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

Anonymous ACL submission

Abstract

Math reasoning is a highly active area of Large Language Model (LLM) research because it is a hallmark of artificial intelligence. However, few works have explored how math reasoning is encoded within LLM parameters and if it is a skill that can be isolated within a model. 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 method for isolating 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 removing those important for general language tasks. Pruning parameters MathNeuro identifies deletes a LLM's math reasoning ability without destroying its general language ability. Scaling these parameters by a small constant improves a pretrained or instruction-tuned LLM's performance by 4-15% on GSM8K 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

Mathematical 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). 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 for language controllability (e.g., Tang et al. 2024) or toxicity mitigation (e.g., Suau et al. 2024), b) doing so without altering behavior on other tasks like these existing works have done, 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 (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 and, therefore, 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) forward-only parameter importance methods (Sun et al., 2023; Tang et al., 2024) to math reasoning. We find one of these methods, LAPE, consistently fails to identify math-specific neurons across models, while the other, Wanda, identifies parameters important for math, 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 to isolate math-specific parameters we call Math Neurosurgery (MathNeuro).

[1] We will release experimental code upon publication.

Building on Wanda, MathNeuro uses weights and activations to calculate parameter importance to achieve a context-aware representation of importance. However, to isolate parameters important for math and not other abilities, MathNeuro removes 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 rigorously evaluating it with five recent LLMs ranging from sizes 1-8B parameters. By pruning the parameters identified by MathNeuro, we show that a model's math reasoning ability is effectively deleted. Despite destroying math reasoning, pruning these parameters results in a performance drop on other, non-math tasks similar to the impact of pruning random model parameters. We also find that scaling up MathNeuro-identified parameters by a small universal factor can boost a LLM's GSM8K (Cobbe et al., 2021) performance by 4-15% across models, both those that are instruction-tuned and pretrained. We further show that most of MathNeuro's effectiveness holds when calculating parameter importance using a *single* sample, highlighting that our method is not only simple and effective, but extremely data efficient. In addition, we show MathNeuro consistently identifies a similar subset of parameters as math-specific across different sets of samples. Lastly, we find math-specific parameters are located roughly evenly throughout a model's decoder blocks, suggesting math reasoning is likely distributed across a model's parameters rather than being concentrated in a specific layer or layers.

Our key contributions are as follows:

- We design MathNeuro, the first parameter identification method thoroughly evaluated for isolating math reasoning in LLMs.
- We demonstrate the effectiveness of this method by showing that deleting parameters identified by MathNeuro destroys a model's math performance and that scaling them by a universal factor can increase it by 4-15% across models of various sizes.
- 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, 2020; Dai et al., 2022; 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; Zhao et al., 2024; Xin et al., 2019). Many of these methods are computationally expensive because they require gradient information to calculate parameter importance, which may be 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 are known to find parameters important for the domains they study, it is unclear whether they would be effective in identifying parameters most important for math reasoning, which could potentially be distributed throughout a model or interwoven with other important natural language abilities given the task's complexity.

Math Skill Localization in LLMs Some studies have explored how math knowledge is encoded within LLMs (Hanna et al., 2023; Rai and Yao, 2024; Stolfo et al., 2023). 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 ultimately do not identify parameters critical for a model's math performance but rather ones relating to processing different math concepts.

3 Methods

We propose MathNeuro, a parameter identification method that calculates importance based solely on forward passes. First, our method separately identifies LLM parameters important for a math and non-math, general language task using input data 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. We describe the method in more detail below.

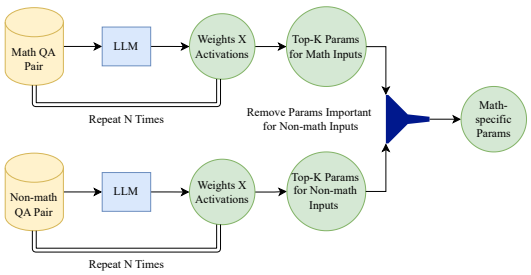

Figure 1: Overview of MathNeuro. First, we sum weights times activations separately over N samples for math and non-math inputs, finding the top-K parameters for each input type. Next, we find math-specific parameters by removing parameters that are important for non-math inputs.

