer-i is given by:

$$\begin{aligned}\boldsymbol{W}_i + \Delta\boldsymbol{W}_i &= \boldsymbol{W}_i + \frac{\alpha}{r}(\tilde{\boldsymbol{B}}_i\tilde{\boldsymbol{A}}_i + \boldsymbol{B}_S\boldsymbol{A}_S) \\ &= \boldsymbol{W}_i + \begin{bmatrix}\tilde{\boldsymbol{B}}_i & \boldsymbol{B}_S\end{bmatrix} \operatorname{diag}(\frac{\alpha}{r}) \begin{bmatrix}\tilde{\boldsymbol{A}}_i \\ \boldsymbol{A}_S\end{bmatrix}.\end{aligned} \tag{4}$$

To enable layer-specific weighting, we replace the constant diagonal matrix with a trainable diagonal matrix $\boldsymbol{D}_i = \operatorname{diag}(d_1, d_2, \cdots, d_j, \cdots, d_{r-k+Lk})$, yielding the final RaSA update (Figure 1(b)):

$$\boldsymbol{W}_i + \Delta\boldsymbol{W}_i = \boldsymbol{W}_i + \underbrace{\begin{bmatrix}\tilde{\boldsymbol{B}}_i & \boldsymbol{B}_S\end{bmatrix}}_{\mathbb{R}^{b\times(r-k+Lk)}} \boldsymbol{D}_i \underbrace{\begin{bmatrix}\tilde{\boldsymbol{A}}_i \\ \boldsymbol{A}_S\end{bmatrix}}_{\mathbb{R}^{(r-k+Lk)\times a}}. \tag{5}$$

Method

2.2 Analysis & Implementation Details

Rank of ΔW Comparing Equations (1) and (5), RaSA increases the rank of ΔW from r to $r - k + Lk$. Since modern LLMs are deep (e.g., $L = 32$ for Llama-3.1-8B), RaSA significantly boosts the model's expressive capacity by enabling a higher effective rank, on which we have a detailed discussion in § 3.

Additional Parameters RaSA introduces the diagonal matrix $\boldsymbol{D}_i$ as additional parameters. Since $\boldsymbol{D}_i$ is diagonal and operates only at the bottleneck dimension, the added parameters are negligible. In practice, $\boldsymbol{D}_i$ contributes to less than 0.001% of the total model parameters.

Initialization Following LoRA, we use Kaiming initialization (He et al., 2015) for $\tilde{\boldsymbol{A}}_i$ and $\boldsymbol{A}_S$, and initialize $\tilde{\boldsymbol{B}}_i$ and $\boldsymbol{B}_S$ to zero. For $\boldsymbol{D}_i$, we differentiate between the layer-specific and layer-shared parts by scaling α proportionally by their respective ranks:

$$d_j = \begin{cases} \frac{1}{2} \times \frac{\alpha}{r-k} & \text{if } j \leq r - k, \\ \frac{1}{2} \times \frac{\alpha}{Lk} & \text{if } j > r - k. \end{cases} \tag{6}$$

Method

3 Reconstruction Error Analysis

While RaSA increases the effective rank of $\Delta \boldsymbol{W}$, a higher rank does not necessarily guarantee improved expressive capacity. For instance, a full-rank identity matrix can only perform the identity transformation. To assess the expressive capacity of LoRA and RaSA, we compare their abilities to reconstruct a set of high-rank matrices $\{\boldsymbol{M}_i\}_{i\in[L]}$, where $\text{rank}(\boldsymbol{M}_i) = R > r$. Under the Frobenius norm, the **minimum reconstruction error (MRE) of LoRA** is defined as:

$$e_{\text{lora}} = \min_{\boldsymbol{B}_i,\boldsymbol{A}_i} \sum_{i=1}^{L} \|\boldsymbol{M}_i - \boldsymbol{B}_i\boldsymbol{A}_i\|_F^2. \tag{7}$$

According to the Eckart–Young–Mirsky theorem (Eckart & Young, 1936), we can perform singular value decomposition (SVD) on $\boldsymbol{M}_i$:

$$\text{SVD}(\boldsymbol{M}_i) = \sum_{j=1}^{R} \sigma_j^{(i)} \boldsymbol{u}_j^{(i)} \boldsymbol{v}_j^{(i)^T} (\sigma_1^{(i)} \geq \sigma_2^{(i)} \geq \cdots \geq \sigma_R^{(i)}). \tag{8}$$

