Math Neurosurgery: Isolating Language Models' Math Reasoning Abilities Using Only Forward Passes

Bryan R. Christ   Zack Gottesman   Jonathan Kropko   Thomas Hartvigsen
University of Virginia
brc4cb@virginia.edu,qdw5jf@virginia.edu,jk8sd@virginia.edu,hartvigsen@virginia.edu

Abstract

Math reasoning is an active area of Large Language Model (LLM) research because it is a hallmark of artificial intelligence and has implications in several domains, including math education. However, few works have explored how math reasoning is encoded within LLM parameters and if it is a skill that can be isolated within models. Doing so could allow targeted intervention to improve math performance without altering non-math behavior and foster understanding of how models encode math reasoning. We introduce Math Neurosurgery (*MathNeuro*), a computationally efficient method we use to isolate math-specific parameters in LLMs using only forward passes. MathNeuro builds on existing work by using weights and activations to calculate parameter importance, but isolates math-specific parameters by filtering out those important for general language tasks. Through pruning parameters MathNeuro identifies, we delete a LLM's math reasoning ability without significantly impacting its general language ability. Scaling the identified parameters by a small constant improves a pretrained or instruction-tuned LLM's performance by 4-17% on GSM8K and 5-35% on MATH while leaving non-math behavior unaltered. MathNeuro is also data efficient: most of its effectiveness holds when identifying math-specific parameters using a single sample. MathNeuro highlights the potential for future work to intervene on math-specific parameters.[1]

1 Introduction

Math reasoning, or solving math problems with logic, is an active area of LLM research because it represents artificial intelligence (e.g., Ahn et al. 2024; Li et al. 2024b) and has implications in many domains, including math education (e.g, Christ et al. 2024; Wang et al. 2024) and automated theorem proving (e.g., Song et al. 2024; Xin et al. 2024). Yet, few works have explored how LLMs encode math reasoning abilities in their parametric knowledge. Identifying math-specific parameters could be beneficial for many reasons, including a) targeting the right parameters to intervene on to improve a model's math reasoning ability as others have done in other domains (e.g., Tang et al. 2024; Suau et al. 2024), b) doing so without altering behavior on other tasks like these works have done in their domains, and c) fostering knowledge of how LLMs encode math reasoning. While some works explore how different math concepts or terms are stored or processed in model layers or neurons (e.g., Hanna et al. 2023; Rai and Yao 2024; Stolfo et al. 2023), none have developed a method for isolating parameters for math reasoning.

Outside of math reasoning, several works have explored how to identify neurons or parameters associated with particular knowledge or skills in LLMs (Chang et al., 2024; Dai et al., 2022; Panigrahi et al., 2023; Tang et al., 2024; Wang et al., 2022). While some methods are computationally expensive because they use gradient information, which may not be feasible for large models (e.g., Panigrahi et al. 2023), others are easier to compute because they rely on information obtained through forward passes, particularly as captured by activations (e.g., Tang et al. 2024). However, it is unknown if these domain-specific methods for single skill identification can effectively isolate a broad concept like math reasoning, which may be entangled with many other abilities within a LLM (e.g., reading comprehension, general knowledge).

We conduct the first study of parameter importance in LLMs for math reasoning. We apply two state-of-the-art (SOTA) gradient-free parameter importance methods to math reasoning. We find one of these methods, LAPE (Tang et al., 2024), consistently fails to identify math-specific neurons across models, while the other, Wanda (Sun et al., 2023), identifies parameters important for math,

[1]Our experimental code can be found at https://github.com/bryanchrist/MathNeuro

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 24803–24840
July 27 - August 1, 2025

but is unable to isolate math-specific parameters because the parameters it identifies overlap significantly with those important for other tasks. To address these limitations of existing methods, we develop a new method called Math Neurosurgery (MathNeuro) we use to isolate math-specific parameters. Building on Wanda, MathNeuro uses weights and activations to calculate parameter importance and achieve a context-aware representation of importance. MathNeuro extends beyond Wanda by adding task-specific parameter identification. Specifically, to isolate parameters important for math and not other abilities, MathNeuro filters out identified parameters that are found to be important for other general language understanding tasks.

We provide evidence that MathNeuro effectively isolates math-specific parameters by evaluating it with five LLMs from 1-8B parameters. Pruning parameters identified by MathNeuro effectively deletes a model's math reasoning ability. Despite destroying math reasoning, pruning these parameters results in a performance drop on other, non-math tasks similar to the impact of random parameter pruning. We also find that scaling up MathNeuro-identified parameters by a small universal factor can boost both instruction-tuned and pre-trained LLMs' GSM8K (Cobbe et al., 2021) or MATH (Hendrycks et al., 2021b) performance by 4-17% or 5-35% across models, respectively. We further show that our method is data efficient: MathNeuro is almost as effective using only a *single* sample to calculate parameter importance. In addition, we show MathNeuro consistently identifies a similar subset of parameters as math-specific across different sets of samples and that these parameters generalize across math reasoning tasks. We find math-specific parameters are located roughly evenly throughout a model's decoder blocks, suggesting math reasoning is likely encoded throughout a model's parameters rather than being concentrated in a specific layer or layers.

Our key contributions are as follows:

- We design MathNeuro, a simple yet effective way to isolate LLM math reasoning by filtering out parameters important for other tasks.
- We demonstrate the effectiveness of this method by showing that deleting parameters identified by MathNeuro destroys a model's math performance and scaling them by a universal factor can increase it by 4-35%.
- We verify MathNeuro isolates math-specific parameters by showing pruning or scaling them does not significantly impact non-math performance more than random perturbation.

2 Related Work

Skill and Knowledge Localization in LLMs Several works have explored skill and knowledge localization in language models, although none focus on math specifically (Bau et al., 2018; Chang et al., 2024; Dalvi et al., 2018; Dai et al., 2022; Dalvi et al., 2020; Gurnee et al., 2023; Kojima et al., 2024; Leng and Xiong, 2024; Panigrahi et al., 2023; Radford et al., 2017; Suau et al., 2024; Sun et al., 2023; Tang et al., 2024; Wang et al., 2022; Xin et al., 2019; Zhao et al., 2024). Many methods use gradient information to calculate parameter importance, which is computationally infeasible for large models (Dai et al., 2022; Leng and Xiong, 2024; Panigrahi et al., 2023; Wang et al., 2022). However, others are more lightweight and calculate parameter importance using only forward passes, predominately through using information obtained through activation values (Kojima et al., 2024; Suau et al., 2024; Sun et al., 2023; Tang et al., 2024; Zhao et al., 2024). While these methods may find parameters important for the domains they study, it is unclear if they could identify parameters important for math reasoning, which could be distributed throughout a model or interwoven with other important natural language abilities given the task's complexity. To identify important parameters, MathNeuro builds upon Wanda (Sun et al., 2023), a SOTA LLM pruning method that prunes parameters *unimportant* for a model's output as measured by the smallest absolute value of weights times activations. MathNeuro inverts Wanda by identifying the *most important* parameters for a task and isolates math-specific parameters by filtering out parameters important for non-math, general language tasks through introducing task-specific data into the importance calculation.

Math Skill Localization in LLMs Some studies have explored how math knowledge is encoded within LLMs (Hanna et al., 2023; Nikankin et al., 2024; Rai and Yao, 2024; Stolfo et al., 2023; Zhang et al., 2024; Zhu et al., 2025). These works focus on how and where particular math concepts and key phrases such as addition and subtraction are processed by LLMs. While these findings are insightful, they do not identify parameters critical for a model's overall math performance but rather ones relating to processing different math concepts.

3 Methods

We propose MathNeuro, a parameter identification method that calculates importance using only forward passes. First, we separately identify LLM parameters important for a math task and a non-math, general language task using samples for each task. Next, MathNeuro isolates math-specific parameters by taking the subset that are important for the math task but not for the non-math task. While MathNeuro may work for other, non-math tasks, we study math specifically. We describe the problem setup and our method in more detail below.

3.1 Preliminaries

Identifying parameters important for math reasoning in LLMs is beneficial because it is a critical AI capability, is understudied in interpretability work, has implications in several domains, and thus is an interesting test domain. However, this is nontrivial given that math reasoning not only involves direct computation, but also natural language reasoning. Thus, it may be difficult to distinguish parameters important only for math reasoning from those important for general language. Indeed, other work has found significant overlap between parameters important for different tasks (Tang et al., 2024).

3.2 Identifying Top Parameters

MathNeuro adapts Wanda (Sun et al., 2023), a SOTA LLM pruning method that prunes, or removes, parameters unimportant for a model's output. Wanda identifies parameters to prune using the absolute value of weights times activations for an input, providing a context-aware representation of parameter importance. Wanda produces a score S_{ij} for weight j in neuron i within a weight matrix:

$$S_{ij} = |W_{ij}| \cdot \|X_j\|_2$$

where W_{ij} represents the weight, $|\cdot|$ is the absolute value operator, and $\|X_j\|_2$ is the ℓ_2 norm of the j-th feature aggregated across input tokens to normalize the input X, or activation values. Wanda then prunes the parameters with the smallest scores. Wanda considers both weights and activations as elements of parameter importance because small but highly activated weights can be highly influential, while large but lightly activated weights may be less influential. MathNeuro inverts Wanda by identifying the parameters with the largest weights times activations as being most important to a given task.

3.3 Isolating Math-specific Parameters

While naively identifying the parameters with the highest absolute value of weights times activations may find parameters important for a given task, it may not isolate the parameters important for that task *only*, as discussed above. Thus, we calculate parameter importance for other unrelated tasks and use the disjoint set between these sets of parameters as the ones that are math-specific, which is the critical innovation of MathNeuro. To do this, we separately sum[2] the absolute value of weights times activations for each parameter in attention and MLP layers for each data point k across N samples from a math dataset and an unrelated natural language task dataset. We focus on attention and MLP layers because recent work has found that knowledge and skills are often distributed in these two model components (Wei et al., 2024; Yin et al., 2024). Using this formulation, we compute scores for each parameter over math and non-math inputs:

$$S_{ij}^{\text{math}} = \sum_{k=1}^{N} |W_{ij}| \cdot \|X_j^k\|_2 \text{ for } X \in \mathcal{D}_{\text{math}}$$

$$S_{ij}^{\text{non-math}} = \sum_{k=1}^{N} |W_{ij}| \cdot \|X_j^k\|_2 \text{ for } X \in \mathcal{D}_{\text{non-math}}$$

Then, we separately identify the top K% of parameters with the highest score for each task in each layer. Lastly, we take the subset of parameters most important for the math task that are not in the set of parameters most important for the unrelated task, or $T_{math} = \text{TopK}_{math} \setminus \text{TopK}_{non-math}$.

4 Experiments

We next validate if MathNeuro successfully identifies math-specific parameters. We compare against SOTA alternatives and a simple baseline in two settings: 1) pruning parameters identified as important for math and 2) scaling these parameters. Pruning or scaling task-specific parameters is equivalent to the approach recent work has taken to deactivate or more highly activate neurons identified as language or knowledge specific (Kojima et al., 2024; Suau et al., 2024; Tang et al., 2024; Zhao et al., 2024), respectively, but intervenes on the weight rather than

[2]This summation is akin to gradient-based identification methods summing gradients over inputs (e.g., Das et al. 2023).

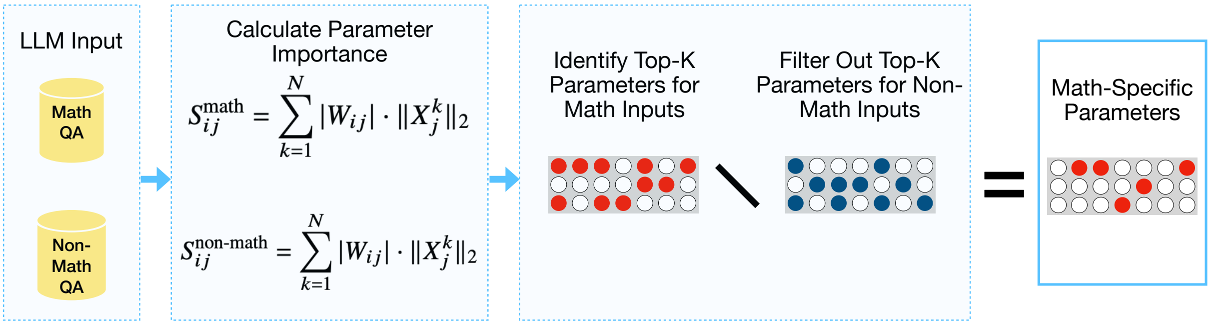

Figure 1: Overview of MathNeuro. First, we sum weights times activations over N samples for math and non-math inputs, finding the top-K parameters for each input type. Next, we find math-specific parameters by filtering out parameters important for non-math inputs.

activation level. We show the impact of each intervention on both math and non-math performance across five LLMs ranging from 1-8B parameters. We perform parameter identification experiments using 500 samples and a single sample.

4.1 Experimental Setup

Models We evaluate five LLMs of varying sizes: Phi 1.5 (1B) (Li et al., 2023), Llama 3.2 1B Instruction Tuned (IT) (MetaAI, 2024b), Gemma 2 2B IT (Team et al., 2024), Llama 3.2 3B IT (MetaAI, 2024b), and Llama 3.1 8B IT (MetaAI, 2024a). We display results for Llama 3.2 1B IT below and report results for the other models in Appendices A, B, C, and D, which follow similar trends to those discussed below. We focus on instruction tuned models to evaluate if MathNeuro can successfully identify math-specific parameters in models that a) perform well at math given their size and b) are trained for a range of tasks, which means it may be more difficult to identify math-specific parameters. Phi 1.5 serves as a baseline for if MathNeuro works for a pretrained, non-IT model.

Datasets For identifying math-specific parameters, we use the popular and high-quality GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021b) datasets. We calculate parameter importance using the GSM8K or MATH training split and evaluate the impact of each method on the GSM8K or MATH test split. We report GSM8K results below and MATH results in Appendix F given the GSM8K results replicate for the MATH dataset. Following prior work (Agarwal et al., 2024; Brown et al., 2024; Lee et al., 2024; Li et al., 2024a), we subset the GSM8K test split to the same 200 random samples for every model for experimental efficiency. For identifying parameters important for non-math tasks and measuring performance drops after eliminating math-specific parameters, we follow recent work that assesses catastrophic forgetting in LLMs (Luo et al., 2024) by using RACE (Lai et al., 2017) for measuring reading comprehension and MMLU (Hendrycks et al., 2021a) for measuring general knowledge. These datasets are general language understanding tasks that are different from math reasoning. While MMLU contains some math-related questions, it assesses a variety of knowledge that, in aggregate, is mostly not math-specific. We conduct all evaluations using the Eleuther AI LM Evaluation Harness (Gao et al., 2024) and use an 8-shot chain-of-thought (CoT) prompting format for GSM8K, as is standard.

Baselines We compare MathNeuro to three identification methods computed using forward passes:

(a) *Wanda* (Sun et al., 2023): We calculate parameter importance for math inputs and choose the top K% of parameters without filtering out those important for other unrelated tasks.

(b) *Language Activation Probability Entropy (LAPE)* (Tang et al., 2024): LAPE finds language-specific neurons by thresholding activation probabilities as calculated by samples for each language under consideration. We use GSM8K, MMLU, and RACE for calculating task-specific neurons using this method. Using LAPE allows us to determine if existing activation-only parameter identification methods can isolate math-specific parameters.

(c) *Random Parameter Identification*: As a sanity check, we randomly select the same number of parameters as those identified by MathNeuro when using MMLU or RACE as $\mathcal{D}_{\text{non-math}}$.

4.2 Pruning Top Math Parameters

To test if the four parameter identification methods (MathNeuro and three baselines) identify parameters important for math reasoning, we iden-

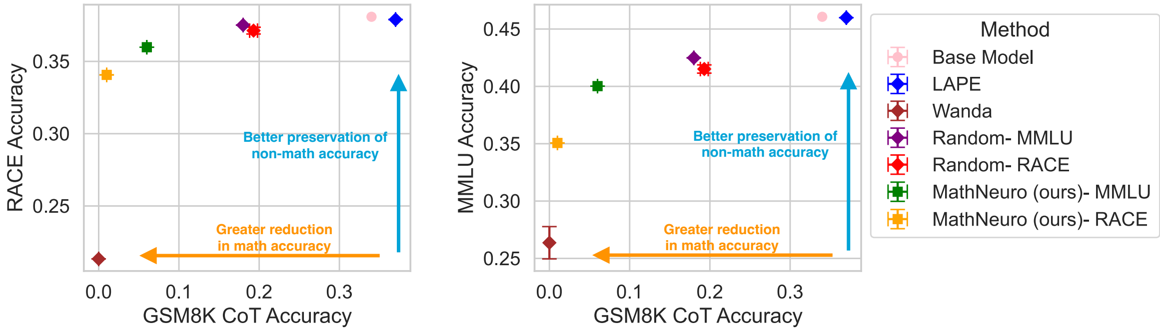

Figure 2: Effect of ***pruning*** identified parameters on math and non-math performance for Llama 3.2 1B IT with $\text{TopK}_{math} = \text{TopK}_{non-math} = 15\%$. Ideal methods fall in the top left of the plot. MMLU and RACE denote the dataset used as $\mathcal{D}_{\text{non-math}}$. Horizontal and vertical lines represent 95% confidence intervals for each point.

tify important parameters using each method for each model and prune them (set them to 0). We then compare each model's GSM8K, RACE, and MMLU accuracy to their own unedited performance. We do this five times for each model with different random subsets of 500 samples from each dataset to identify the average performance of each method. We identify the top .01, .1, .5, 1, 2.5, 5, 10 and 15% of parameters for each comparison and report the parameter proportion with the best performance. Appendix A explores how parameter proportion impacts GSM8K performance; notably, this hyperparameter does not impact performance for the comparison methods Wanda or LAPE.

Figures 2, 9, 10, 11 and 12 show results from this experiment. An ideal method would fall in the top left of these plots, meaning math performance (GSM8K) is deleted while non-math performance (RACE and MMLU) is maintained. As seen in the figures, MathNeuro and Wanda eliminate math performance across models, while LAPE is unable to identify parameters important for math. However, while Wanda also destroys each model's ability to perform non-math tasks, MathNeuro effectively isolates math-specific parameters across models, as shown in non-math performance decreases that are similar to the effect of random pruning.

4.3 Scaling Top Math Parameters

We next evaluate performance when more highly activating math-specific parameters by scaling the weights by a universal factor. For smaller models, we find the scalar 1.1 works best, while for larger models (Llama 3.1 8B IT), a smaller factor (1.01) works better. While we leave a rigorous study of this hyperparameter to future work due to its computational expense, see Appendix H for our experimentation with scale factors. As in Section 4.2, we scale the parameters each method identifies based on 500 random samples from each relevant dataset and repeat the process five times, reporting the parameter proportion that performs best.