3.1 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 individual parameters:

$$S_{ij} = |W_{ij}| \cdot \|X_j\|_2 \quad (1)$$

where W_{ij} represents the weight, $|\cdot|$ is absolute value, 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 critical elements of parameter importance because both small but highly activated weights and large but lightly activated weights can significantly influence model outputs. MathNeuro flips Wanda by identifying the parameters with the largest weights times activations as being most important to a given task.

3.2 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 previous work has found a high level of overlap between parameters important for different tasks (Tang et al., 2024). As a result, 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 accumulate[2] the absolute value of weights times activations for each parameter in attention heads and MLP layers across N samples from a math dataset and a dataset for an unrelated natural language task. We focus on MLP layers and attention heads 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). We compute scores for each parameter over math and non-math inputs:

$$S_{ij}^{\text{math}} = \sum_{k=1}^{N} |W_{ij}| \cdot \|X_{jk}\|_2 \quad \text{for} \quad X \in \mathcal{D}_{\text{math}} \quad (2)$$

$$S_{ij}^{\text{non-math}} = \sum_{k=1}^{N} |W_{ij}| \cdot \|X_{jk}\|_2 \quad \text{for} \quad X \in \mathcal{D}_{\text{non-math}} \quad (3)$$

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 activation level. We show the impact of each intervention on both math and non-math performance across five popular LLMs varying in size from 1-8B parameters. We perform experiments for parameter identification using both a larger number of samples (500) 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, b), Gemma 2 2B IT (Team et al., 2024), Llama 3.2 3B IT (MetaAI,

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

b), and Llama 3.1 8B IT (MetaAI, a). We display results for Llama 3.2 1B IT in the sections 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 because we want to evaluate whether 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, however, serves as a baseline for whether MathNeuro works for a pretrained, non-IT model.

Datasets For identifying math-specific parameters, we use the GSM8K dataset due to its popularity and high quality (Cobbe et al., 2021). We calculate parameter importance based on the GSM8K training split and evaluate the impact of each method on the GSM8K test split for each model. 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 finding and 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., 2021) for measuring general knowledge. These datasets are general natural language understanding tasks that are different from math reasoning. While MMLU does contain some math-related questions, it assesses a wide variety of knowledge which, in aggregate, is mostly not math specific. We conduct all evaluations using the Eleuther AI LM Evaluation Harness (Gao et al., 2024) and prompt models to respond to GSM8K questions in a 8-shot chain-of-thought (CoT) format, 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 removing 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. We use LAPE's default implementation for pruning and more highly activating neurons it identifies.

(c) *Random Parameter Identification*: As a sanity check, we randomly select the same number of parameters as those identified by MathNeuro.

4.2 Pruning Top Math Parameters

To test if the four parameter identification methods (MathNeuro and the three baselines) identify parameters important for math reasoning, we identify 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 GSM8K performance is impacted by parameter proportion.

Figures 2, 12, 13, 14 and 15 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 shown in these plots, only MathNeuro and Wanda eliminate math performance across models, while LAPE is unable to identify parameters important for math. However, Wanda also destroys each model's ability to perform non-math tasks, as shown in large drops in MMLU and RACE performance. Unlike Wanda, MathNeuro effectively isolates math-specific parameters across models, as reflected in the method resulting in significantly smaller decreases in non-math performance that are similar to the effect of randomly pruning parameters in most cases.