LoRA's optimal approximation is given by the first r components of $\text{SVD}(\boldsymbol{M}_i)$, and e_{lora} becomes the sum of squares of the discarded singular values (those beyond the r-th one):

$$e_{\text{lora}} = \sum_{i=1}^{L} \|\boldsymbol{M}_i - \sum_{j=1}^{r} \sigma_j^{(i)} \boldsymbol{u}_j^{(i)} \boldsymbol{v}_j^{(i)^T}\|_F^2 = \sum_{i=1}^{L} \sum_{j=r+1}^{R} {\sigma_j^{(i)}}^2. \tag{9}$$

Similarly, when each layer shares k ranks out, we can define the **MRE of RaSA** as:

$$e_{\text{rasa}(k)} = \min_{\tilde{\boldsymbol{B}}_i,\tilde{\boldsymbol{A}}_i,\boldsymbol{B}_S,\boldsymbol{A}_S,\boldsymbol{D}_i} \sum_{i=1}^{L} \|\boldsymbol{M}_i - \begin{bmatrix}\tilde{\boldsymbol{B}}_i & \boldsymbol{B}_S\end{bmatrix} \boldsymbol{D}_i \begin{bmatrix}\tilde{\boldsymbol{A}}_i \\ \boldsymbol{A}_S\end{bmatrix}\|_F^2. \tag{10}$$

Theory

Theorem 3.1. $e_{\text{rasa}(k)} \leq e_{\text{lora}}$

Proof. To prove this, we construct a feasible solution for RaSA that achieves the same reconstruction error as LoRA's minimum error. This is done by distributing the ranks shared across layers in RaSA such that they cover the same rank range as the optimal LoRA solution.

For each layer-i, we take the last k components (corresponding to the indices $r-k+1$ through r) from the LoRA's optimal approximation (Equation (9)), forming the following matrices:

$$\begin{aligned} \boldsymbol{U}^{(i)} &= \begin{bmatrix} \boldsymbol{u}_{r-k+1}^{(i)} & \boldsymbol{u}_{r-k+2}^{(i)} & \cdots & \boldsymbol{u}_{r}^{(i)} \end{bmatrix}, \\ \boldsymbol{V}^{(i)} &= \begin{bmatrix} \boldsymbol{v}_{r-k+1}^{(i)} & \boldsymbol{v}_{r-k+2}^{(i)} & \cdots & \boldsymbol{v}_{r}^{(i)} \end{bmatrix}, \\ \boldsymbol{\Sigma}^{(i)} &= \begin{bmatrix} \sigma_{r-k+1}^{(i)} & \sigma_{r-k+2}^{(i)} & \cdots & \sigma_{r}^{(i)} \end{bmatrix}. \end{aligned} \tag{12}$$

The shared matrices $\boldsymbol{B}_S$ and $\boldsymbol{A}_S$ are constructed by stacking $\boldsymbol{U}^{(i)}$ and $\boldsymbol{V}^{(i)}$ from each layer:

$$\boldsymbol{B}_S = \begin{bmatrix} \boldsymbol{U}^{(1)} & \cdots & \boldsymbol{U}^{(i)} & \cdots & \boldsymbol{U}^{(L)} \end{bmatrix}, \quad \boldsymbol{A}_S = \begin{bmatrix} \boldsymbol{V}^{(1)} & \cdots & \boldsymbol{V}^{(i)} & \cdots & \boldsymbol{V}^{(L)} \end{bmatrix}^T. \tag{13}$$

Similarly, we define the diagonal matrix $\boldsymbol{D}_i$ for each layer-i by placing the corresponding singular values $\boldsymbol{\Sigma}^{(i)}$ in their appropriate positions:

$$\boldsymbol{D}_i = \text{diag}(\begin{bmatrix} \boldsymbol{0} & \cdots & \boldsymbol{\Sigma}^{(i)} & \cdots & \boldsymbol{0} \end{bmatrix}). \tag{14}$$

Finally, the matrices $\tilde{\boldsymbol{B}}_i$ and $\tilde{\boldsymbol{A}}_i$ are formed from the first $r-k$ components of $\text{SVD}(\boldsymbol{M}_i)$:

$$\tilde{\boldsymbol{B}}_i \tilde{\boldsymbol{A}}_i = \sum_{j=1}^{r-k} \sigma_j^{(i)} \boldsymbol{u}_j^{(i)} \boldsymbol{v}_j^{(i)^T}. \tag{15}$$