Figures 3, 14, 15, 16, and 17 display results from this experiment. An ideal method would fall in the top right of these plots, meaning GSM8K accuracy increases while non-math performance is maintained. As shown in these figures, scaling parameters identified by MathNeuro results in a GSM8K performance increase of 4-17% across models, while scaling Wanda-identified parameters tends to either harm or slightly improve performance. As with pruning, LAPE has no effect for most models except for increasing GSM8K performance for Gemma 2 2B IT. Scaling random parameters can help for some models, although the effect is not consistent across models. Each parameter identification method does not harm performance on RACE or MMLU, suggesting scaling's impact tends to be localized to math performance.

4.4 MathNeuro with a Single Sample

If a method can identify math-specific parameters using a single sample, then it could inform math interventions for settings where data are limited such as for assessing a specific math operation or topic. To test this with MathNeuro and the baselines, we conduct experiments to identify parameters based on a single math and non-math input. We then prune or scale parameters identified by each method and run each experiment five times using different random samples from each dataset. As shown in Figures 4, 18, 19, 20 and 21, MathNeuro performs best at isolating math-specific parameters when pruning using a single sample, as shown

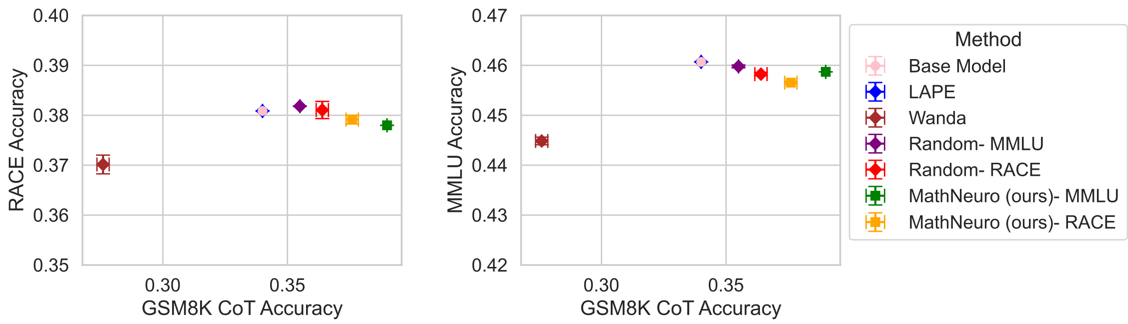

Figure 3: Effect of ***scaling*** identified parameters by 1.1 on math and non-math performance for Llama 3.2 1B IT with $\text{TopK}_{math} = \text{TopK}_{non-math} = 5\%$. Ideal methods fall in the top right of the plot. MMLU and RACE denote the dataset used as $\mathcal{D}_{\text{non-math}}$. Horizontal and vertical lines represent 95% confidence intervals for each point.

in lower drops in non-math performance relative to Wanda. However, these performance drops are larger than when using more samples, suggesting additional samples help MathNeuro more effectively isolate math-specific parameters.

As shown in Figures 5, 22, 23, 24, and 25, we see similar or smaller, but still meaningful, boosts in GSM8K accuracy when scaling parameters MathNeuro identifies using one math and non-math sample. While random scaling sometimes helps as observed in Section 4.3, the effect is again not consistent across models. In some cases, LAPE and Wanda increase GSM8K accuracy, though the effects are not consistent across models. For all methods, there is no meaningful performance drop in MMLU or RACE accuracy, suggesting scaling's impact on non-math performance is still minor.

4.5 MathNeuro Parameter Consistency, Number, Location and Qualitative Impact

Next, we conduct experiments to explore if MathNeuro identifies the same set of parameters as math-specific across different random subsets of math and non-math data and the number and location of these parameters. We report results for these experiments in the sections below using RACE as the non-math dataset and equivalent results for using MMLU as the non-math dataset in Appendix I. We also conduct a qualitative evaluation of model outputs after pruning or scaling parameters MathNeuro identifies to explore how math and non-math outputs are affected by the method. All experiments are conducted using Llama 3.2 1B IT.

Consistency of Math-specific Parameters We next explore if MathNeuro consistently identifies the same parameters as math-specific across different random subsets from a math and non-math dataset. This allows us to identify if math reasoning is in fact reliably concentrated in a subset of model parameters like the experiments above suggest. We first identify math-specific parameters using MathNeuro on two different random subsets from a math and non-math dataset. Next, we calculate the percentage overlap between the parameters identified in both subsets. We do this five times for different sample sizes (1, 10, 100, 500, and 1,000) and for calculating different proportions of top parameters from each dataset. This allows us to construct confidence intervals and see how parameter identification consistency varies when calculating based on different sample sizes and top parameter proportions. As shown in Figure 6 and Appendix I, with 100 or more samples, roughly 95% or more of the parameters MathNeuro identifies overlap between two random subsets regardless of the proportion of top parameters calculated, which shows that the method is able to consistently identify the most important parameters for math performance and that these parameters are largely invariant with regard to the subset of data used to calculate them.

Number and Location of Math-specific Parameters We next examine the proportion of parameters MathNeuro identifies as math-specific. We first identify math-specific parameters using random subsets from each dataset. Next, we calculate the percentage of the top K% of parameters that are identified as math-specific using those subsets. We repeat this five times for different sample sizes and top K%s to construct confidence intervals.

As shown in Figure 7 and Appendix I, while the most parameters are identified as math-specific when calculating importance with one sample due

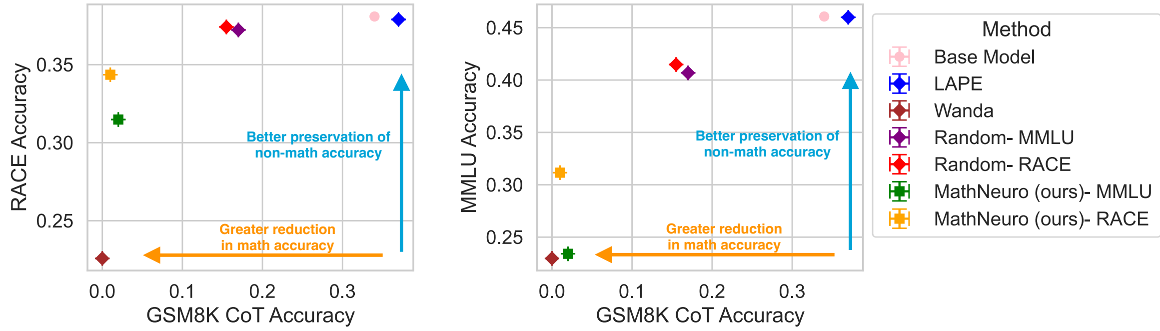

Figure 4: Effect of ***pruning*** identified parameters on math and non-math performance for Llama 3.2 1B IT with TopK_{math} = $\text{TopK}_{non-math}$ = 10% ***based on one sample***. Ideal methods fall in the top left of the plot. MMLU and RACE denote the dataset used as $\mathcal{D}_{\text{non-math}}$. Horizontal/vertical lines show each point's 95% confidence intervals.

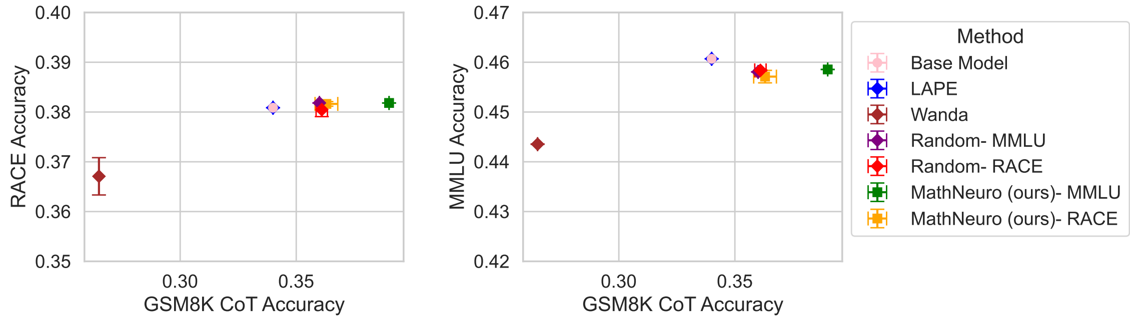

Figure 5: Effect of ***scaling*** identified parameters by 1.1 on math and non-math performance for Llama 3.2 1B IT with TopK_{math} = $\text{TopK}_{non-math}$ = 2.5% ***based on one sample***. Ideal methods fall in the top right. MMLU and RACE denote the dataset used as $\mathcal{D}_{\text{non-math}}$. Horizontal/vertical lines show each point's 95% confidence intervals.

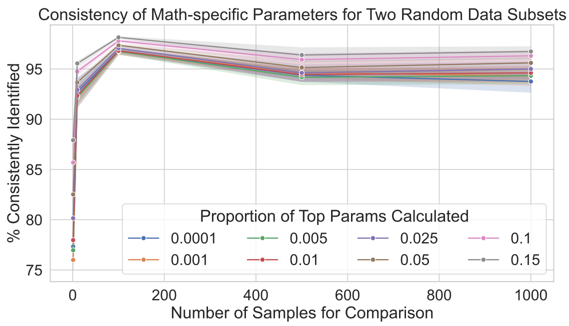

Figure 6: Consistency of math-specific parameters identified by MathNeuro for Llama 3.2 1B IT when identifying using GSM8K compared to RACE.

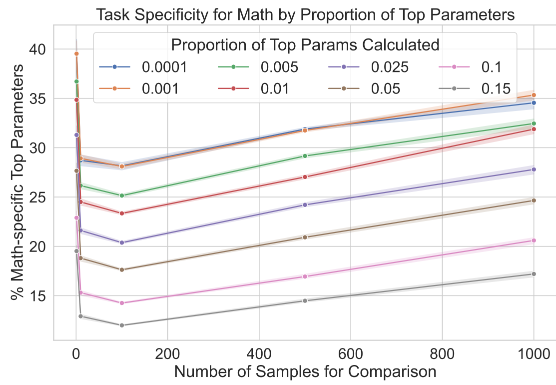

Figure 7: Percentage of top parameters that are math-specific as identified by MathNeuro for Llama 3.2 1B IT when identifying using GSM8K compared to RACE.

to randomness, the amount of math-specific parameters identified by MathNeuro generally increases with the number of samples considered for all proportions of top parameters calculated. The relatively high amount of overlap in top parameters between tasks displayed in these figures is likely why MathNeuro performs better than existing parameter identification methods that do not filter out parameters important for other tasks. The percentage of math-specific parameters in the top K% of parameters declines as the proportion of top parameters calculated increases because as this proportion increases, more of the model's top parameters are considered. These top parameters are likely more task-invariant than those found when considering a smaller percentage of the model's top parameters.

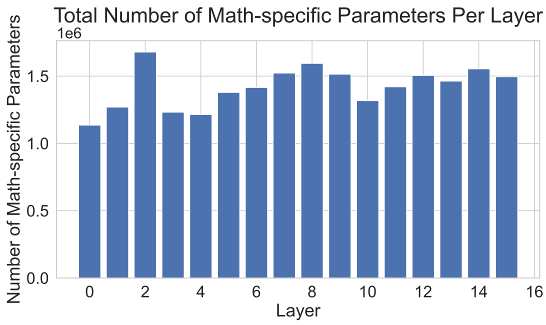

Figure 8: Distribution of math-specific parameters identified by MathNeuro for Llama 3.2 1B IT when identifying using GSM8K compared to RACE.

To explore where math-specific parameters are located within a model, we sum the number of parameters MathNeuro identifies in each decoder block for Llama 3.2 1B IT. To do this, we calculate the top 15% of parameters, which is the parameter proportion for which MathNeuro performs best for this model. As shown in Figure 8 and Appendix I, the number of math-specific parameters MathNeuro identifies is relatively consistent across decoder blocks when using either RACE or MMLU for parameter identification. This suggests that models encode math reasoning by distributing the capability throughout their parametric knowledge rather than concentrating it in a few layers. The parameters identified in these figures correspond to just 1.51% and 1.84% of the model's overall parameters when calculating parameter importance based on MMLU and RACE, respectively, despite being responsible for nearly all of its math performance.

Qualitative Analysis To verify model outputs are still coherent after pruning or scaling, we conduct a qualitative analysis of outputs before and after pruning or scaling parameters MathNeuro identifies. As shown in Tables 1, 3, and 4, only the output for answering a GSM8K question becomes incoherent after pruning math-specific parameters. Before pruning, the model correctly solves the GSM8K problem; after pruning, it fails to generate an effective CoT. The pruned model effectively responds to RACE questions, although it gets the question wrong before and after pruning. The pruned model generates coherent output to MMLU questions, though it gets the answer right when using RACE as $\mathcal{D}_{\text{non-math}}$ and wrong when using MMLU as $\mathcal{D}_{\text{non-math}}$. These findings confirm our quantitative findings showing the model can still perform non-math tasks after pruning math-specific parameters, although it experiences a performance drop similar to that obtained from random pruning. As shown in Tables 2, 3, and 5, the scaled model's outputs for RACE and MMLU questions remain mostly unmodified, while it correctly solves a GSM8K question after scaling based on MMLU as $\mathcal{D}_{\text{non-math}}$ that it solved incorrectly before scaling. These findings parallel our quantitative findings that math reasoning increases post-scaling while non-math performance remains unchanged.

4.6 Impact of MathNeuro on Unseen Tasks

To explore if math-specific parameters MathNeuro identifies are consistently important across unseen math tasks and unimportant for unseen general language or non-math reasoning tasks, we repeat the pruning experiments reported in Section 4.2 using GSM8K as the math dataset and MMLU or RACE as the non-math dataset and evaluate the pruned model on 8 unseen tasks (see Appendix E for implementation details). Unseen tasks include 5 that are non-math reasoning or general language tasks (HellaSwag, MuTual, PIQA, WikiText, and WinoGrande) (Zellers et al., 2019; Cui et al., 2020; Bisk et al., 2020; Merity et al., 2016; Sakaguchi et al., 2021) and 3 that are math tasks, both in domain (EGSM; Christ et al. 2024) and out of domain (MATH, MATHQA) (Hendrycks et al., 2021b; Amini et al., 2019). The figures in Appendix E display this experiment's results where the Wanda baseline represents the lowest possible bound on performance for a task given that it deletes all of a model's top parameters. Across models, pruning parameters identified by MathNeuro effectively deletes math performance on both in domain and out of domain tasks while mostly maintaining performance on general language and non-math reasoning tasks. These findings expand those discussed in Section 4.2 by showing that math-specific parameters MathNeuro identifies are universally important across math reasoning tasks and unimportant across non-math tasks regardless of the math dataset used for identification. This generalizability across math tasks allays concerns that parameters identified in one dataset are really targeted towards characteristics of that dataset and, instead, shows MathNeuro isolates parameters important for math reasoning broadly.

5 Conclusion

Although math reasoning is an active area of LLM research, few works have explored how it is encoded within LLM parameters and if it is a skill that can be isolated within a model. We introduce MathNeuro, a forward-only identification method we use to isolate math-specific parameters in LLMs. Through comprehensive experiments, we demonstrate MathNeuro's effectiveness by showing pruning or scaling the parameters it identifies can delete or reinforce a LLM's math reasoning ability, respectively, despite its simplicity and ease of calculation. Future work should build on this method by applying it to other domains and developing interventions for math-specific parameters that improve a model's performance on mathematical reasoning without catastrophic forgetting.

Limitations

While we comprehensively evaluate MathNeuro using several math and non-math datasets used in other works and focus our evaluations on math reasoning specifically, there are many other natural language and mathematical reasoning tasks models could be evaluated on. Future work should consider extending MathNeuro to these additional tasks and explore if MathNeuro can isolate parameters important for non-math tasks. While we used five recent models for our experiments, future work should also include additional models, especially those of larger sizes (>8B). Additionally, due to computational expense, we were unable to conduct a full hyperparameter sweep for an optimal universal scaling factor for parameters identified by MathNeuro, though the rough grid search we report in Appendix H highlights that larger scale factors tend to work better for smaller models and smaller scale factors tend to work better for larger models.

Ethics Statement

All data used in this paper come from open-access datasets and, therefore, should not contain any private sensitive information.

References

Rishabh Agarwal, Avi Singh, Lei M. Zhang, Bernd Bohnet, Luis Rosias, Stephanie Chan, Biao Zhang, Ankesh Anand, Zaheer Abbas, Azade Nova, John D. Co-Reyes, Eric Chu, Feryal Behbahani, Aleksandra Faust, and Hugo Larochelle. 2024. Many-Shot In-Context Learning. *arXiv preprint*. ArXiv:2404.11018 [cs].

Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. 2024. Large Language Models for Mathematical Reasoning: Progresses and Challenges. *arXiv preprint*. ArXiv:2402.00157 [cs].

Aida Amini, Saadia Gabriel, Peter Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. MathQA: Towards Interpretable Math Word Problem Solving with Operation-Based Formalisms. *arXiv preprint*. ArXiv:1905.13319 [cs].

Anthony Bau, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. 2018. Identifying and Controlling Important Neurons in Neural Machine Translation. *arXiv preprint*. ArXiv:1811.01157 [cs].

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. PIQA: Reasoning about Physical Commonsense in Natural Language. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7432–7439. Number: 05.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. 2024. Large Language Monkeys: Scaling Inference Compute with Repeated Sampling. *arXiv preprint*. ArXiv:2407.21787 [cs].

Ting-Yun Chang, Jesse Thomason, and Robin Jia. 2024. Do Localization Methods Actually Localize Memorized Data in LLMs? A Tale of Two Benchmarks. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3190–3211, Mexico City, Mexico. Association for Computational Linguistics.

Bryan R Christ, Jonathan Kropko, and Thomas Hartvigsen. 2024. MATHWELL: Generating Educational Math Word Problems Using Teacher Annotations. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 11914–11938, Miami, Florida, USA. Association for Computational Linguistics.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training Verifiers to Solve Math Word Problems. *arXiv preprint*. ArXiv:2110.14168 [cs].

Leyang Cui, Yu Wu, Shujie Liu, Yue Zhang, and Ming Zhou. 2020. MuTual: A Dataset for Multi-Turn Dialogue Reasoning. *arXiv preprint*. ArXiv:2004.04494 [cs].

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge Neurons in Pretrained Transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational*

Linguistics (Volume 1: Long Papers), pages 8493–8502, Dublin, Ireland. Association for Computational Linguistics.

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, Yonatan Belinkov, Anthony Bau, and James Glass. 2018. What Is One Grain of Sand in the Desert? Analyzing Individual Neurons in Deep NLP Models.

Fahim Dalvi, Hassan Sajjad, Nadir Durrani, and Yonatan Belinkov. 2020. Analyzing Redundancy in Pretrained Transformer Models.