4.3 Scaling Top Math Parameters

We next evaluate model performance when more highly activating math-specific parameters by scaling the weights by a universal factor. For smaller IT models, we find the scalar 1.1 performs best, while for larger IT models (Llama 3.1 8B and 3.2 3B), a slightly smaller factor (1.0625) works better. For Phi 1.5, a pretrained model, 1.025 works best. While we leave a rigorous study of this hyperpa-

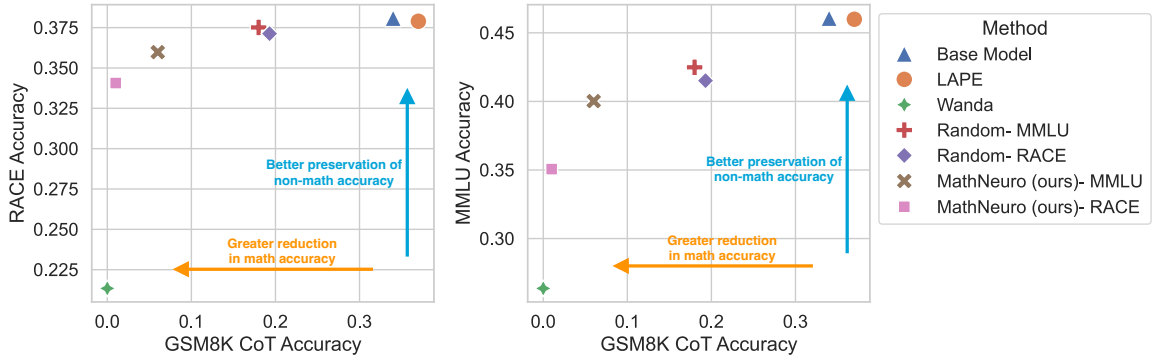

Figure 2: Effect of ***pruning*** identified parameters on math and non-math performance for Llama 3.2 1B IT based on calculating the top 15% of parameters. Ideal methods should fall in the top left of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$.

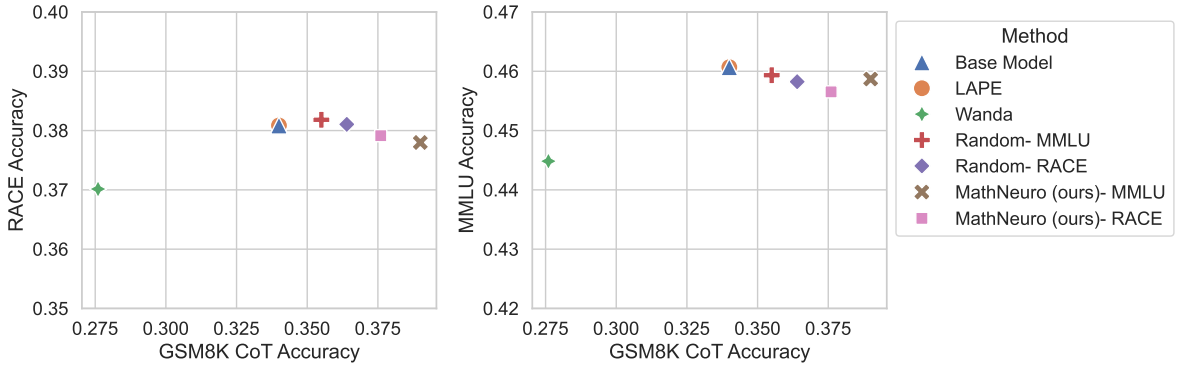

Figure 3: Effect of ***scaling*** identified parameters by 1.1 on math and non-math performance for Llama 3.2 1B IT based on calculating the top 5% of parameters. Ideal methods should fall in the top right of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$.

rameter to future work due to its computational expense, see Appendix F for our experimentation with this hyperparameter. As in Section 4.2, we scale the top parameters identified by each method based on a random subset of 500 samples from each relevant dataset and repeat the process five times to identify average performance, reporting the parameter proportion with the best performance.

Figures 3, 17, 18, 19, and 20 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 4-15% across models, while scaling Wanda-identified parameters tends to harm performance. As with pruning, LAPE has no effect for most models except for increasing GSM8K performance for Llama 3.1 8B IT and 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. The GSM8K performance gains from scaling parameters identified by MathNeuro and the lack of catastrophic forgetting are consistent across models and highlight the potential for future work to utilize these parameters for interventions.