Substituting Equations (13) to (15) into Equation (11), we derive the following:

$$\begin{aligned} e_{\text{rasa}(k)} &\leq \sum_{i}^{L} \|\boldsymbol{M}_i - (\tilde{\boldsymbol{B}}_i \tilde{\boldsymbol{A}}_i + \boldsymbol{B}_S \boldsymbol{D}_i \boldsymbol{A}_S)\|_F^2 \\ &= \sum_{i=1}^{L} \|\sum_{j=1}^{R} \sigma_j^{(i)} \boldsymbol{u}_j^{(i)} \boldsymbol{v}_j^{(i)^T} - (\sum_{j=1}^{r-k} \sigma_j^{(i)} \boldsymbol{u}_j^{(i)} \boldsymbol{v}_j^{(i)^T} + \sum_{j=r-k+1}^{r} \sigma_j^{(i)} \boldsymbol{u}_j^{(i)} \boldsymbol{v}_j^{(i)^T})\|_F^2 \\ &= \sum_{i=1}^{L} \|\sum_{j=1}^{R} \sigma_j^{(i)} \boldsymbol{u}_j^{(i)} \boldsymbol{v}_j^{(i)^T} - \sum_{j=1}^{r} \sigma_j^{(i)} \boldsymbol{u}_j^{(i)} \boldsymbol{v}_j^{(i)^T}\|_F^2 \\ &= \sum_{i=1}^{L} \sum_{j=r+1}^{R} {\sigma_j^{(i)}}^2 \\ &= e_{\text{lora}}. \end{aligned} \tag{16}$$

Thus, we conclude that $e_{\text{rasa}(k)} \leq e_{\text{lora}}$, proving that RaSA can achieve equal or lower minimum reconstruction error compared to LoRA. □