Sarkar Snigdha Sarathi Das, Ranran Haoran Zhang, Peng Shi, Wenpeng Yin, and Rui Zhang. 2023. Unified Low-Resource Sequence Labeling by Sample-Aware Dynamic Sparse Finetuning. *arXiv preprint*. ArXiv:2311.03748 [cs].

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2024. A framework for few-shot language model evaluation. Version Number: v0.4.3.

Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. 2023. Finding Neurons in a Haystack: Case Studies with Sparse Probing.

Michael Hanna, Ollie Liu, and Alexandre Variengien. 2023. How does GPT-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. *Advances in Neural Information Processing Systems*, 36:76033–76060.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. Measuring Massive Multitask Language Understanding. *arXiv preprint*. ArXiv:2009.03300 [cs].

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring Mathematical Problem Solving With the MATH Dataset. *arXiv preprint*. ArXiv:2103.03874 [cs].

Takeshi Kojima, Itsuki Okimura, Yusuke Iwasawa, Hitomi Yanaka, and Yutaka Matsuo. 2024. On the Multilingual Ability of Decoder-based Pre-trained Language Models: Finding and Controlling Language-Specific Neurons. *arXiv preprint*. ArXiv:2404.02431 [cs].

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding Comprehension Dataset From Examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.

Nicholas Lee, Thanakul Wattanawong, Sehoon Kim, Karttikeya Mangalam, Sheng Shen, Gopala Anumanchipalli, Michael W. Mahoney, Kurt Keutzer, and Amir Gholami. 2024. LLM2LLM: Boosting LLMs with Novel Iterative Data Enhancement.

Yongqi Leng and Deyi Xiong. 2024. Towards Understanding Multi-Task Learning (Generalization) of LLMs via Detecting and Exploring Task-Specific Neurons. *arXiv preprint*. ArXiv:2407.06488 [cs].

Chen Li, Weiqi Wang, Jingcheng Hu, Yixuan Wei, Nanning Zheng, Han Hu, Zheng Zhang, and Houwen Peng. 2024a. Common 7B Language Models Already Possess Strong Math Capabilities. *arXiv preprint*. ArXiv:2403.04706 [cs].

Qintong Li, Leyang Cui, Xueliang Zhao, Lingpeng Kong, and Wei Bi. 2024b. GSM-Plus: A Comprehensive Benchmark for Evaluating the Robustness of LLMs as Mathematical Problem Solvers. *arXiv preprint*. ArXiv:2402.19255 [cs].

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023. Textbooks Are All You Need II: phi-1.5 technical report. *arXiv preprint*. ArXiv:2309.05463 [cs].

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2024. An Empirical Study of Catastrophic Forgetting in Large Language Models During Continual Fine-tuning. *arXiv preprint*. ArXiv:2308.08747 [cs].

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer Sentinel Mixture Models. _eprint: 1609.07843.

MetaAI. 2024a. Introducing Llama 3.1: Our most capable models to date.

MetaAI. 2024b. Llama 3.2: Revolutionizing edge AI and vision with open, customizable models.

Yaniv Nikankin, Anja Reusch, Aaron Mueller, and Yonatan Belinkov. 2024. Arithmetic Without Algorithms: Language Models Solve Math With a Bag of Heuristics. *arXiv preprint*. ArXiv:2410.21272.

Abhishek Panigrahi, Nikunj Saunshi, Haoyu Zhao, and Sanjeev Arora. 2023. Task-Specific Skill Localization in Fine-tuned Language Models. In *Proceedings of the 40th International Conference on Machine Learning*, pages 27011–27033. PMLR. ISSN: 2640-3498.

Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. 2017. Learning to Generate Reviews and Discovering Sentiment.

Daking Rai and Ziyu Yao. 2024. An Investigation of Neuron Activation as a Unified Lens to Explain Chain-of-Thought Eliciting Arithmetic Reasoning of LLMs. *arXiv preprint*. ArXiv:2406.12288 [cs].

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. WinoGrande: an adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Peiyang Song, Kaiyu Yang, and Anima Anandkumar. 2024. Towards Large Language Models as Copilots for Theorem Proving in Lean. *arXiv preprint*. ArXiv:2404.12534 [cs].

Alessandro Stolfo, Yonatan Belinkov, and Mrinmaya Sachan. 2023. A Mechanistic Interpretation of Arithmetic Reasoning in Language Models using Causal Mediation Analysis. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7035–7052, Singapore. Association for Computational Linguistics.

Xavier Suau, Pieter Delobelle, Katherine Metcalf, Armand Joulin, Nicholas Apostoloff, Luca Zappella, and Pau Rodríguez. 2024. Whispering Experts: Neural Interventions for Toxicity Mitigation in Language Models. *arXiv preprint*. ArXiv:2407.12824 [cs].

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. 2023. A Simple and Effective Pruning Approach for Large Language Models. *arXiv preprint*. ArXiv:2306.11695 [cs].

Tianyi Tang, Wenyang Luo, Haoyang Huang, Dongdong Zhang, Xiaolei Wang, Xin Zhao, Furu Wei, and Ji-Rong Wen. 2024. Language-Specific Neurons: The Key to Multilingual Capabilities in Large Language Models. *arXiv preprint*. ArXiv:2402.16438 [cs].

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju-yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. 2024. Gemma 2: Improving Open Language Models at a Practical Size. *arXiv preprint*. ArXiv:2408.00118 [cs].

Rose E. Wang, Ana T. Ribeiro, Carly D. Robinson, Susanna Loeb, and Dora Demszky. 2024. Tutor CoPilot: A Human-AI Approach for Scaling Real-Time Expertise. *arXiv preprint*. ArXiv:2410.03017.

Xiaozhi Wang, Kaiyue Wen, Zhengyan Zhang, Lei Hou, Zhiyuan Liu, and Juanzi Li. 2022. Finding Skill Neurons in Pre-trained Transformer-based Language Models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11132–11152, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yifan Wei, Xiaoyan Yu, Yixuan Weng, Huanhuan Ma, Yuanzhe Zhang, Jun Zhao, and Kang Liu. 2024. Does Knowledge Localization Hold True? Surprising Differences Between Entity and Relation Perspectives in Language Models. *arXiv preprint*. ArXiv:2409.00617 [cs].

Huajian Xin, Daya Guo, Zhihong Shao, Zhizhou Ren, Qihao Zhu, Bo Liu, Chong Ruan, Wenda Li, and Xiaodan Liang. 2024. DeepSeek-Prover: Advancing Theorem Proving in LLMs through Large-Scale Synthetic Data. *arXiv preprint*. ArXiv:2405.14333 [cs].

Ji Xin, Jimmy Lin, and Yaoliang Yu. 2019. What Part of the Neural Network Does This? Understanding LSTMs by Measuring and Dissecting Neurons. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5823–5830, Hong Kong, China. Association for Computational Linguistics.

Fangcong Yin, Xi Ye, and Greg Durrett. 2024. LoFiT: Localized Fine-tuning on LLM Representations. *arXiv preprint*. ArXiv:2406.01563 [cs].

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a Machine Really Finish Your Sentence? *arXiv preprint*. ArXiv:1905.07830 [cs].

Wei Zhang, Chaoqun Wan, Yonggang Zhang, Yiu-ming Cheung, Xinmei Tian, Xu Shen, and Jieping Ye. 2024. Interpreting and Improving Large Language Models in Arithmetic Calculation. *arXiv preprint*. ArXiv:2409.01659.

Yiran Zhao, Wenxuan Zhang, Guizhen Chen, Kenji Kawaguchi, and Lidong Bing. 2024. How do Large Language Models Handle Multilingualism? *arXiv preprint*. ArXiv:2402.18815 [cs].

Fangwei Zhu, Damai Dai, and Zhifang Sui. 2025. Language Models Encode the Value of Numbers Linearly. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 693–709, Abu Dhabi, UAE. Association for Computational Linguistics.

A GSM8K Pruning Results

Figures 9, 10, 11 and 12 show the results for pruning parameters identified by each method for Phi 1.5, Gemma 2 2B IT, Llama 3.2 3B IT, and Llama 3.1 8B IT, respectively. As shown in these figures, the results for each model closely mirror those discussed in Section 4.2, where MathNeuro performs the best at isolating math-specific parameters as shown by destroying GSM8K performance while having low drops in MMLU and RACE performance that are similar to the impact of pruning random model parameters.

Figure 13 shows the impact of parameter proportion on GSM8K performance when pruning parameters identified by each method for Llama 3.2 1B IT. As shown in the figure, GSM8K performance declines with increasing proportion of parameters considered when using MathNeuro until the parameter proportion reaches .05, at which point the effectiveness of the method levels off. GSM8K performance begins to increase after the top 10% of parameters are considered due to the top 15% of model parameters being more invariant across tasks, as shown in Section 4.5. For comparison methods, Wanda deletes math performance regardless of parameter proportion, LAPE actually increases performance, and pruning random parameters tends to hurt performance as the proportion of top parameters considered increases, which is expected. The other four models show similar trends when considering different proportions of top parameters.

B GSM8K Scaling Results

Figures 14, 15, 16 and 17 show the results for scaling parameters identified by each method for Phi 1.5, Gemma 2 2B IT, Llama 3.2 3B IT, and Llama 3.1 8B IT, respectively. As shown in these figures, the results for each model closely mirror those discussed in Section 4.3, where scaling parameters identified by MathNeuro consistently increases GSM8K performance by 3-6 percentage points across models, representing a 4-17% overall increase in math performance depending on the model.

C One Sample GSM8K Pruning Results

Figures 18, 19, 20 and 21 show the results for pruning parameters identified by each method for Phi 1.5, Gemma 2 2B IT, Llama 3.2 3B IT, and Llama 3.1 8B IT, respectively, when calculating parameter importance based on a single sample. As shown in these figures, the results for each model closely mirror those discussed in Section 4.4, where MathNeuro still performs the best at isolating math-specific parameters as shown by destroying GSM8K performance while having lower drops in MMLU and RACE performance than Wanda. However, as reported in Section 4.4, these results suggest additional samples help the method more effectively isolate math-specific parameters because the non-math drops in performance are larger than those shown in Appendix A, where we used 500 samples to calculate parameter importance.

D One Sample GSM8K Scaling Results

Figures 22, 23, 24 and 25 show the results for scaling parameters identified by each method for Phi 1.5, Gemma 2 2B IT, Llama 3.2 3B IT, and Llama 3.1 8B IT, respectively, when calculating parameter importance based on one sample. As shown in these figures, the results for each model closely

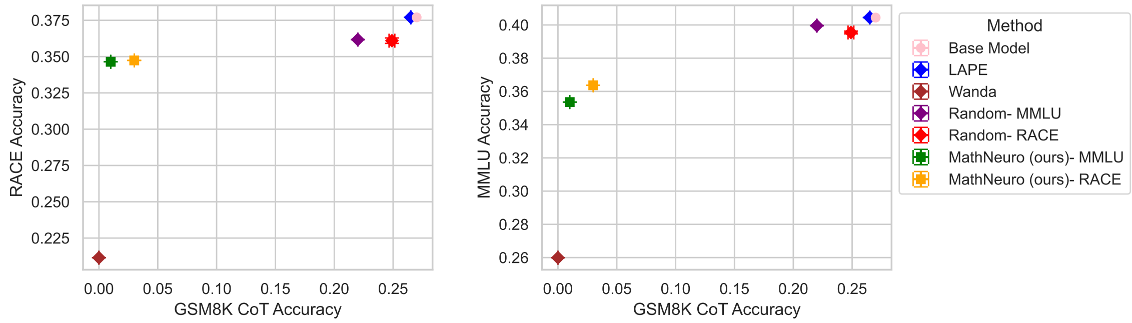

Figure 9: Effect of ***pruning*** identified parameters on math and non-math performance for Phi 1.5 based on calculating the top 5% of parameters. Ideal methods should fall in the top left of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$. Horizontal and vertical lines represent 95% confidence intervals for each point on the plot.

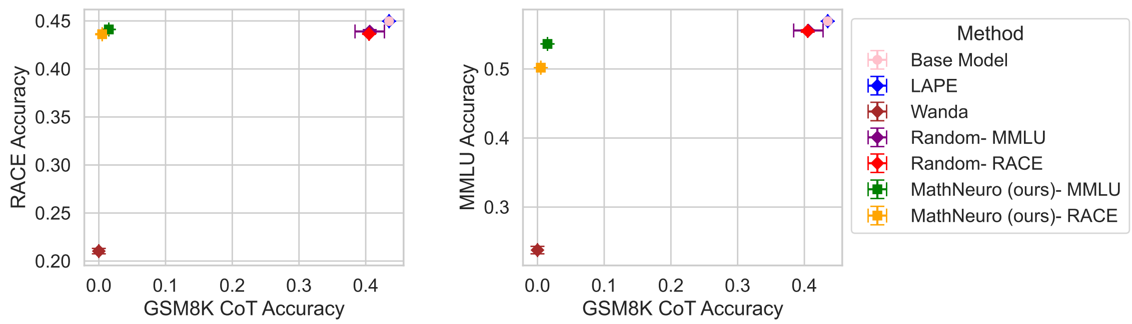

Figure 10: Effect of ***pruning*** identified parameters on math and non-math performance for Gemma 2 2B IT based on calculating the top 5% of parameters. Ideal methods should fall in the top left of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$. Horizontal and vertical lines represent 95% confidence intervals for each point on the plot.

mirror those discussed in Section 4.4, where scaling parameters identified by MathNeuro using a single sample consistently increases GSM8K performance across models. These increases are either similar to those reported in Appendix B, or smaller but still meaningful. For some models, the comparison methods can increase GSM8K performance when calculating parameter importance based off a single sample, but MathNeuro is the only method for which a meaningful positive increase is consistent across models.

E Impact of MathNeuro on Unseen Downstream Tasks

As discussed in Section 4.6, Figures 26, 27, 28, 29, and 30 display the results for evaluating unseen task performance after pruning parameters identified by MathNeuro using GSM8K as $\mathcal{D}_{\text{math}}$ and MMLU or Race as $\mathcal{D}_{\text{non-math}}$. In these figures, MuTual performance is unimpaired regardless of the parameter identification method used, suggesting that performance on this task is consistently equal to random guessing regardless of which parameters are pruned. All tasks are implemented using their default implementation in the Eleuther AI LM Evaluation Harness (Gao et al., 2024). For MATH, we use Eleuther's implementation of the 4-shot CoT Minerva prompt and for EGSM we use a fork of the Eleuther AI LM Evaluation Harness where we implemented the task using GSM8K's 8-shot CoT prompt given they are both grade school math datasets. For all tasks except MATH and EGSM, we run evaluations using the full testing split for each task given they are in multiple-choice format. For MATH and EGSM, which require long-form responses, we follow our other experiments by using the same set of random samples from each dataset for experimental efficiency. For MATH, we use

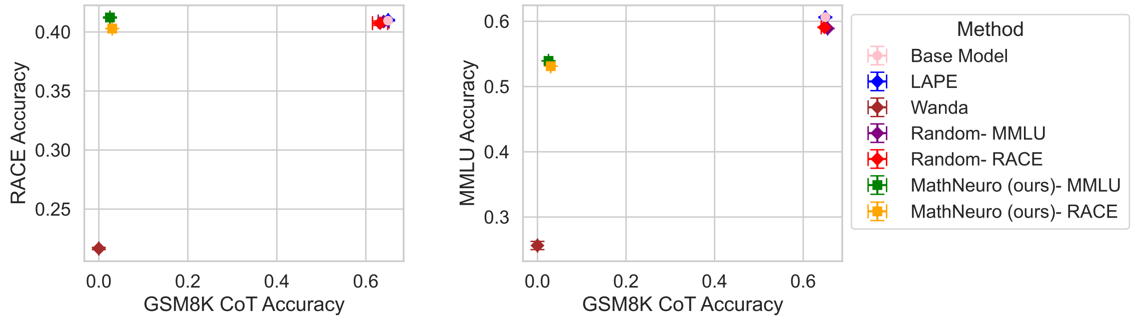

Figure 11: Effect of ***pruning*** identified parameters on math and non-math performance for Llama 3.2 3B IT based on calculating the top 2.5% (left) and 1% (right) of parameters. Ideal methods should fall in the top left of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$. Horizontal and vertical lines represent 95% confidence intervals for each point on the plot.

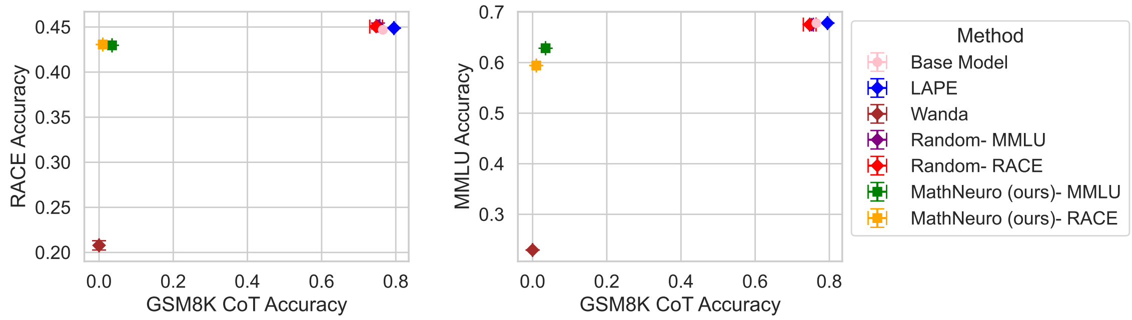

Figure 12: Effect of ***pruning*** identified parameters on math and non-math performance for Llama 3.1 8B IT based on calculating the top 1% of parameters. Ideal methods should fall in the top left of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$. Horizontal and vertical lines represent 95% confidence intervals for each point on the plot.

700 samples and for EGSM we use 100 due to its smaller size.

F MathNeuro Using the MATH Dataset

F.1 MATH Pruning Experiments

We replicate our GSM8K pruning experiments using MATH as the math dataset and MMLU or RACE as the non-math dataset. Similar to our approach with GSM8K, we subset the MATH testing split to the same 700 random samples for each model and experimental run for experimental efficiency. As with our other experiments, we use the Eleuther AI LM Evaluation Harness (Gao et al., 2024) for implementing the MATH evaluations, using their default implementation of the Minerva MATH prompt with 4-shot CoT examples. For each model, we use the exact same hyperparameters as those reported for the pruning experiments in Section 4.2 and Appendix A. As shown in Figures 31, 32, 33, 34, and 35, our GSM8K pruning results replicate when using MATH as the math dataset, as pruning math-specific parameters MathNeuro identifies effectively deletes MATH performance while leaving MMLU and RACE performance largely unaltered. These results buttress those reported in Section 4.6 by showing that MathNeuro's effectiveness in isolating math-specific parameters holds when using a different math dataset.