4.4 MathNeuro with a Single Sample

If MathNeuro could identify math-specific parameters using a single sample, then it could inform math reasoning interventions for settings in which data is limited such as for assessing a specific mathematical operation or topic. To evaluate if this is the case, we conduct experiments to see if MathNeuro's performance holds when identifying parameters based on a single math and non-math input. We conduct these experiments by pruning

or scaling parameters identified by each method and run each experiment five times using different random samples from each dataset. As shown in Figures 4, 21, 22, 23 and 24, MathNeuro still performs best at isolating math-specific parameters when pruning using a single sample, as shown in lower drops in RACE and MMLU performance. However, these performance drops, particularly for MMLU, are larger than when using more samples, suggesting additional samples help the method more effectively isolate math-specific parameters.

As shown in Figures 5, 25, 26, 27, and 28, 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

In the sections below, we conduct additional experiments to explore whether 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 also conduct a qualitative evaluation of sample model outputs after pruning parameters identified by MathNeuro to further explore how math and non-math outputs are affected by pruning. All experiments are conducted using Llama 3.2 1B IT.

Consistency of Math-specific Parameters We next explore whether 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 Llama 3.2 1B IT's 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 Figures 6 and 7, by 100 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 Llama 3.2 1B IT parameters MathNeuro identifies as math-specific. We first identify math-specific parameters using MathNeuro on 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 do this five times for different numbers of samples (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 the proportion of math-specific parameters identified varies when calculating based on different sample sizes and top parameter proportions.

As shown in Figures 8 and 9, while the largest number of parameters are identified as math-specific when calculating importance based on one sample due to randomness, in general, the amount of math-specific parameters identified by MathNeuro increases with the number of samples considered regardless of the proportion of top parameters calculated. The relatively high amount of overlap in top parameters between tasks displayed in these figures is likely why we see that MathNeuro performs better than existing parameter identification methods that do not remove 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 due to the fact that as this proportion increases, more of the model's top parameters are being considered, which are likely more invariant based on the task than when considering a smaller percentage of the model's top parameters. Indeed, the top 15% of model parameters are likely similar across tasks while the top 1% will vary more.

To explore where math-specific parameters are located within a model, we sum the number of

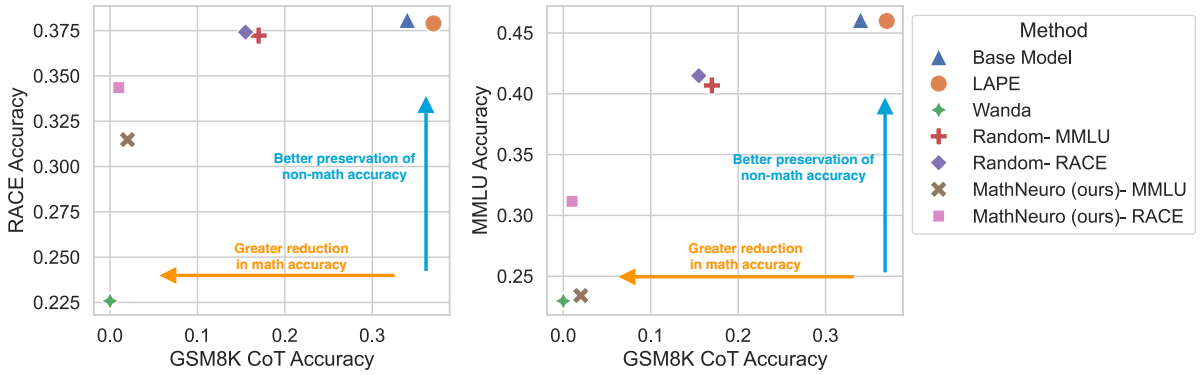

Figure 4: Effect of ***pruning*** identified parameters on math and non-math performance for Llama 3.2 1B IT for calculating the top 10% of parameters ***based on one sample***. Ideal methods should fall in the top left of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$.