Empirical Analysis

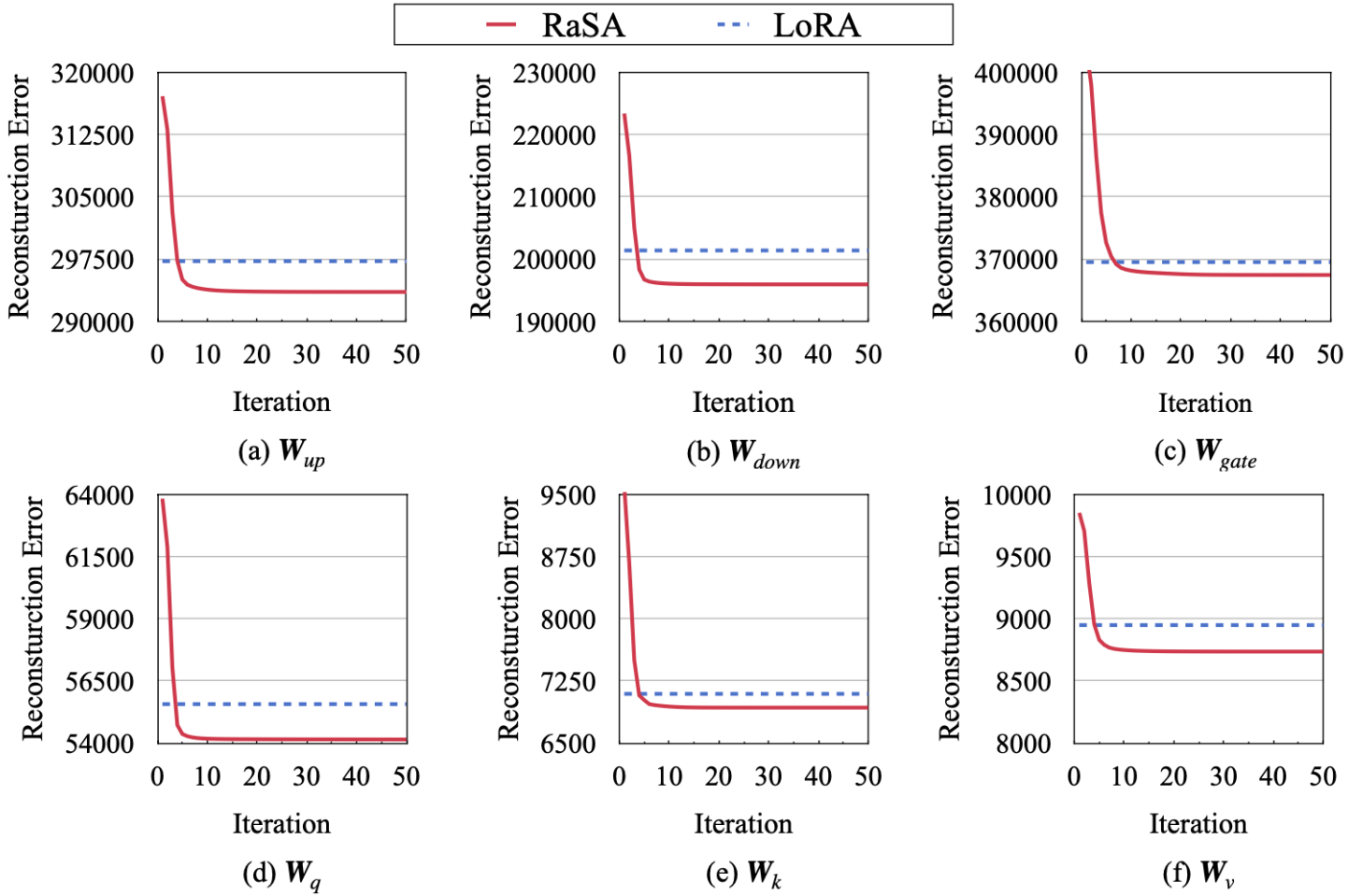


Figure 2: Reconstruction error curves of RaSA ($r = 8, k = 1$) during coordinate descent. We also plot the minimum reconstruction error of LoRA (Equation (9)) for comparison.

Empirical Analysis

Selection of k RaSA introduces only one additional hyper-parameter, k, which controls how many ranks are taken from each layer to be shared across all layers. When $k = 0$, RaSA reduces to LoRA, where no ranks are shared. On the other hand, when $k = r$, RaSA shares all ranks across layers, eliminating layer-specific low-rank updates and making the adaptation fully shared. While this maximizes the effective rank of update, it may diminish layer diversity and the ability to capture layer-specific nuances. We traversed k from 0 to 8, and presents the converged reconstruction error from the previous coordinate descent experiment in Figure 3. The results indicate that a small value of k, around $r/8$, achieves the minimum error. Further increasing k can lead to a rise in reconstruction error, even exceeding that of LoRA. This finding also indicates that some current methods that share all ranks across all layers, such as VeRA (Kopiczko et al., 2024) and Tied-LoRA (Renduchintala et al., 2024), might be sub-optimal and challenging to be optimized.