Using the same hyperparameters, we also replicate our one sample pruning experiments using MATH as $\mathcal{D}_{\text{math}}$. Results are shown in Figures 36, 37, 38, 39, and 40, where MathNeuro still performs best at isolating math-specific parameters when using a single MATH sample for parameter identification.

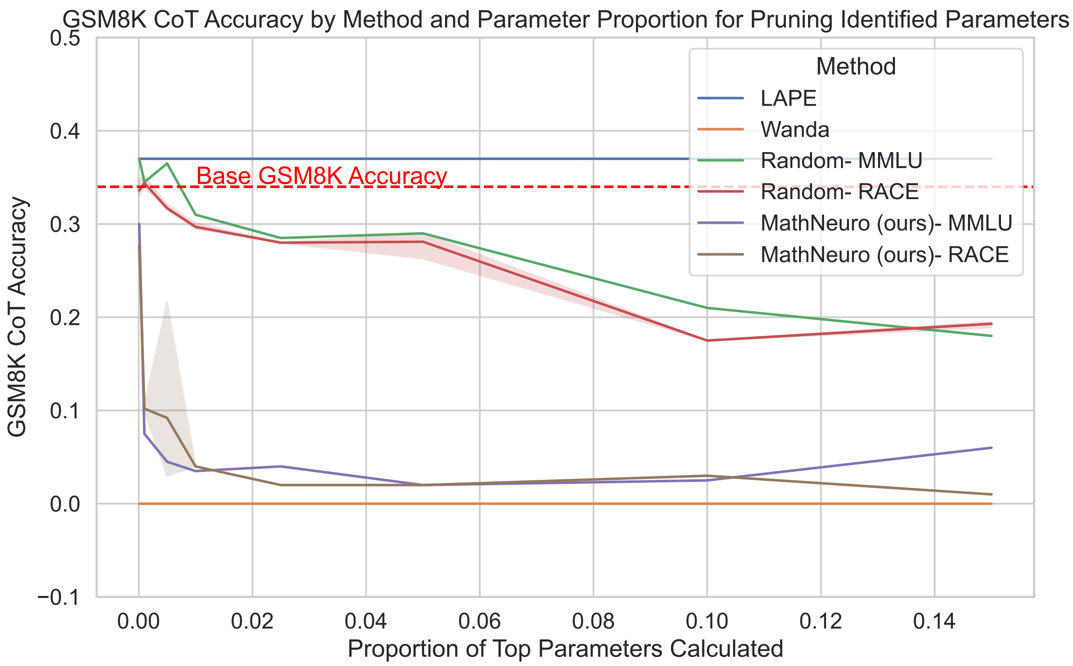

Figure 13: Impact of parameter proportion on GSM8K performance for ***pruning*** parameters identified by each method for Llama 3.2 1B IT when considering the top .01, .1, .5, 1, 2.5, 5, 10 and 15% of parameters.

F.2 MATH Scaling Experiments

We also replicate our GSM8K scaling experiments using MATH as the math dataset and MMLU or RACE as the non-math dataset. Similar to our MATH pruning experiments, we subset the MATH testing split to the same random samples for each model and experimental run for experimental efficiency and implement the MATH task with the same prompting approach described above. We use a smaller sample of the MATH testing split (350 samples) for our scaling experiments given that we conduct the rough grid search for an optimal scaling factor and parameter proportion described in Appendix H and Section 4.3, respectively, and using a larger set of samples would be computationally prohibitive. As shown in Figures 41, 42, 43, 44, and 45, our GSM8K scaling results replicate when using MATH as the math dataset, as scaling math-specific parameters MathNeuro identifies boosts MATH performance by a small but meaningful amount while leaving MMLU and RACE performance unchanged. These increases in performance correspond to increasing baseline MATH performance by 5-35% depending on the model. These results show that scaling parameters identified using MathNeuro boosts math performance even when using different math datasets.

Using the same scaling factors and conducting a small grid search for the optimal parameter proportion like we did in Section 4.4, we also replicate our one sample scaling experiments using MATH as $\mathcal{D}_{\text{math}}$. Results are shown in Figures 46, 47, 48, 49, and 50, where scaling MathNeuro-identified parameters using a single MATH sample for parameter identification still results in a small but meaningful boost in MATH performance while leaving non-math performance unaltered.

G Sample Outputs

Tables 1, 2, 3, 4, and 5 display sample outputs from Llama 3.2 1B IT before and after pruning or scaling parameters identified by MathNeuro. The tables display outputs for a GSM8K, RACE, or MMLU question.

H Scaling Factor Grid Search

Because an exhaustive grid search for the optimal scaling factor for MathNeuro would be computationally prohibitive, we used a rough bisection grid search to find a factor that worked best for each model for the GSM8K and MATH scaling experiments. For each model, we tried three scaling factors based on initial experiments that showed scale factors above 1.1 were too large: 1.01, 1.05, and 1.075. For GSM8K, for smaller models (Phi 1.5, Gemma 2 2B IT, Llama 3.2 1B IT, and Llama

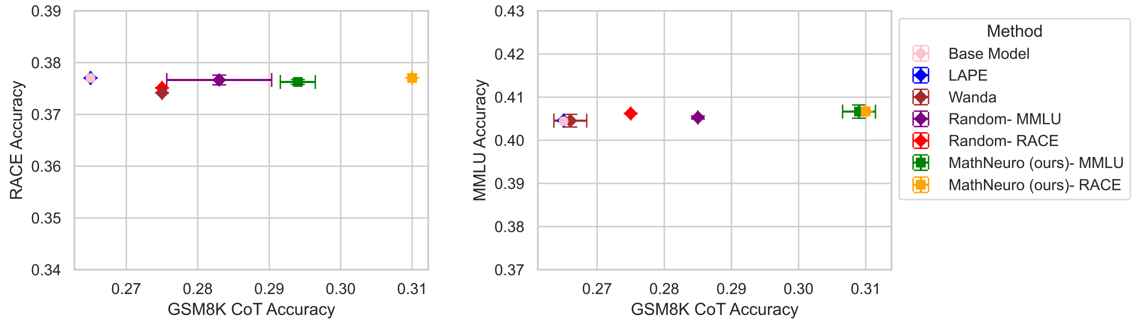

Figure 14: Effect of ***scaling*** identified parameters by 1.1 on math and non-math performance for Phi 1.5 based on calculating the top .1% (left) and .01% (right) of parameters. Ideal methods should fall in the top right of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$. Horizontal and vertical lines represent 95% confidence intervals for each point on the plot.

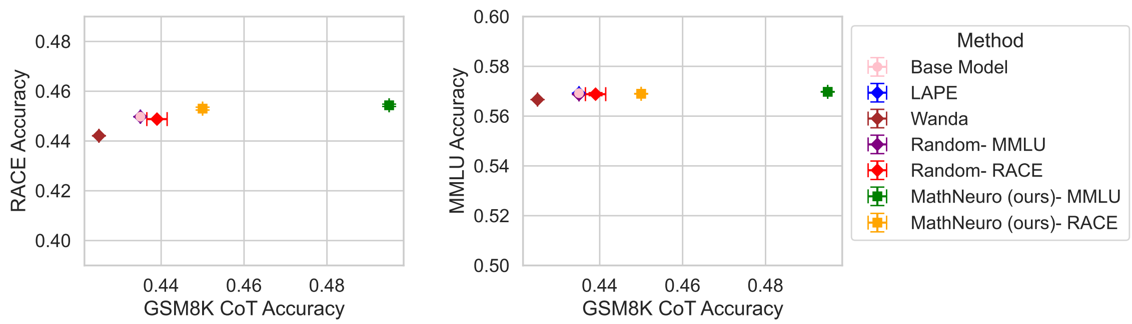

Figure 15: Effect of ***scaling*** identified parameters by 1.1 on math and non-math performance for Gemma 2 2B IT based on calculating the top 5% of parameters. Ideal methods should fall in the top right of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$. Horizontal and vertical lines represent 95% confidence intervals for each point on the plot.

3.2 3B IT), 1.075 worked best or tied with 1.05, while for Llama 3.1 8B IT, a larger model, 1.01 worked best. For the smaller models, we next tried scale factors between 1.075 and 1.05 (1.0625) and between 1.075 and the maximum scale factor we saw improved results based on initial experiments (1.1), finding that 1.1 worked best for all models except for Llama 3.2 1B IT, where 1.1 tied with the results of 1.075. For GSM8K for Llama 3.1 8B IT, we next tried a scale factor between 1.05 and the minimum scale factor we used (1.025), finding that 1.01 still worked best. Each scale factor considered increased performance across models for GSM8K.

For MATH, for small models, 1.01 or 1.05 worked best, while for Llama 3.1 8B IT, 1.01 worked best. For the smaller models, we next tried a scale factor between 1.01 and 1.05 (1.025), finding that 1.025 worked best for Llama 3.2 1B IT, Phi 1.5, and Llama 3.2 3B IT, and that 1.05 worked best for Gemma 2 2B IT. For MATH, scale factors beyond 1.05 either did not improve performance or harmed performance, suggesting a smaller scale factor is more optimal for this task. However, scale factors between 1.01 and 1.05 improved MATH performance across models. The results of this grid search for Llama 3.2 1B for GSM8K are displayed in Figure 51.

I Number and Location of Math-specific Parameters Using MMLU as $\mathcal{D}_{\text{non-math}}$

Figures 52, 53, and 54 show the consistency of math-specific parameters, percentage of top parameters that are math-specific, and distribution of math-specific parameters identified by MathNeuro using MMLU as the non-math dataset, respectively, based on the experiments described in Section 4.5.

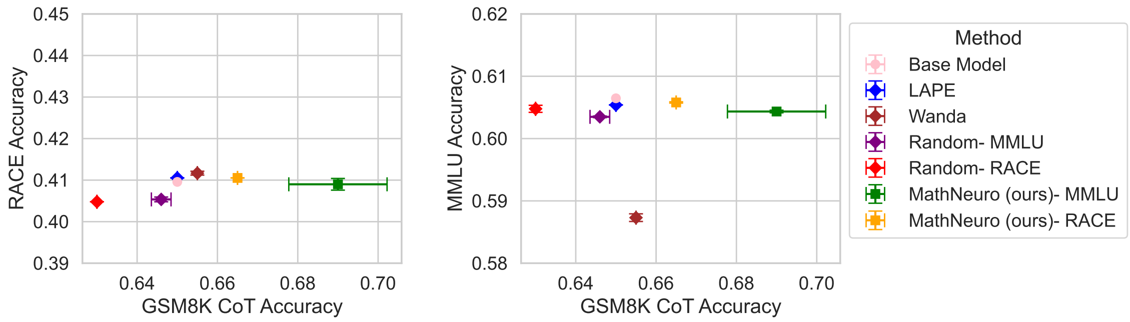

Figure 16: Effect of ***scaling*** identified parameters by 1.1 on math and non-math performance for Llama 3.2 3B IT based on calculating the top 5% of parameters. Ideal methods should fall in the top right of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$. Horizontal and vertical lines represent 95% confidence intervals for each point on the plot.

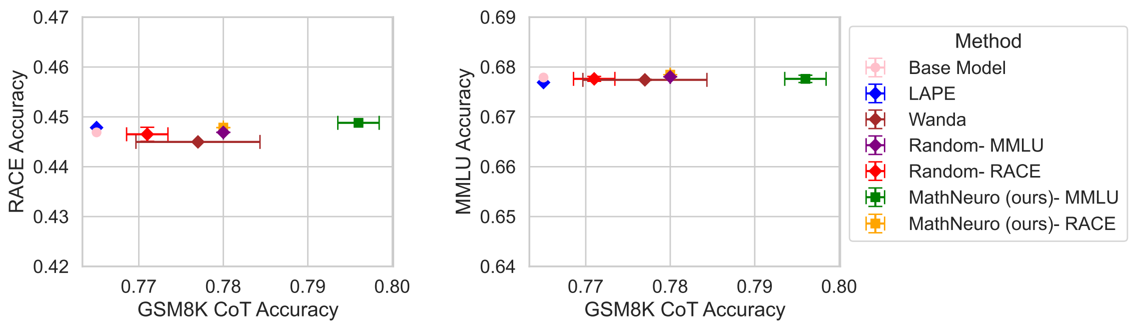

Figure 17: Effect of ***scaling*** identified parameters by 1.01 on math and non-math performance for Llama 3.1 8B IT based on calculating the top .5% of parameters. Ideal methods should fall in the top right of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$. Horizontal and vertical lines represent 95% confidence intervals for each point on the plot.

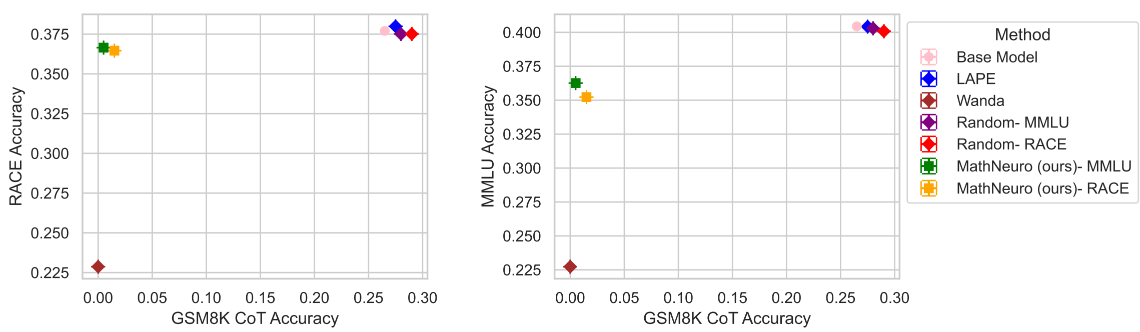

Figure 18: Effect of ***pruning*** identified parameters on math and non-math performance for Phi 1.5 for calculating the top .5% of parameters ***based on one sample***. Ideal methods should fall in the top left of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$. Horizontal and vertical lines represent 95% confidence intervals for each point on the plot.

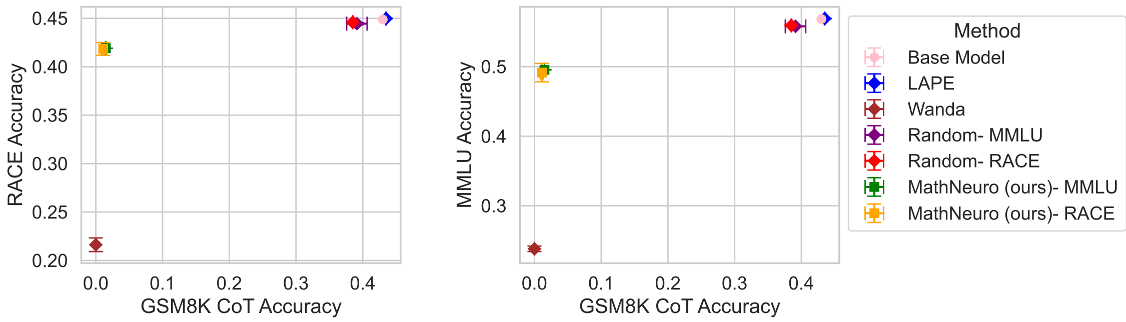

Figure 19: Effect of ***pruning*** identified parameters on math and non-math performance for Gemma 2 2B IT for calculating the top 2.5% of parameters ***based on one sample***. Ideal methods should fall in the top left of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$. Horizontal and vertical lines represent 95% confidence intervals for each point on the plot.

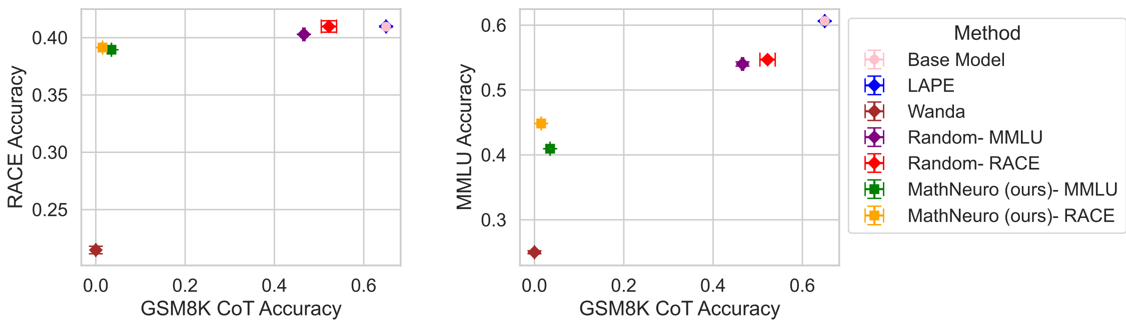

Figure 20: Effect of ***pruning*** identified parameters on math and non-math performance for Llama 3.2 3B IT for calculating the top 10% of parameters ***based on one sample***. Ideal methods should fall in the top left of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$. Horizontal and vertical lines represent 95% confidence intervals for each point on the plot.

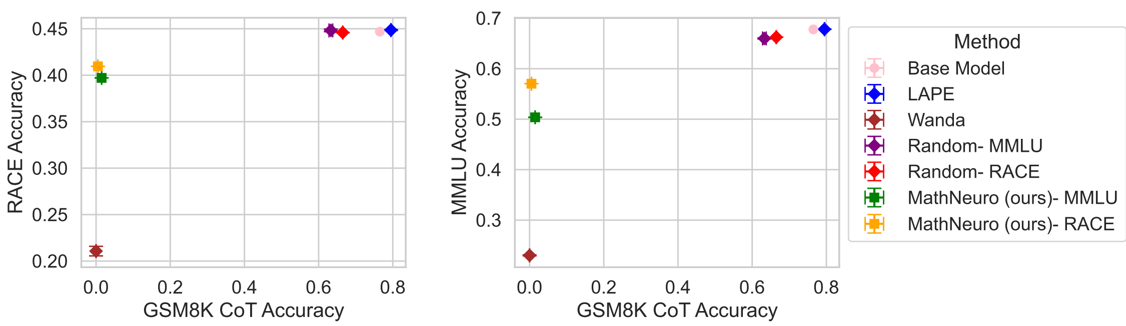

Figure 21: Effect of ***pruning*** identified parameters on math and non-math performance for Llama 3.1 8B IT for calculating the top 5% of parameters ***based on one sample***. Ideal methods should fall in the top left of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$. Horizontal and vertical lines represent 95% confidence intervals for each point on the plot.