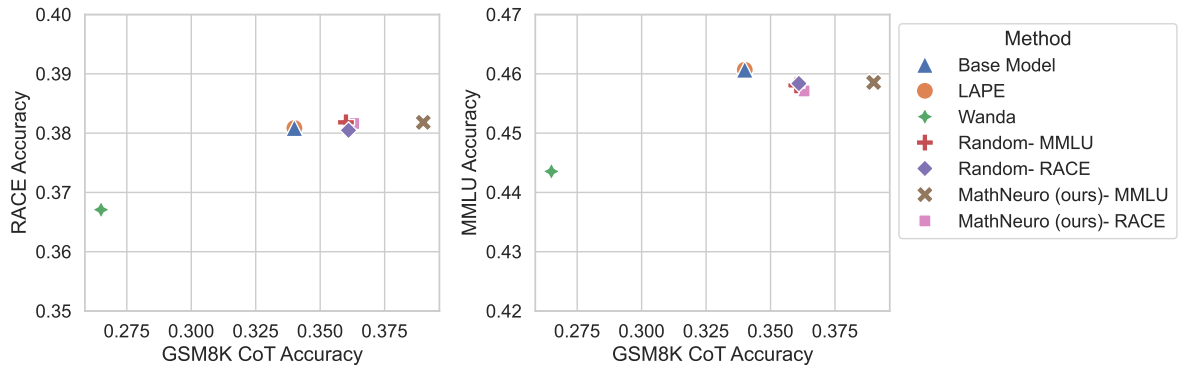

Figure 5: Effect of ***scaling*** identified parameters by 1.1 on math and non-math performance for Llama 3.2 1B IT for calculating the top 2.5% of parameters ***based on one sample***. Ideal methods should fall in the top right of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$.

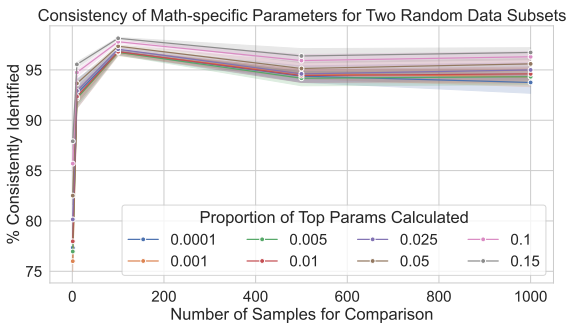

Figure 6: Consistency of math-specific parameters identified by MathNeuro for Llama 3.2 1B IT when identifying using GSM8K compared to RACE.

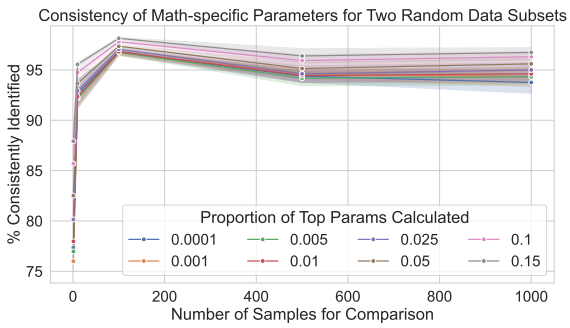

Figure 7: Consistency of math-specific parameters identified by MathNeuro for Llama 3.2 1B IT when identifying using GSM8K compared to MMLU.

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 Figures 10 and 11, the number of math-specific parameters MathNeuro identifies is relatively consistent across decoder blocks when using either MMLU or RACE for parameter identification. This suggests that math reasoning is distributed throughout the model rather than being concentrated in a few layers. The parameters identified in these figures correspond to

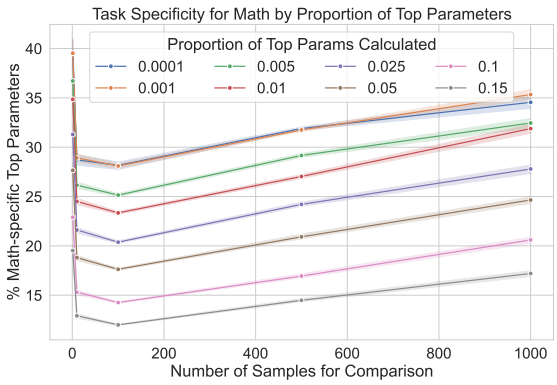

Figure 8: Percentage of top parameters that are math-specific as identified by MathNeuro for Llama 3.2 1B IT when identifying using GSM8K compared to RACE.