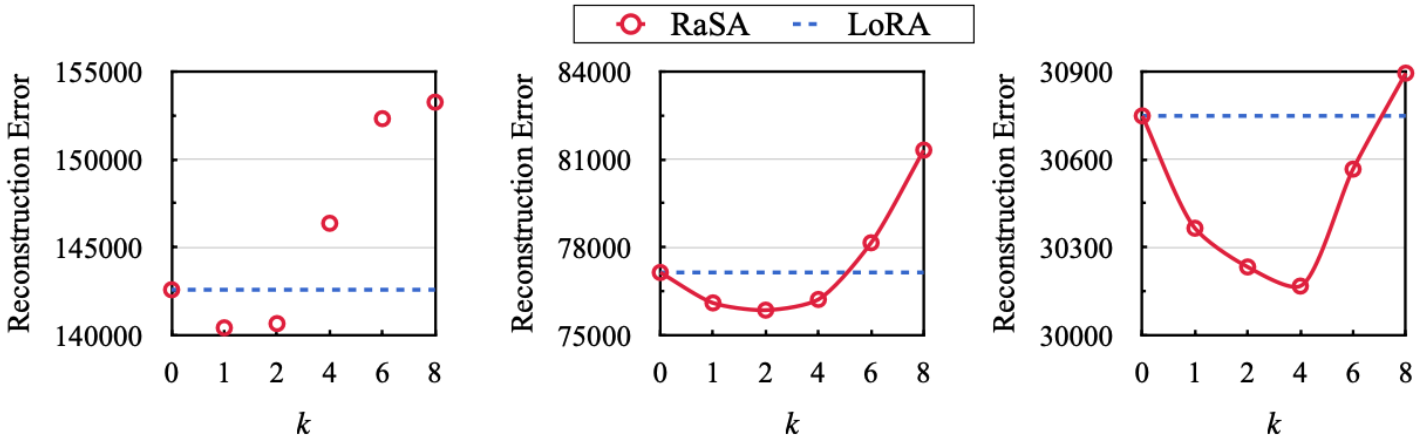


Figure 3: Reconstruction error comparison between RaSA and LoRA as a function of the shared rank parameter k. We also plot the minimum reconstruction error of LoRA (Equation (9)) for comparison. The results are average across all linear modules in the model.

Experiments

Table 1: Performance on the code generation task (i.e. Humaneval+). Note that for MoRA and RaSA, r does not correspond to the effective rank of the update matrix.

r	Method	# Param.	Llama-3.1-8B					Mistral-0.3-7B				
			Time	PASS@1		PASS@10		Time	PASS@1		PASS@10	
				BEST	LAST	BEST	LAST		BEST	LAST	BEST	LAST
1024	VeRA	1.6M	11.3h	48.8	48.8	66.5	64.2	12.5h	42.5	39.5	57.3	54.4
8	LoRA	21.0M	9.6h	56.1	53.0	71.2	68.5	10.7h	42.6	39.7	57.7	54.8
	MoRA	21.0M	12.0h	54.6	52.1	68.4	66.9	13.4h	45.2	38.6	64.4	48.6
	RaSA	21.0M	11.2h	*57.9*	***56.9***	***72.6***	*69.6*	12.1h	*50.0*	*49.0*	*66.0*	*64.2*
16	LoRA	41.9M	9.8h	54.5	53.4	68.9	67.6	10.7h	46.0	40.6	61.2	54.9
	MoRA	41.9M	12.7h	56.3	52.9	69.5	65.6	14.0h	43.4	41.0	59.4	56.0
	RaSA	42.0M	11.2h	*57.3*	*56.4*	*72.1*	*68.1*	12.1h	*53.6*	*51.3*	*68.5*	*63.7*
32	LoRA	83.9M	10.0h	57.9	***56.9***	69.8	69.2	10.8h	50.2	44.4	64.4	57.0
	MoRA	83.9M	12.4h	55.6	53.0	69.0	68.3	14.0h	42.2	42.2	56.4	56.0
	RaSA	83.9M	11.5h	***59.5***	56.2	*72.5*	***71.4***	12.5h	***55.7***	***55.7***	***70.0***	***65.7***

Data Scale

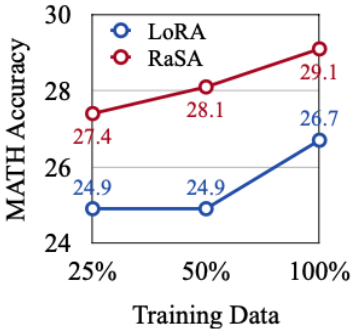


Figure 7: MATH performance of scaled data.

Model Scale

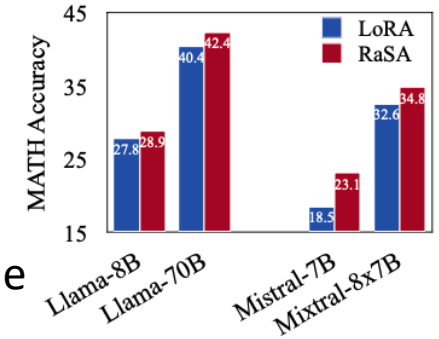


Figure 6: MATH performance of scaled models.