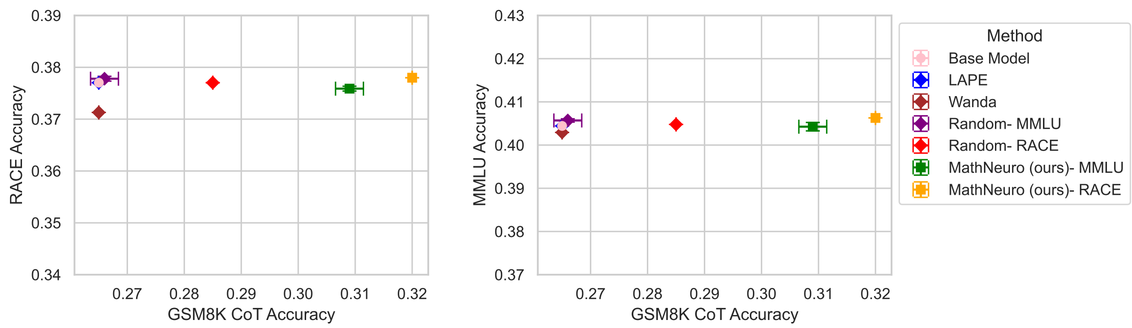

Figure 22: Effect of ***scaling*** identified parameters by 1.1 on math and non-math performance for Phi 1.5 for calculating the top .1% of parameters ***based on one sample***. Ideal methods should fall in the top right of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$. Horizontal and vertical lines represent 95% confidence intervals for each point on the plot.

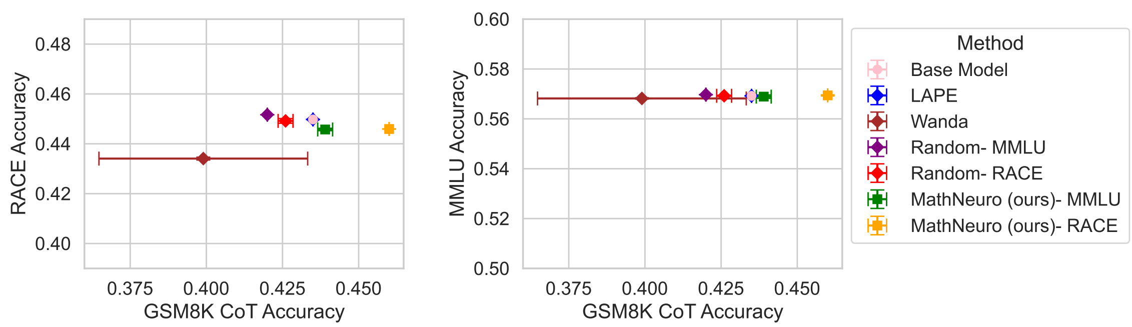

Figure 23: Effect of ***scaling*** identified parameters by 1.1 on math and non-math performance for Gemma 2 2B IT for calculating the top 2.5% of parameters ***based on one sample***. Ideal methods should fall in the top right of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$. Horizontal and vertical lines represent 95% confidence intervals for each point on the plot.

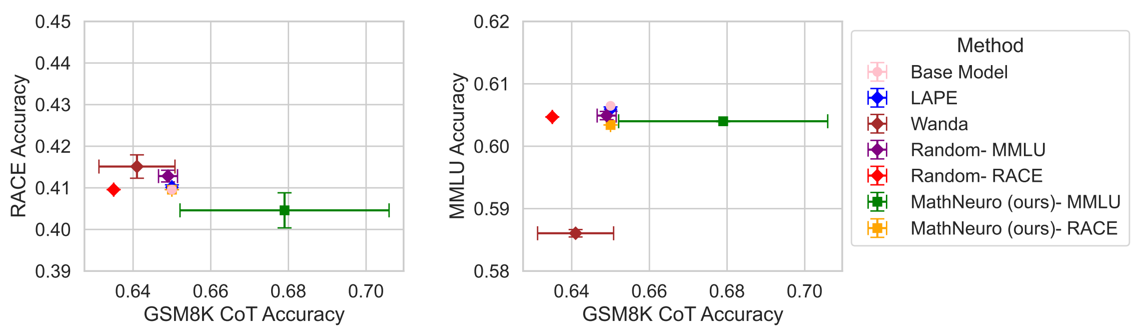

Figure 24: Effect of ***scaling*** identified parameters by 1.1 on math and non-math performance for Llama 3.2 3B IT for calculating the top 5% of parameters ***based on one sample***. Ideal methods should fall in the top right of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$. Horizontal and vertical lines represent 95% confidence intervals for each point on the plot.

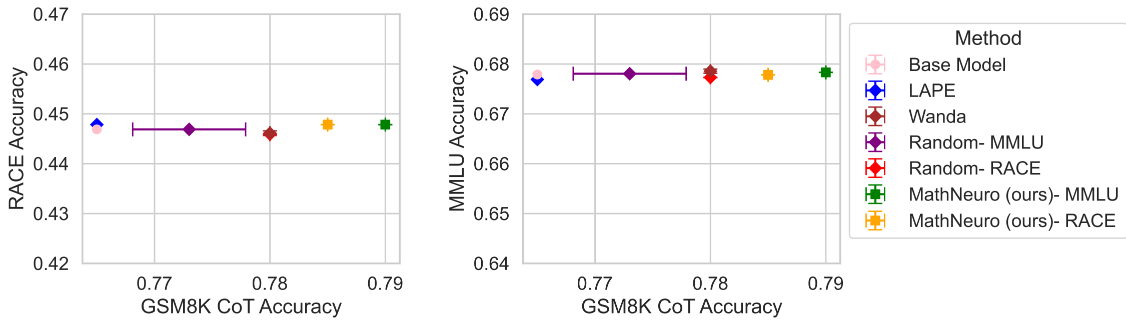

Figure 25: Effect of ***scaling*** identified parameters by 1.01 on math and non-math performance for Llama 3.1 8B IT for calculating the top 1% of parameters ***based on one sample***. Ideal methods should fall in the top right of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$. Horizontal and vertical lines represent 95% confidence intervals for each point on the plot.

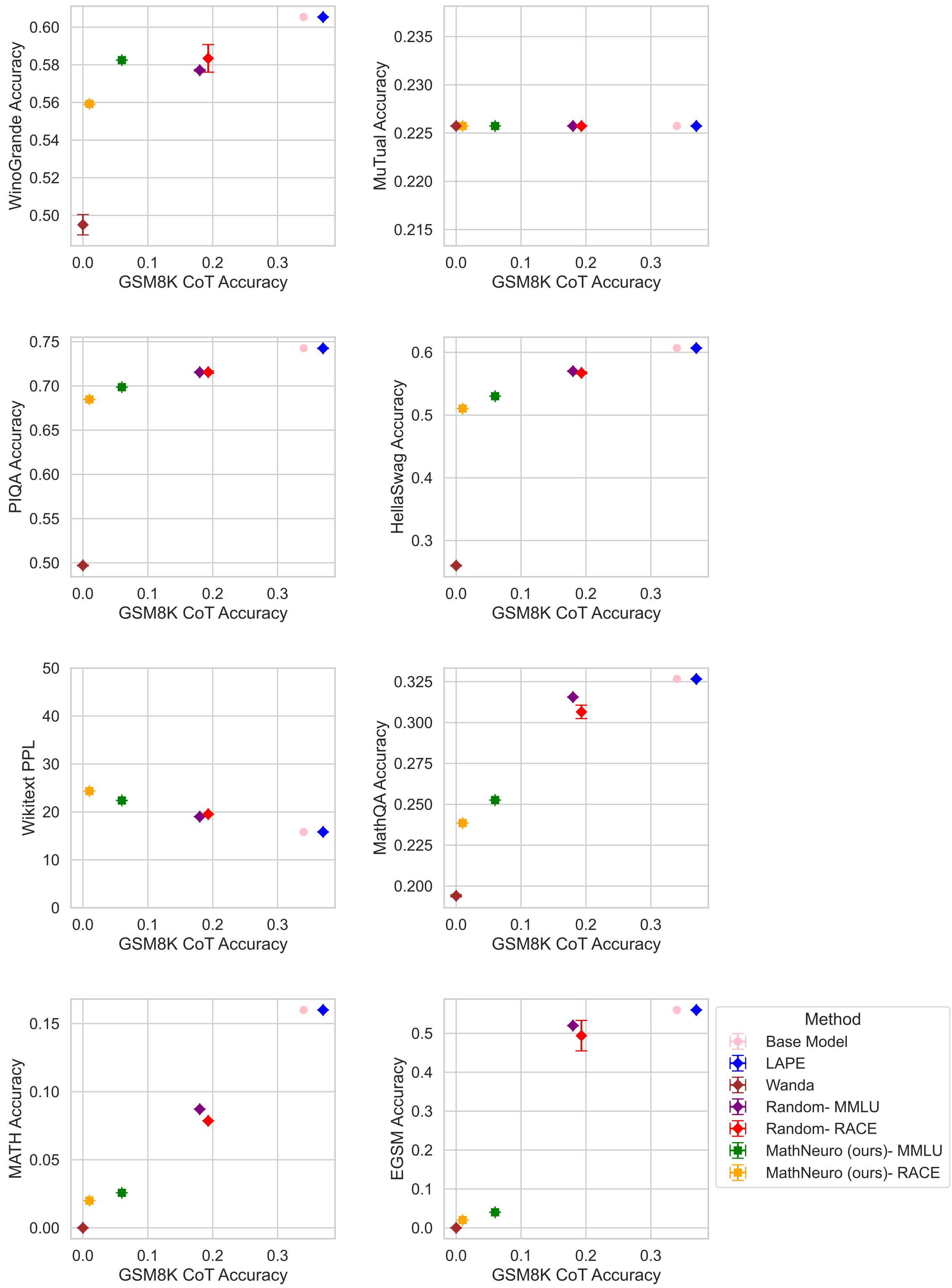

Figure 26: Effect of ***pruning*** identified parameters on performance for unseen math and non-math tasks for Llama 3.2 1B IT when using GSM8K as $\mathcal{D}_{\text{math}}$ and MMLU or Race as $\mathcal{D}_{\text{non-math}}$. Ideal methods for the first two rows of figures should fall in the top left of the plot, while ideal methods for the last two rows of figures should fall in the bottom left of the plot. Wanda results are not pictured in the Wikitext figures because PPL increased dramatically when using this method. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$. Horizontal and vertical lines represent 95% confidence intervals for each point on the plot.

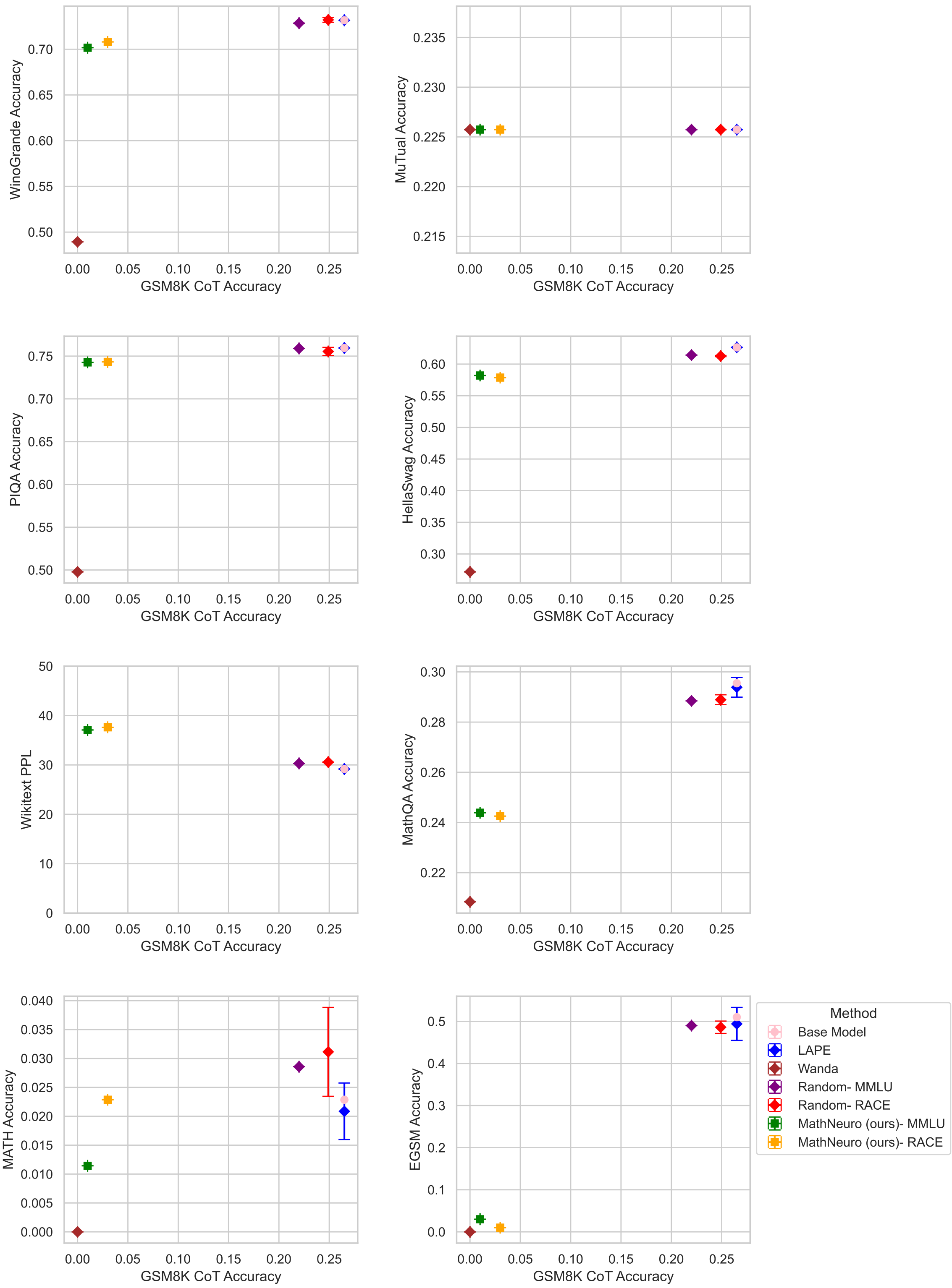

Figure 27: Effect of ***pruning*** identified parameters on performance for unseen math and non-math tasks for Phi 1.5 when using GSM8K as $\mathcal{D}_{\text{math}}$ and MMLU or Race as $\mathcal{D}_{\text{non-math}}$. Ideal methods for the first two rows of figures should fall in the top left of the plot, while ideal methods for the last two rows of figures should fall in the bottom left of the plot. Wanda results are not pictured in the Wikitext figures because PPL increased dramatically when using this method. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$. Horizontal and vertical lines represent 95% confidence intervals for each point on the plot.

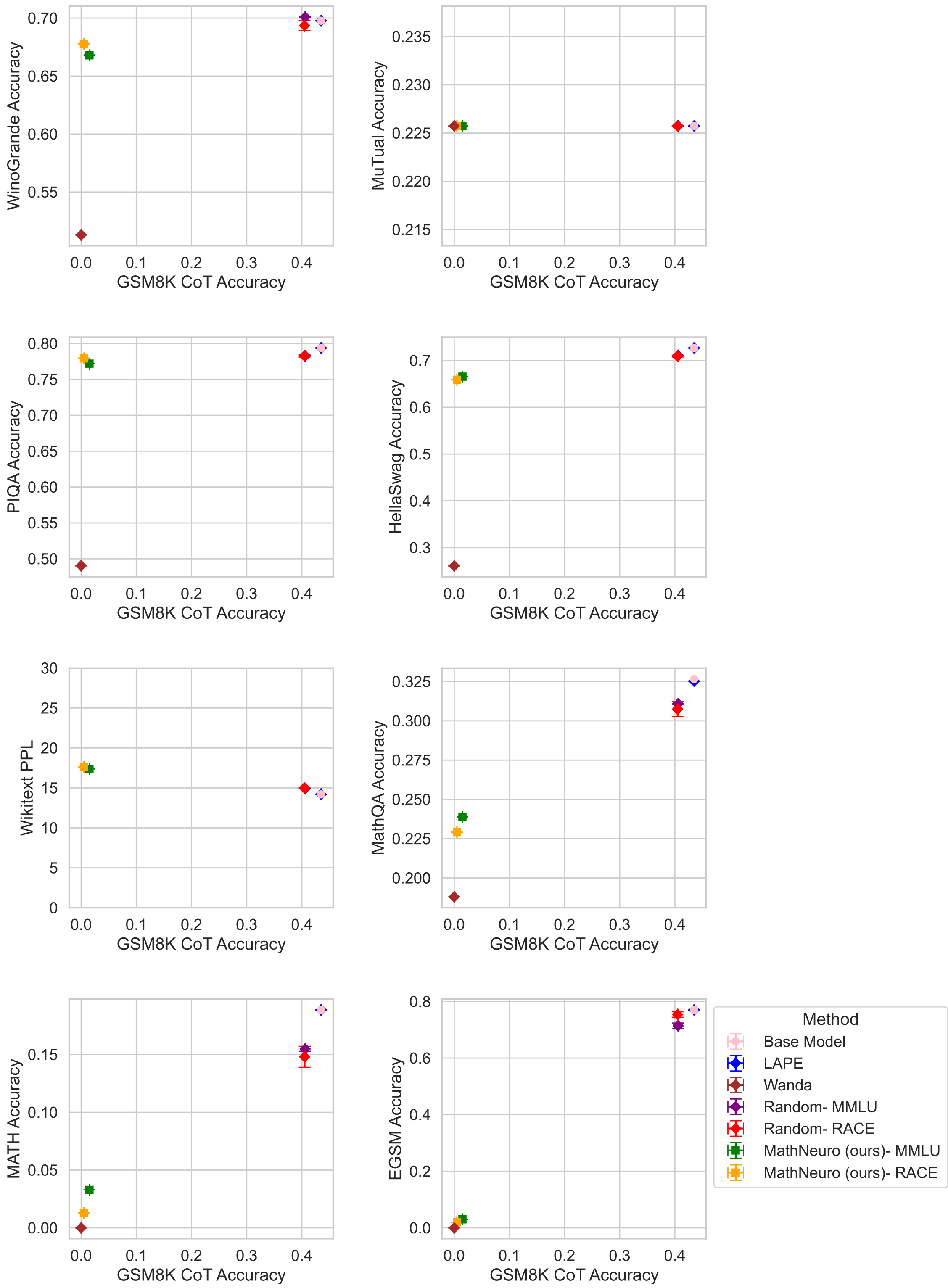

Figure 28: Effect of ***pruning*** identified parameters on performance for unseen math and non-math tasks for Gemma 2 2B IT when using GSM8K as $\mathcal{D}_{\text{math}}$ and MMLU or Race as $\mathcal{D}_{\text{non-math}}$. Ideal methods for the first two rows of figures should fall in the top left of the plot, while ideal methods for the last two rows of figures should fall in the bottom left of the plot. Wanda results are not pictured in the Wikitext figures because PPL increased dramatically when using this method. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$. Horizontal and vertical lines represent 95% confidence intervals for each point on the plot.