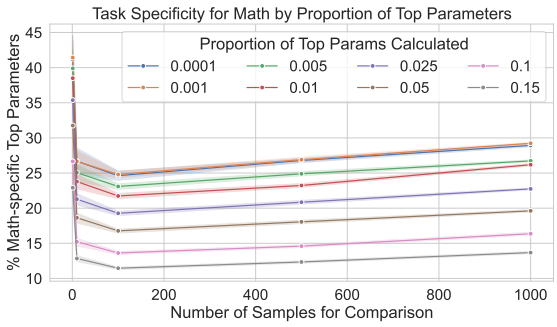

Figure 9: Percentage of top parameters that are math-specific as identified by MathNeuro for Llama 3.2 1B IT when identifying using GSM8K compared to MMLU.

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, we conduct a small qualitative analysis of Llama 3.2 1B outputs before and after pruning parameters identified by MathNeuro. As shown in Tables 1, 2, and 3, 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 model effectively responds to RACE questions, although it gets the same question wrong both before and after pruning. The model generates coherent output to MMLU questions both before and after pruning, though it gets the answer right when calculating parameters to prune based on RACE and wrong when calculating parameters to prune based on MMLU. These findings largely confirm those discussed above where the model is still able to perform non-math tasks after pruning math-specific parameters, although it does experience a drop in performance similar to that obtained from randomly pruning parameters.

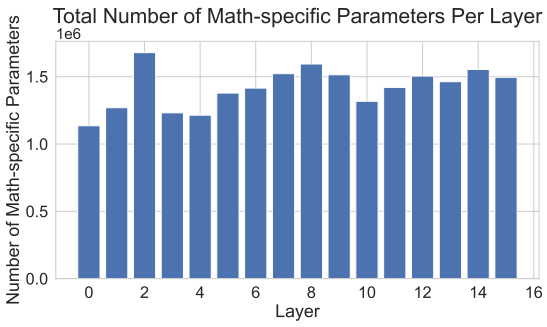

Figure 10: Distribution of math-specific parameters identified by MathNeuro for Llama 3.2 1B IT when identifying using GSM8K compared to RACE.

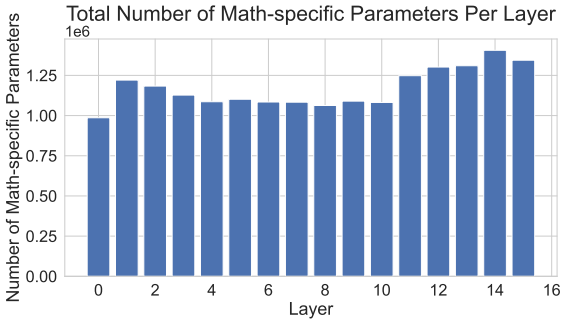

Figure 11: Distribution of math-specific parameters identified by MathNeuro for Llama 3.2 1B IT when identifying using GSM8K compared to MMLU.

5 Conclusion

Despite math reasoning being 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 for isolating 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 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 three key task-specific datasets used in other works (RACE for general language understanding, MMLU for general knowledge, and GSM8K for math reasoning), 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. 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 F 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].

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].

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.

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].

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. 2021. Measuring Massive Multitask Language Understanding. *arXiv preprint*. ArXiv:2009.03300 [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].

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

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

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].

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].

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].

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].

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].