RaSA learns more and faster than LoRA

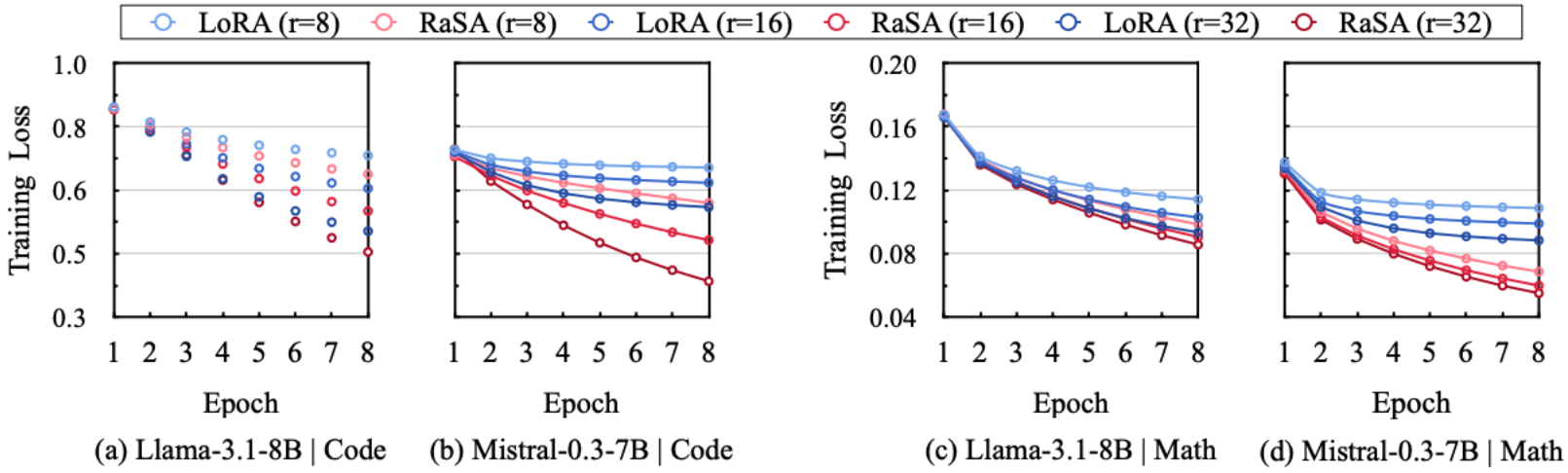


Figure 4: **RaSA learns more and faster than LoRA**. Training curves of LoRA and RaSA with different ranks. RaSA consistently outperforms LoRA with the same rank across models and tasks.

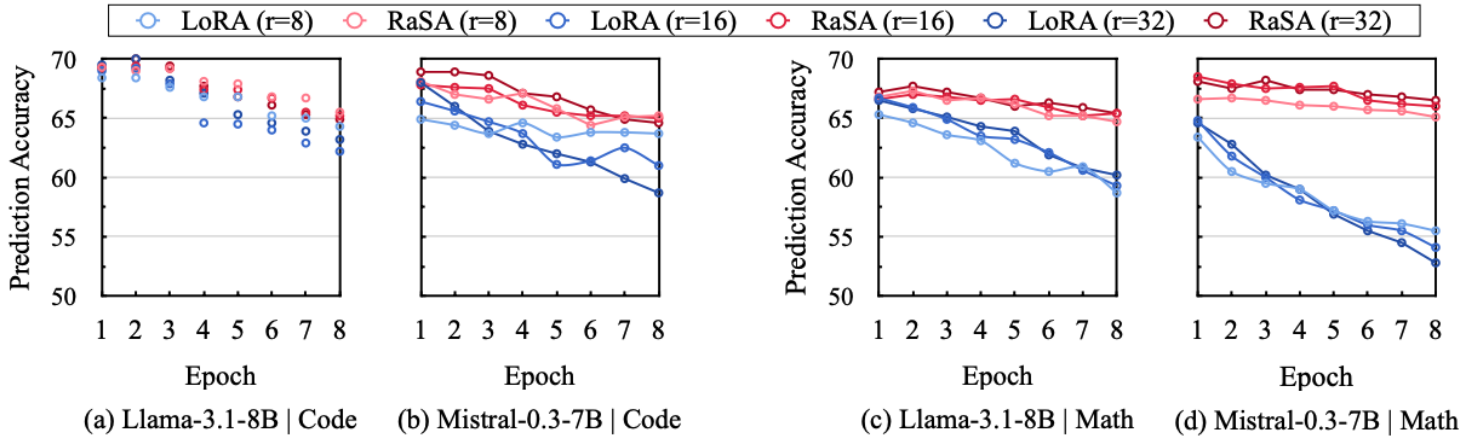


Figure 5: **RaSA forgets less than LoRA**. Y-axis shows the average of prediction accuracy on three benchmarks to evaluate model's forgetting. Higher prediction accuracy denotes less forgetting.

Thanks