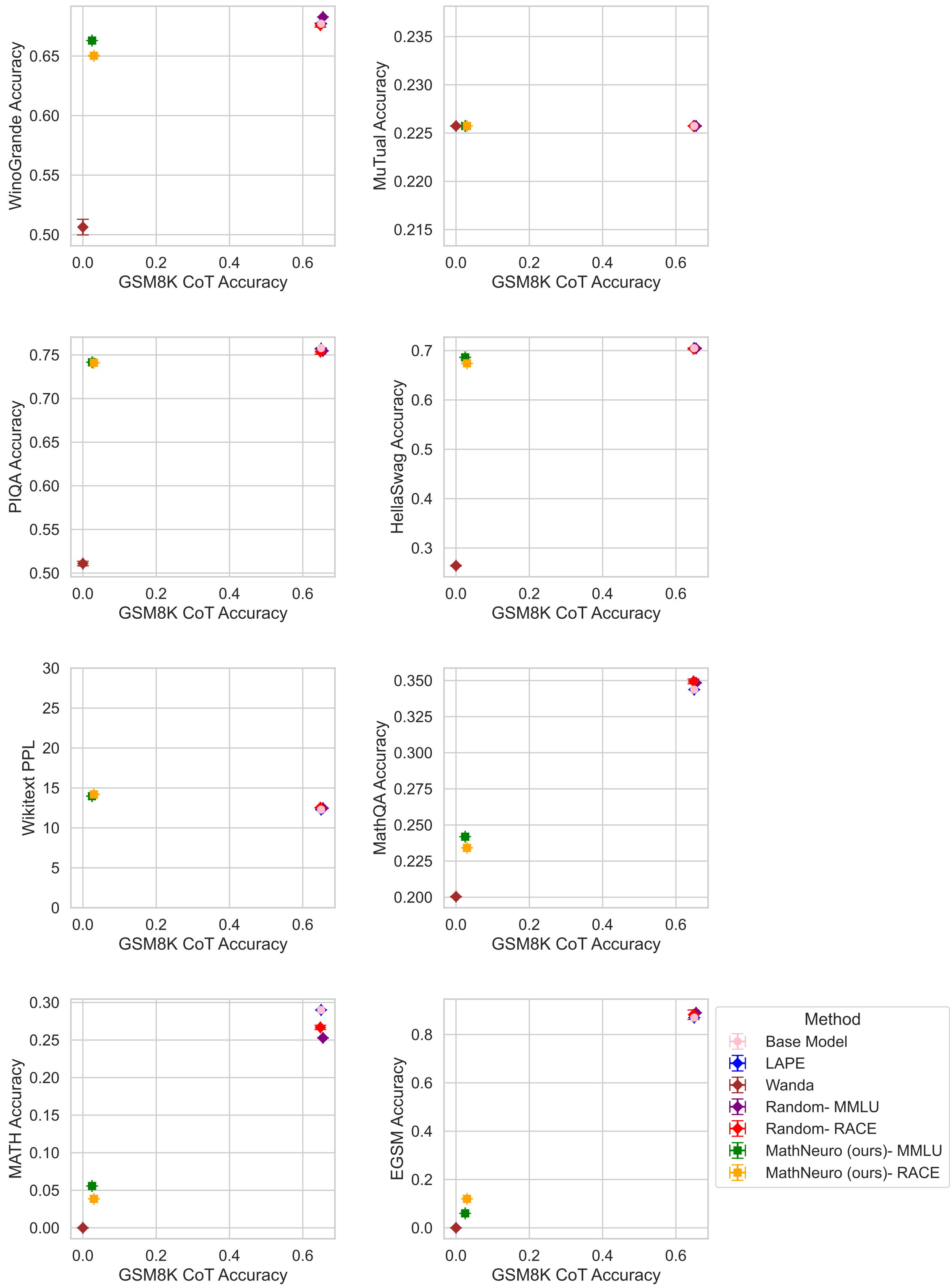

Figure 29: Effect of ***pruning*** identified parameters on performance for unseen math and non-math tasks for Llama 3.2 3B IT when using GSM8K as $\mathcal{D}_{\text{math}}$ and MMLU or Race as $\mathcal{D}_{\text{non-math}}$. Ideal methods for the first two rows of figures should fall in the top left of the plot, while ideal methods for the last two rows of figures should fall in the bottom left of the plot. Wanda results are not pictured in the Wikitext figures because PPL increased dramatically when using this method. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$. Horizontal and vertical lines represent 95% confidence intervals for each point on the plot.

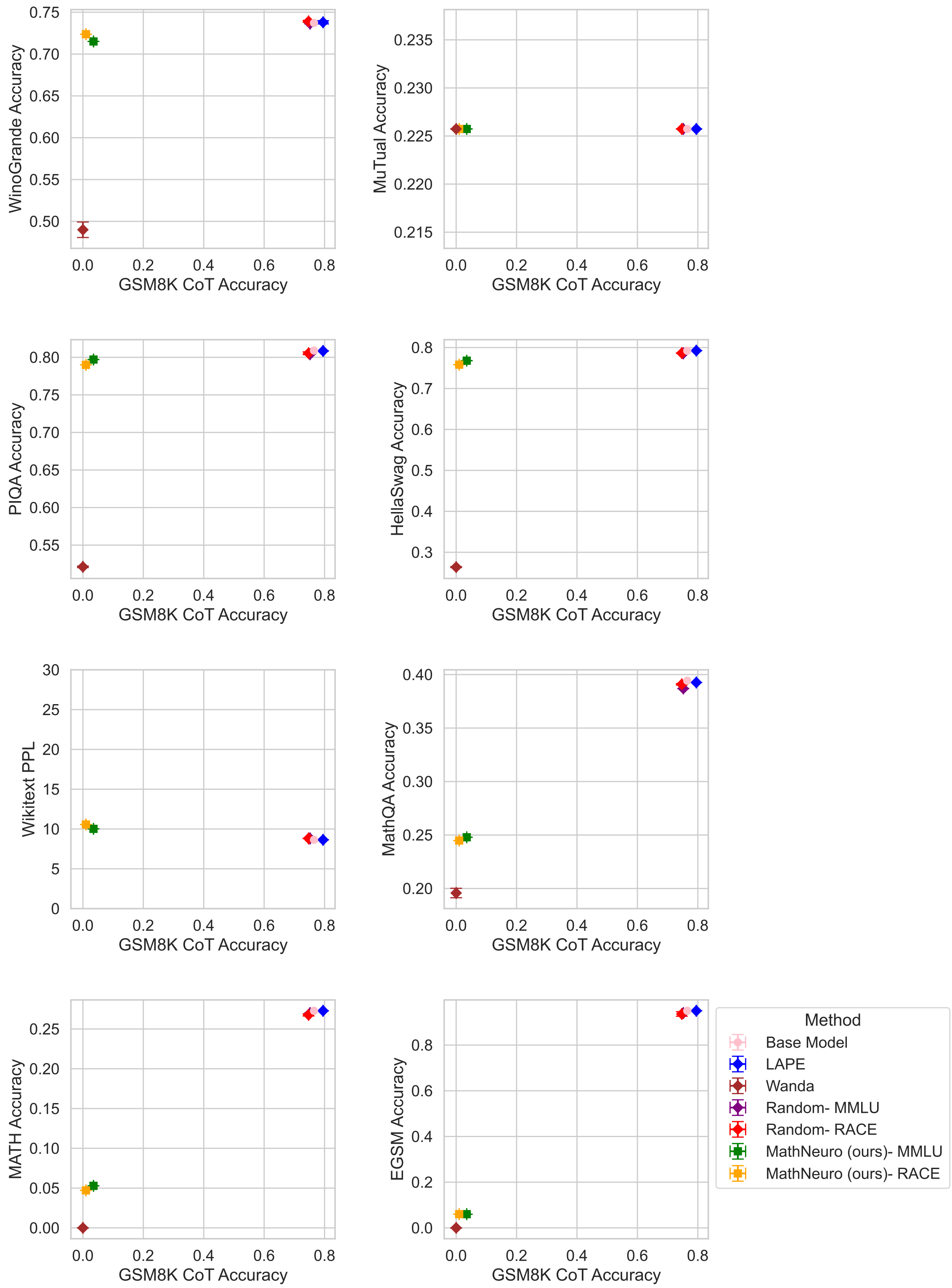

Figure 30: Effect of ***pruning*** identified parameters on performance for unseen math and non-math tasks for Llama 3.1 8B IT when using GSM8K as $\mathcal{D}_{\text{math}}$ and MMLU or Race as $\mathcal{D}_{\text{non-math}}$. Ideal methods for the first two rows of figures should fall in the top left of the plot, while ideal methods for the last two rows of figures should fall in the bottom left of the plot. Wanda results are not pictured in the Wikitext figures because PPL increased dramatically when using this method. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$. Horizontal and vertical lines represent 95% confidence intervals for each point on the plot.

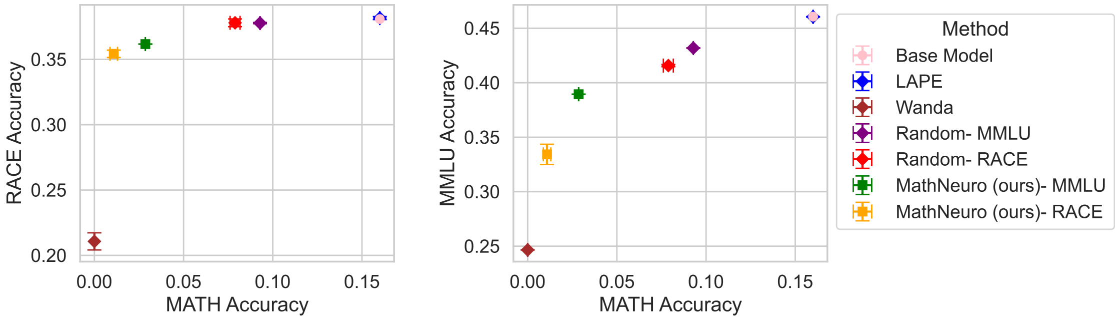

Figure 31: Effect of ***pruning*** identified parameters on math and non-math performance for Llama 3.2 1B IT based on calculating the top 15% of parameters using the MATH dataset as $\mathcal{D}_{\text{math}}$. Ideal methods should fall in the top left of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$. Horizontal and vertical lines represent 95% confidence intervals for each point on the plot.

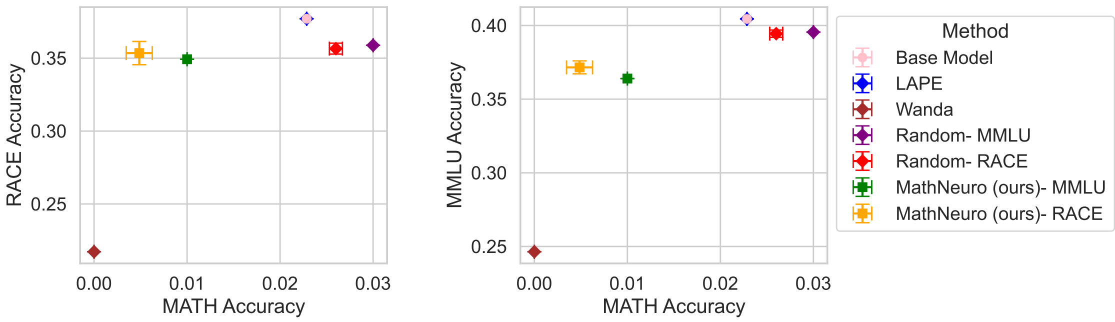

Figure 32: Effect of ***pruning*** identified parameters on math and non-math performance for Phi 1.5 based on calculating the top 5% of parameters using the MATH dataset as $\mathcal{D}_{\text{math}}$. Ideal methods should fall in the top left of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$. Horizontal and vertical lines represent 95% confidence intervals for each point on the plot.

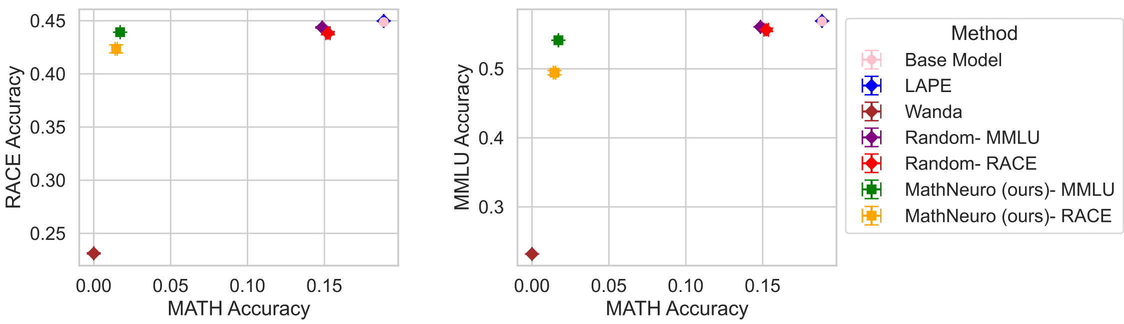

Figure 33: Effect of ***pruning*** identified parameters on math and non-math performance for Gemma 2 2B IT based on calculating the top 5% of parameters using the MATH dataset as $\mathcal{D}_{\text{math}}$. Ideal methods should fall in the top left of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$. Horizontal and vertical lines represent 95% confidence intervals for each point on the plot.

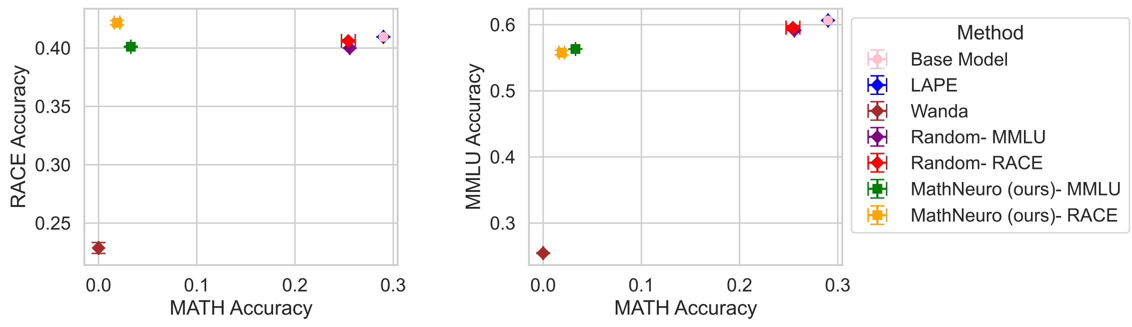

Figure 34: Effect of ***pruning*** identified parameters on math and non-math performance for Llama 3.2 3B IT based on calculating the top 2.5% (left) and 1% (right) of parameters using the MATH dataset as $\mathcal{D}_{\text{math}}$. Ideal methods should fall in the top left of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$. Horizontal and vertical lines represent 95% confidence intervals for each point on the plot.

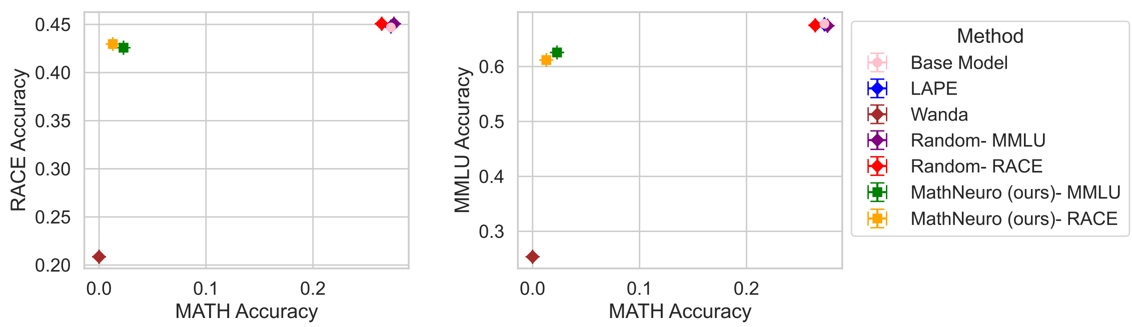

Figure 35: Effect of ***pruning*** identified parameters on math and non-math performance for Llama 3.1 8B IT based on calculating the top 1% of parameters using the MATH dataset as $\mathcal{D}_{\text{math}}$. Ideal methods should fall in the top left of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$. Horizontal and vertical lines represent 95% confidence intervals for each point on the plot.

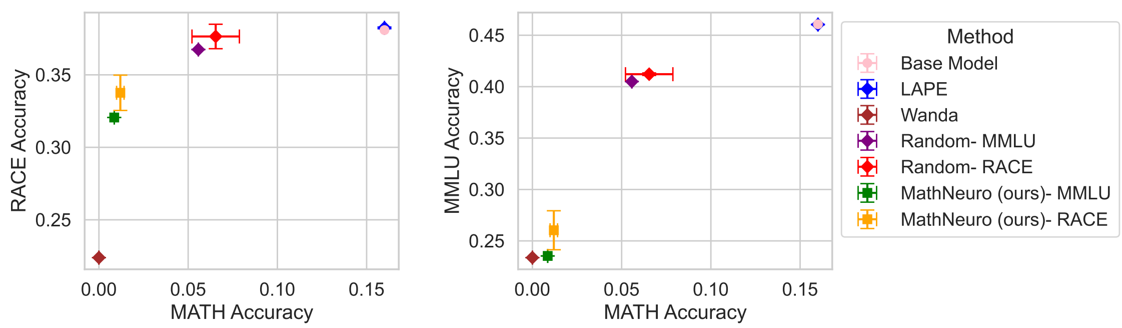

Figure 36: Effect of ***pruning*** identified parameters on math and non-math performance for Llama 3.2 1B IT based on calculating the top 15% of parameters using the MATH dataset as $\mathcal{D}_{\text{math}}$ **based on one sample**. Ideal methods should fall in the top left of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$. Horizontal and vertical lines represent 95% confidence intervals for each point on the plot.

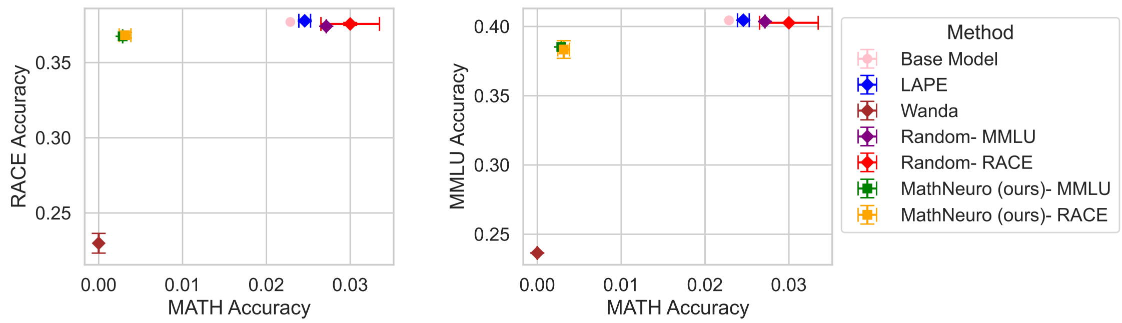

Figure 37: Effect of ***pruning*** identified parameters on math and non-math performance for Phi 1.5 based on calculating the top 5% of parameters using the MATH dataset as $\mathcal{D}_{\text{math}}$ **based on one sample**. Ideal methods should fall in the top left of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$. Horizontal and vertical lines represent 95% confidence intervals for each point on the plot.