A Pruning Results

Figures 12, 13, 14 and 15 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 16 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 Scaling Results

Figures 17, 18, 19 and 20 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 4-15% overall increase in math performance depending on the model.

C One Sample Pruning Results

Figures 21, 22, 23 and 24 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

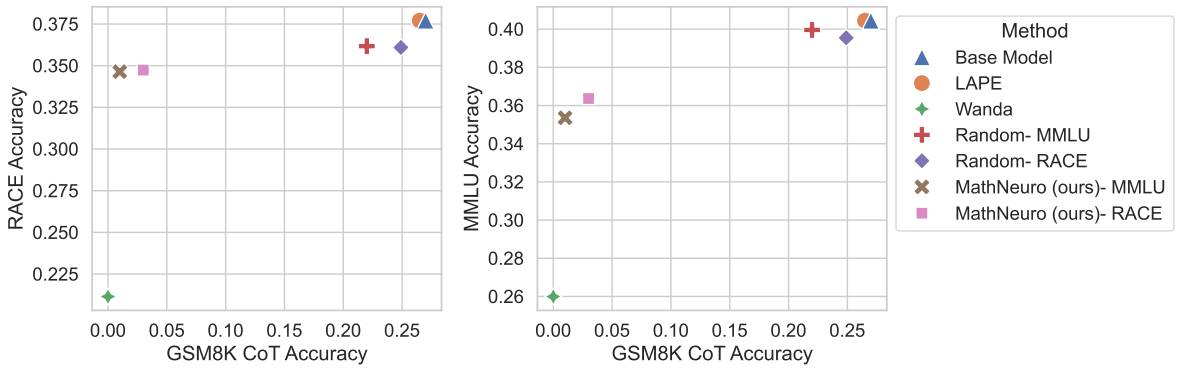

Figure 12: Effect of ***pruning*** identified parameters on math and non-math performance for Phi 1.5 based on calculating the top 5% of parameters. Ideal methods should fall in the top left of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$.

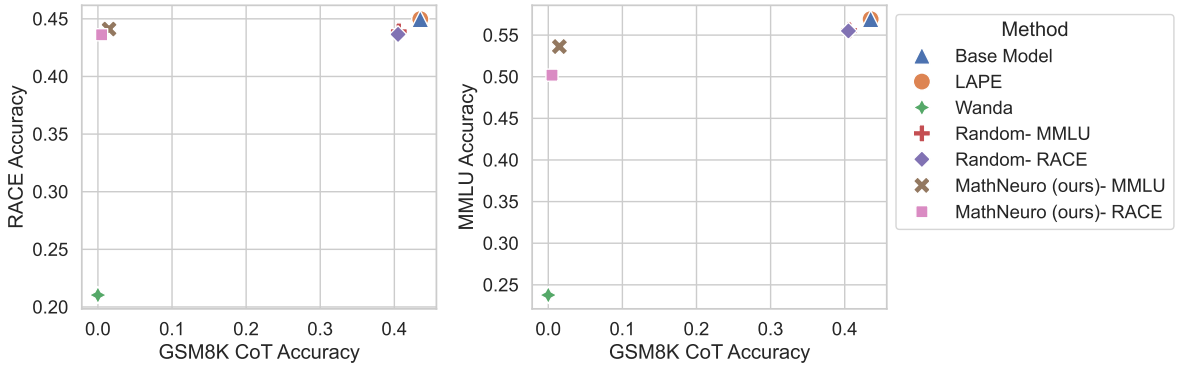

Figure 13: Effect of ***pruning*** identified parameters on math and non-math performance for Gemma 2 2B IT based on calculating the top 5% of parameters. Ideal methods should fall in the top left of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$.