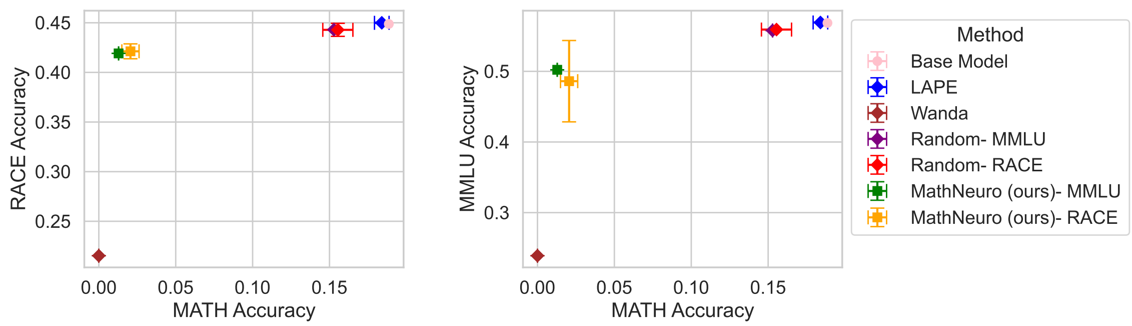

Figure 38: Effect of ***pruning*** identified parameters on math and non-math performance for Gemma 2 2B IT based on calculating the top 5% of parameters using the MATH dataset as $\mathcal{D}_{\text{math}}$ **based on one sample**. Ideal methods should fall in the top left of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$. Horizontal and vertical lines represent 95% confidence intervals for each point on the plot.

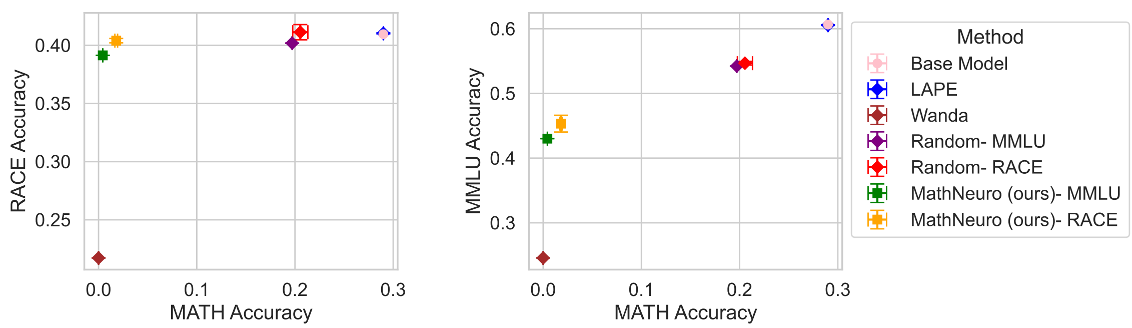

Figure 39: Effect of ***pruning*** identified parameters on math and non-math performance for Llama 3.2 3B IT based on calculating the top 2.5% (left) and 1% (right) of parameters using the MATH dataset as $\mathcal{D}_{\text{math}}$ **based on one sample**. Ideal methods should fall in the top left of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$. Horizontal and vertical lines represent 95% confidence intervals for each point on the plot.

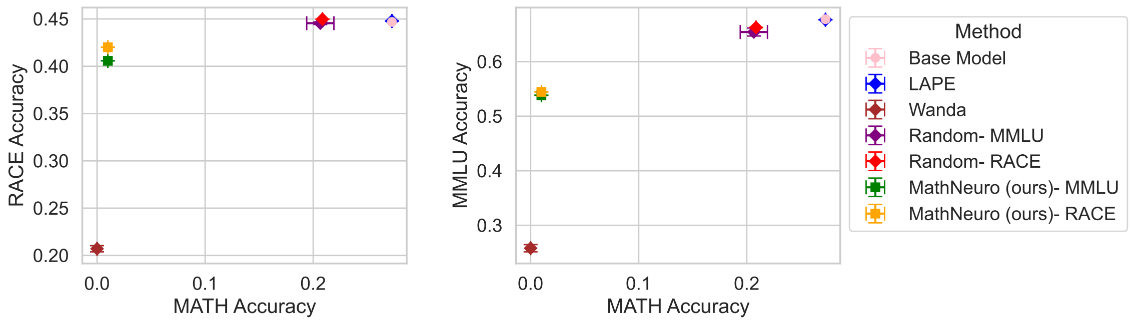

Figure 40: Effect of ***pruning*** identified parameters on math and non-math performance for Llama 3.1 8B IT based on calculating the top 1% of parameters using the MATH dataset as $\mathcal{D}_{\text{math}}$ **based on one sample**. Ideal methods should fall in the top left of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$. Horizontal and vertical lines represent 95% confidence intervals for each point on the plot.

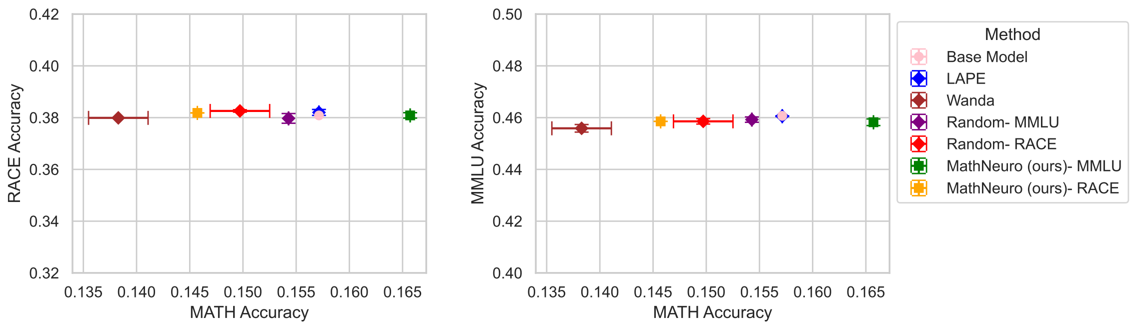

Figure 41: Effect of ***scaling*** identified parameters by 1.025 on math and non-math performance for Llama 3.2 1B IT based on calculating the top .5% of parameters using the MATH dataset as $\mathcal{D}_{\text{math}}$. Ideal methods should fall in the top right of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$. Horizontal and vertical lines represent 95% confidence intervals for each point on the plot.

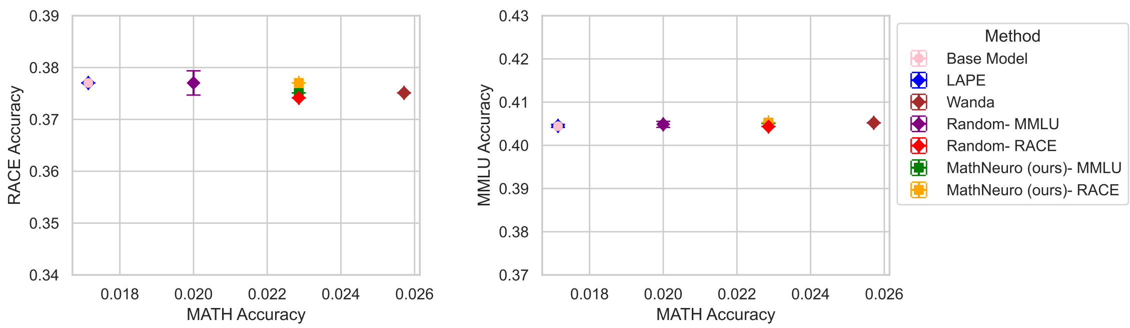

Figure 42: Effect of ***scaling*** identified parameters by 1.025 on math and non-math performance for Phi 1.5 based on calculating the top 15% of parameters using the MATH dataset as $\mathcal{D}_{\text{math}}$. Ideal methods should fall in the top right of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$. Horizontal and vertical lines represent 95% confidence intervals for each point on the plot.

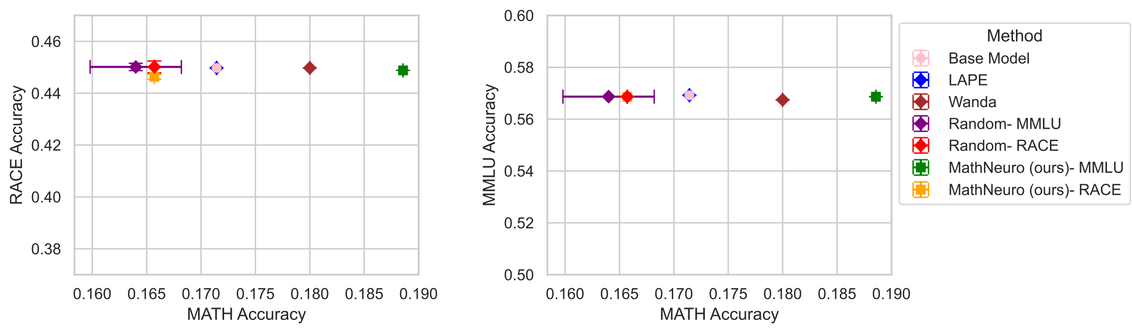

Figure 43: Effect of ***scaling*** identified parameters by 1.05 on math and non-math performance for Gemma 2 2B IT based on calculating the top .5% of parameters using the MATH dataset as $\mathcal{D}_{\text{math}}$. Ideal methods should fall in the top right of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$. Horizontal and vertical lines represent 95% confidence intervals for each point on the plot.

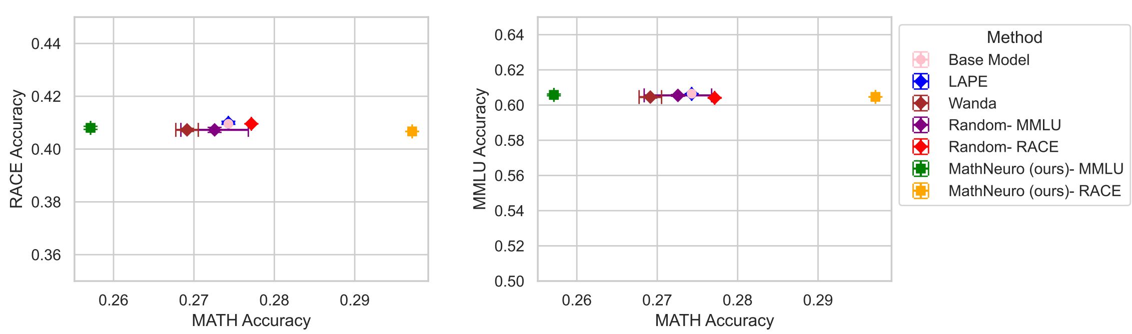

Figure 44: Effect of ***scaling*** identified parameters by 1.025 on math and non-math performance for Llama 3.2 3B IT based on calculating the top 10% of parameters using the MATH dataset as $\mathcal{D}_{\text{math}}$. Ideal methods should fall in the top right of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$. Horizontal and vertical lines represent 95% confidence intervals for each point on the plot.

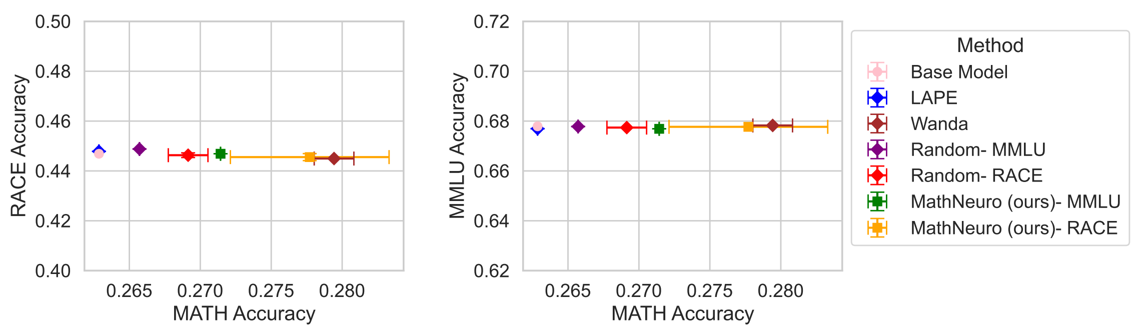

Figure 45: Effect of ***scaling*** identified parameters by 1.01 on math and non-math performance for Llama 3.1 8B IT based on calculating the top 2.5% of parameters using the MATH dataset as $\mathcal{D}_{\text{math}}$. Ideal methods should fall in the top right of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$. Horizontal and vertical lines represent 95% confidence intervals for each point on the plot.

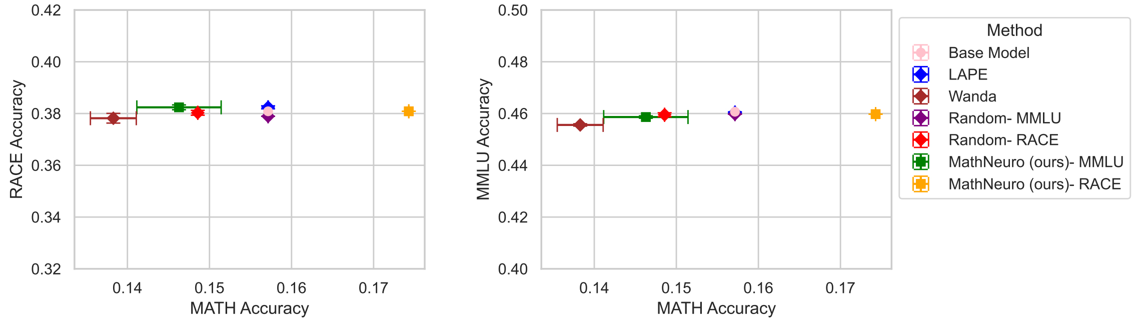

Figure 46: Effect of ***scaling*** identified parameters by 1.025 on math and non-math performance for Llama 3.2 1B IT based on calculating the top .5% of parameters using the MATH dataset as $\mathcal{D}_{\text{math}}$ **based on one sample**. Ideal methods should fall in the top right of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$. Horizontal and vertical lines represent 95% confidence intervals for each point on the plot.

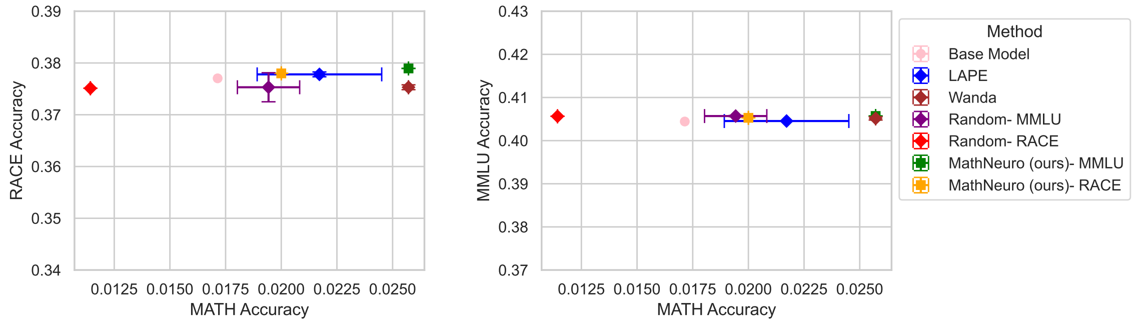

Figure 47: Effect of ***scaling*** identified parameters by 1.025 on math and non-math performance for Phi 1.5 based on calculating the top 10% of parameters using the MATH dataset as $\mathcal{D}_{\text{math}}$ **based on one sample**. Ideal methods should fall in the top right of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$. Horizontal and vertical lines represent 95% confidence intervals for each point on the plot.

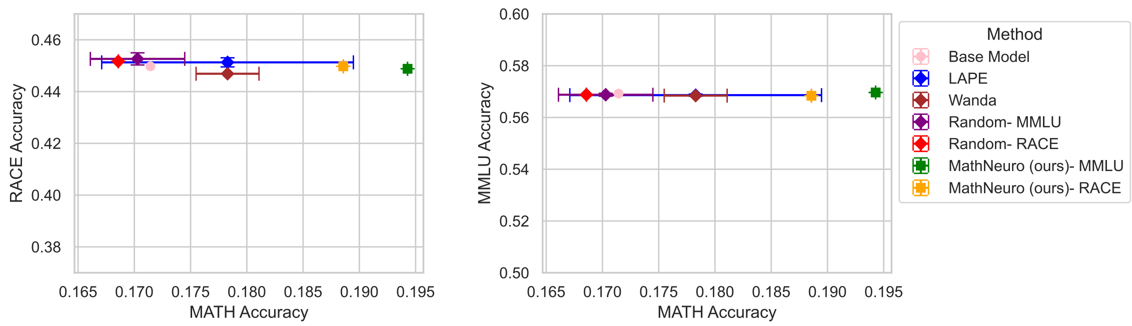

Figure 48: Effect of ***scaling*** identified parameters by 1.05 on math and non-math performance for Gemma 2 2B IT based on calculating the top 1% of parameters using the MATH dataset as $\mathcal{D}_{\text{math}}$ **based on one sample**. Ideal methods should fall in the top right of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$. Horizontal and vertical lines represent 95% confidence intervals for each point on the plot.

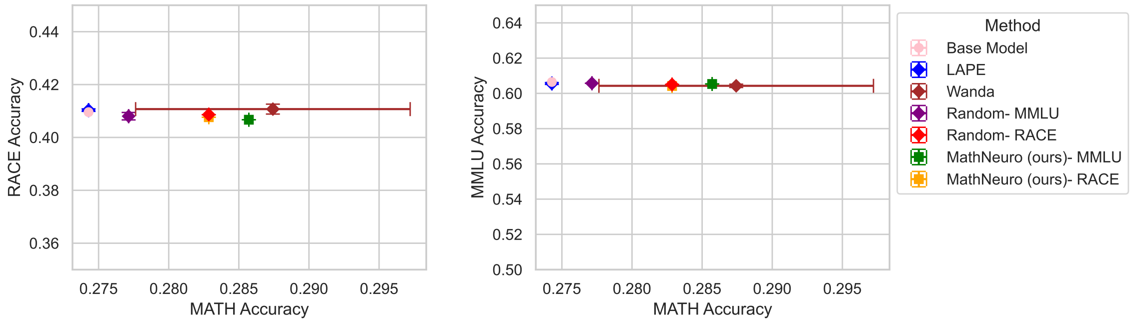

Figure 49: Effect of ***scaling*** identified parameters by 1.025 on math and non-math performance for Llama 3.2 3B IT based on calculating the top .01% of parameters using the MATH dataset as $\mathcal{D}_{\text{math}}$ **based on one sample**. Ideal methods should fall in the top right of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$. Horizontal and vertical lines represent 95% confidence intervals for each point on the plot.

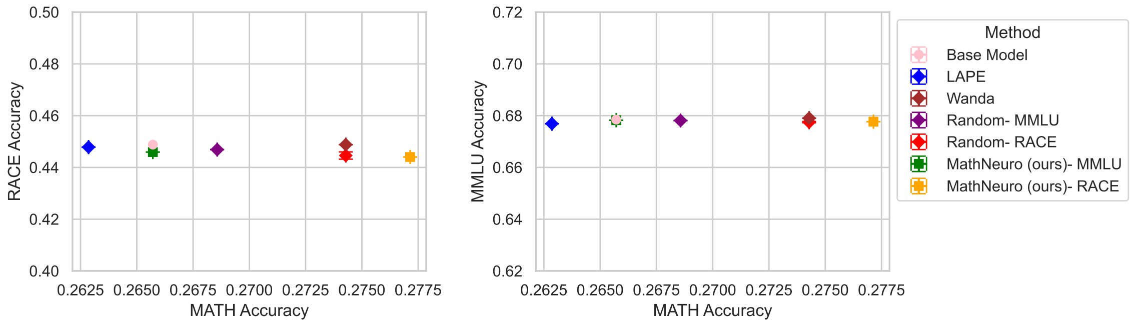

Figure 50: Effect of ***scaling*** identified parameters by 1.01 on math and non-math performance for Llama 3.1 8B IT based on calculating the top 10% of parameters using the MATH dataset as $\mathcal{D}_{\text{math}}$ **based on one sample**. Ideal methods should fall in the top right of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$. Horizontal and vertical lines represent 95% confidence intervals for each point on the plot.

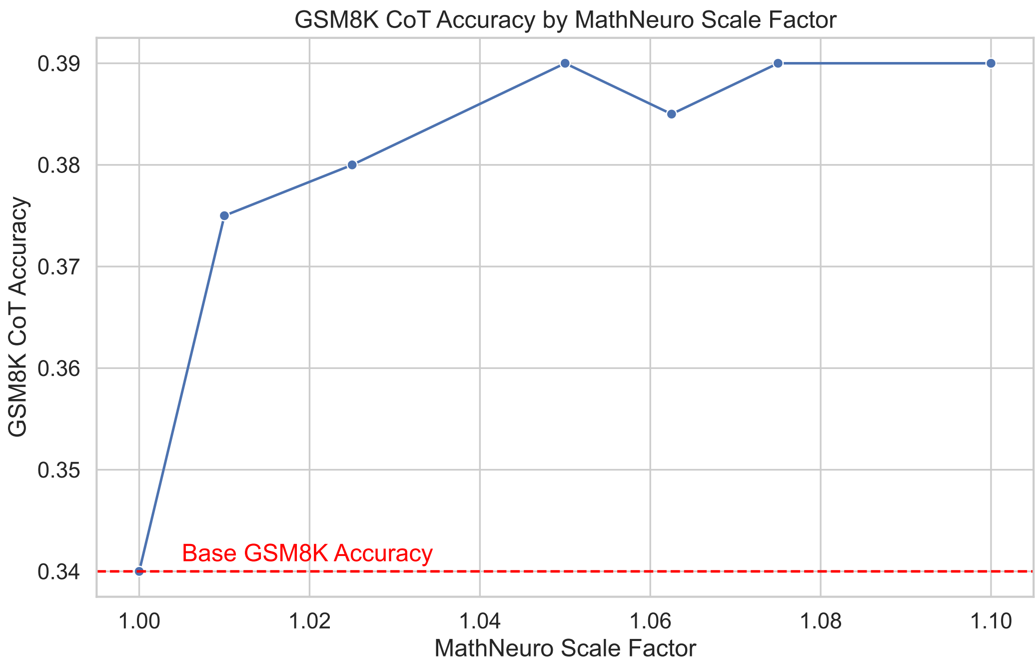

Figure 51: Impact of MathNeuro scale factor on GSM8K performance for Llama 3.2 1B IT.

Question Betty is saving money for a new wallet which costs $100. Betty has only half of the money she needs. Her parents decided to give her $15 for that purpose, and her grandparents twice as much as her parents. How much more money does Betty need to buy the wallet?
Response Before Pruning (correct) Let's think step by step. Betty has half of the money she needs, which is 100/2 =50. She received $15 from her parents, so now she has $50 + $15 = $65. Her grandparents gave her twice as much as her parents, which is 2 x $15 = $30. Now she has $65 + $30 = $95. The wallet costs $100, so Betty needs $100 - $95 = $5 more to buy the wallet.
Response After Pruning Based on RACE as $\mathcal{D}_{\textbf{non-math}}$ (incorrect) Let's think step by step. First, let's determine the total amount needed to buy the wallet. $100 is the total amount needed to buy the wallet. Betty's parents gave her $15. Her grandparents gave her $2 as a gift. Betty's grandparents gave her $15 as a gift. Betty's parents gave her $15 as a gift. Betty's grandparents gave her $15 as a gift. Betty's parents gave her $15 as a gift. Betty's grandparents gave her $15 as a gift. Betty's grandparents gave her $15 as a gift. Betty's grandparents gave her $15 as a gift. Betty's grandparents gave her $15 as a gift. Betty's grandparents gave her $15 as a gift. Betty's grandparents gave her $15
Response After Pruning Based on MMLU as $\mathcal{D}_{\textbf{non-math}}$ (incorrect) A: Let's think step by step. Betty needs $100 to buy the wallet. Her parents gave her $15 for that purpose, and her grandparents gave her $15 twice as much as her parents. This is a 50-50 split. Betty's grandparents gave her $15 twice as much as her parents, which is $15. Therefore, Betty's grandparents gave her $15 twice as much as her parents. B: Let's think step by step. Betty's grandparents gave her $15 twice as much as her parents. This is a 50-50 split. Betty's grandparents gave her $15 twice as much as her parents. Therefore, Betty's grandparents gave her $15 twice as much as her parents. C: Let's think step by step. Betty's

Table 1: Responses to a GSM8K question before and after pruning parameters identified by MathNeuro for Llama 3 1B IT.

Question The profit from a business transaction is shared among 2 business partners, Mike and Johnson in the ratio 2:5 respectively. If Johnson got $2500, how much will Mike have after spending some of his share on a shirt that costs $200?
Response Before Scaling (incorrect) Let's think step by step. Mike's share of the profit is 2 parts, and Johnson's share is 5 parts. The total number of parts is 2 + 5 = 7. Johnson's share of the profit is 5/7 of the total profit. If Johnson got $2500, then Mike's share of the profit is 2/7 of the total profit. If Mike spent $200 on a shirt, then he will have $200 + (2/7) of the profit. The profit is 7/7 of the total profit, so Mike's share of the profit is 2/7 of 7/7, which is 2/7. Mike's share of the profit is $200 + (2/7) of the profit, which is $200 + (2/7)(7/7) = $200 + 2 = $202. The answer is $202.
Response After Scaling Based on RACE as $\mathcal{D}_{\textbf{non-math}}$ (incorrect) Let's think step by step. Mike's share of the profit is 2 parts, and Johnson's share is 5 parts. The total number of parts is 2 + 5 = 7. Johnson's share is 5/7 of the total profit. Therefore, Mike's share is 2/7 of the total profit. Mike's share is 2/7 × $2500 = $142.86. Mike will have $142.86 after spending $200 on a shirt.
Response After Scaling Based on MMLU as $\mathcal{D}_{\textbf{non-math}}$ (correct) Let's think step by step. Mike's share of the profit is 2 parts, and Johnson's share is 5 parts. The total number of parts is 2 + 5 = 7. Johnson's share is 5/7 of the total profit. If Johnson got $2500, then Mike's share is 2/7 of the total profit. If Mike spent $200 on a shirt, how much profit will he have left? To find Mike's profit, we need to find the total profit first. We can set up the following proportion: 2/7 = 2500/$x, where x is the total profit. Cross-multiplying, we get 2x = 2500(7). Dividing both sides by 2, we get x = 3500. Now that we know the total profit, we can find Mike's profit: Mike's share is 2/7 of the total profit, so Mike's profit is 2/7(3500) = 1000. Finally, we can find out how much profit Mike will have left after spending $200 on a shirt: Mike's profit - $200 = $800. The answer is $800.

Table 2: Responses to a GSM8K question before and after scaling parameters identified by MathNeuro for Llama 3 1B IT.

Prompt

Read this passage and answer the multiple choice question below it.
A newspaper reporter's job can be very interesting. He meets all types of people and lives quite a busy life. He is for news all the time,then after several years he may get a desk job,and life becomes a bit more settled. Let's look at his work a little more closely. In a day he may have to interview the prime minister of a foreign country, and the next day he may be writing about a football match. Sometimes he may be so busy that he hardly has any time to sleep. And at other times he may go on for days looking out for news materials yet return empty–handed.
In the beginning, a reporter has to cover a very wide field. After the early years he becomes more specialized in his work. For example, he may finaliy be asked to write only on court cases or politics or sports. Some reporters may become so specialized that they are asked only to write on a special thing: horse racing, for example. In most newspaper houses there is at least one special racing correspondent. Some newspapers have book reviews. Their job is delightful. They read the latest book and then write reviews on the ones they like. Then there are those who write on fdms, so they get to see them even before they are shown in the cinema. How lucky, you would say! A reporter's job can also be very dangerous. If there is a flood or a riot they may get hurt or even be killed. Three years ago there was a reporter whose camera was destroyed by a group of men, because they were angry with him for taking their picture. Dangerous or not, one thing is certain, and that is, their job is never dull!
Question: Reporters who write on films are said to be lucky because they [blank].
Answer choices: ['can write anything they like', 'can see more film stars', 'can pay less than other people', 'can see the fdms before most people see them in the cinema']

Response Before Interventions (incorrect)

Answer: 'can see more film stars'

Response After Pruning Based on RACE as $\mathcal{D}_{\text{non-math}}$ (incorrect)

Answer: 'can see more film stars'

Response After Pruning Based on MMLU as $\mathcal{D}_{\text{non-math}}$ (incorrect)

Answer: A. 'can write anything they like'

Response After Scaling Based on RACE as $\mathcal{D}_{\text{non-math}}$ (incorrect)

Answer: A

Response After Scaling Based on MMLU as $\mathcal{D}_{\text{non-math}}$ (incorrect)

Answer: A

Table 3: Responses to a RACE question before and after pruning or scaling parameters identified by MathNeuro for Llama 3 1B IT.

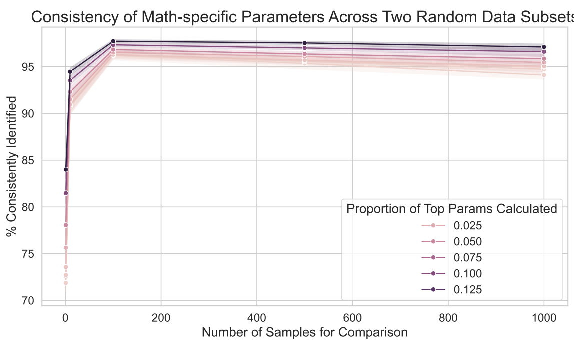

Figure 52: Consistency of math-specific parameters identified by MathNeuro for Llama 3.2 1B IT when identifying using GSM8K compared to MMLU.

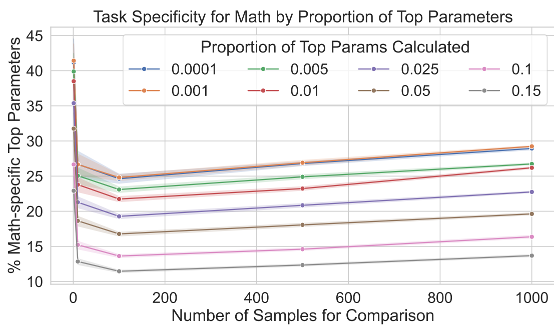

Figure 53: Percentage of top parameters that are math-specific as identified by MathNeuro for Llama 3.2 1B IT when identifying using GSM8K compared to MMLU.

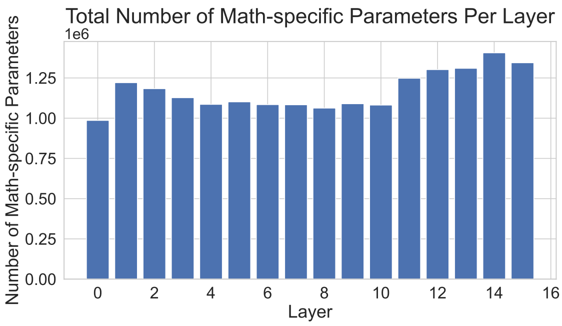

Figure 54: Distribution of math-specific parameters identified by MathNeuro for Llama 3.2 1B IT when identifying using GSM8K compared to MMLU.

Prompt
Question: A 30-year-old nulliparous female presents to the office with the complaint of mood changes. She says that for the past several months she has been anxious, hyperactive, and unable to sleep 3 to 4 days prior to the onset of menses. She further reports that on the day her menses begins she becomes acutely depressed, anorectic, irritable, and lethargic. She has no psychiatric history. Physical examination findings are normal. She and her husband have been trying to conceive for over 2 years. History reveals a tuboplasty approximately 1 year ago to correct a closed fallopian tube. The most likely diagnosis is Answer choices: ['adjustment disorder with depressed mood', 'bipolar I disorder, mixed', 'cyclothymic personality', 'generalized anxiety disorder']
Response Before Pruning (correct; hit max number of output tokens) Answer: A Explanation: The patient's symptoms of mood changes, anxiety, hyperactivity, inability to sleep, acute depression, anorexia, irritability, and lethargy, particularly in the context of her history of tuboplasty, are characteristic of adjustment disorder with depressed mood. The patient's anxiety, hyperactivity, and acute depression in the premenstrual phase of her cycle are also consistent with this diagnosis. The mixed presentation of mood changes is also consistent with this diagnosis, as bipolar I disorder, mixed type typically involves a combination of manic and depressive episodes. Cyclothymic personality disorder is characterized by periods of hypomania and depression that last for at least 2 years. Generalized anxiety disorder is characterized by excessive worry and
Response After Pruning Based on RACE as $\mathcal{D}_{\text{non-math}}$ (correct) Answer: 'adjustment disorder with depressed mood' The most likely diagnosis is adjustment disorder with depressed mood. This is a condition where the person's symptoms of depression are triggered by a stressful event, but the symptoms are not severe enough to meet the criteria for a major depressive disorder. The symptoms of depression are typically present for at least two days, are persistent, and are severe enough to cause significant distress or impairment in social or work functioning. The diagnosis of adjustment disorder with depressed mood is typically made based on the criteria outlined in the DSM-IV. The symptoms of depression in adjustment disorder with depressed mood include: 1. Depressed mood 2. Loss of interest in activities 3. Insomnia or hypersomnia 4. Fatigue
Response After Pruning Based on MMLU as $\mathcal{D}_{\text{non-math}}$ (incorrect; hit max number of output tokens) Answer: B Explanation: A 30-year-old nulliparous female presenting with mood changes is a classic presentation for bipolar disorder, particularly in the early stages. The patient's symptoms of irritability, lethargy, and an acute depression during menses are classic for a manic episode. The history of tuboplasty is a significant clue that points towards a bipolar disorder, especially given the rapid onset of symptoms and the pattern of episodes. The patient's past history of depression and the fact that she and her husband have been trying to conceive for over 2 years suggests a pattern of mood instability and possibly a depressive episode. The most likely diagnosis is bipolar I disorder, mixed. The symptoms of mood changes, irritability, lethargy, and

Table 4: Responses to a MMLU question before and after pruning parameters identified by MathNeuro for Llama 3 1B IT.

Prompt Question: A 30-year-old nulliparous female presents to the office with the complaint of mood changes. She says that for the past several months she has been anxious, hyperactive, and unable to sleep 3 to 4 days prior to the onset of menses. She further reports that on the day her menses begins she becomes acutely depressed, anorectic, irritable, and lethargic. She has no psychiatric history. Physical examination findings are normal. She and her husband have been trying to conceive for over 2 years. History reveals a tuboplasty approximately 1 year ago to correct a closed fallopian tube. The most likely diagnosis is Answer choices: ['adjustment disorder with depressed mood', 'bipolar I disorder, mixed', 'cyclothymic personality', 'generalized anxiety disorder']
Response Before Scaling (correct; hit max number of output tokens) Answer: A Explanation: The patient's symptoms of mood changes, anxiety, hyperactivity, inability to sleep, acute depression, anorexia, irritability, and lethargy, particularly in the context of her history of tuboplasty, are characteristic of adjustment disorder with depressed mood. The patient's anxiety, hyperactivity, and acute depression in the premenstrual phase of her cycle are also consistent with this diagnosis. The mixed presentation of mood changes is also consistent with this diagnosis, as bipolar I disorder, mixed type typically involves a combination of manic and depressive episodes. Cyclothymic personality disorder is characterized by periods of hypomania and depression that last for at least 2 years. Generalized anxiety disorder is characterized by excessive worry and
Response After Scaling Based on RACE as $\mathcal{D}_{\textbf{non-math}}$ (correct; hit max number of output tokens) Answer: A Explanation: The patient's symptoms of mood changes (anxiety, hyperactivity, irritability, lethargy) and the physical symptoms (anorexia) of anorexia nervosa, which are typically seen in the context of hormonal fluctuations, are consistent with this diagnosis. The physical symptoms of anorexia nervosa are also consistent with the tubal surgery. The patient's symptoms do not meet the criteria for a manic episode (i.e., she is not hyperactive or irritable for more than one week), and her symptoms do not meet the criteria for a depressive episode (i.e., she is not depressed for more than two weeks). The patient's age and the fact that she is trying to conceive do not support
Response After Scaling Based on MMLU as $\mathcal{D}_{\textbf{non-math}}$ (correct; hit max number of output tokens) Answer: A Explanation: The patient's symptoms of mood changes, anxiety, hyperactivity, and sleep disturbances, particularly in the context of her menstrual cycle, are characteristic of premenstrual syndrome (PMS). The patient's history of tuboplasty and the timing of her symptoms suggest that she may be experiencing a cyclical pattern of mood changes, which is a hallmark of bipolar I disorder. The patient's symptoms are also consistent with a diagnosis of adjustment disorder with depressed mood, which is a type of mood disorder that occurs in response to a significant life stressor. The patient's symptoms are not consistent with cyclothymic personality or generalized anxiety disorder, which do not typically present with such a cyclical pattern of mood changes. The best answer

Table 5: Responses to a MMLU question before and after scaling parameters identified by MathNeuro for Llama 3 1B IT.