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 Scaling Results

Figures 25, 26, 27 and 28 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 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 Sample Outputs

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

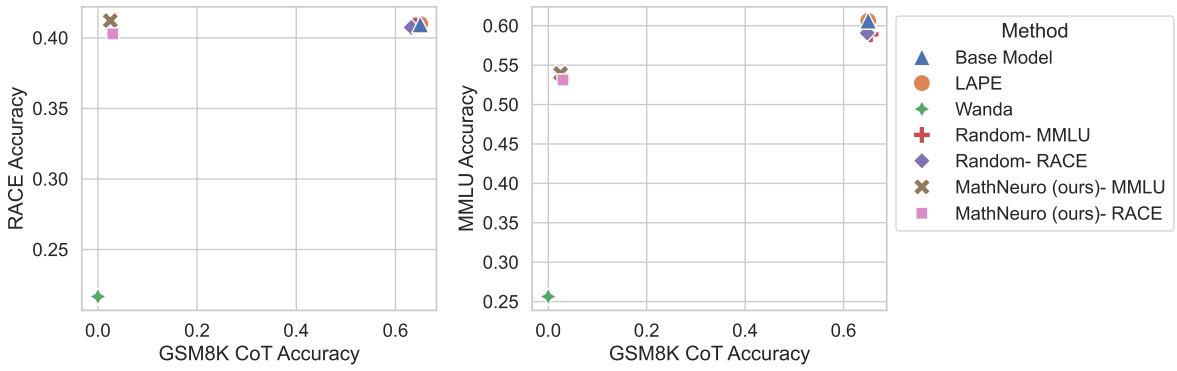

Figure 14: Effect of ***pruning*** identified parameters on math and non-math performance for Llama 3.2 3B IT based on calculating the top 2.5% (left) and 1% (right) of parameters. Ideal methods should fall in the top left of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$.

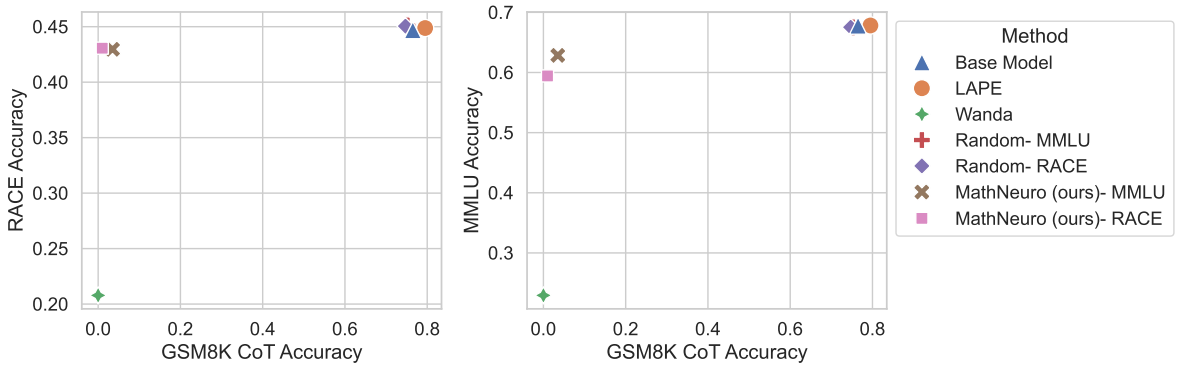

Figure 15: Effect of ***pruning*** identified parameters on math and non-math performance for Llama 3.1 8B IT based on calculating the top 1% of parameters. Ideal methods should fall in the top left of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$.

F 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 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 smaller IT models (Gemma 2 2B IT and Llama 3.2 1B IT), 1.075 worked best or tied with 1.05, while for larger IT models (Llama 3.1 8B IT and Llama 3.2 3B IT) and Phi 1.5, 1.05 worked best. For the smaller IT 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 Gemma 2 2B IT and 1.1 tied with the results of 1.075 for Llama 3.2 1B IT. For Phi 1.5 and the larger IT models, we next tried scale factors between 1.05 and the minimum scale factor we used (1.025) and between 1.05 and the maximum scale factor we used (1.0625), finding that 1.025 worked best for Phi 1.5 and 1.0625 worked best for the larger IT models. Each scale factor considered increased performance across models. The results of this grid search for Llama 3.2 1B are displayed in Figure 29.

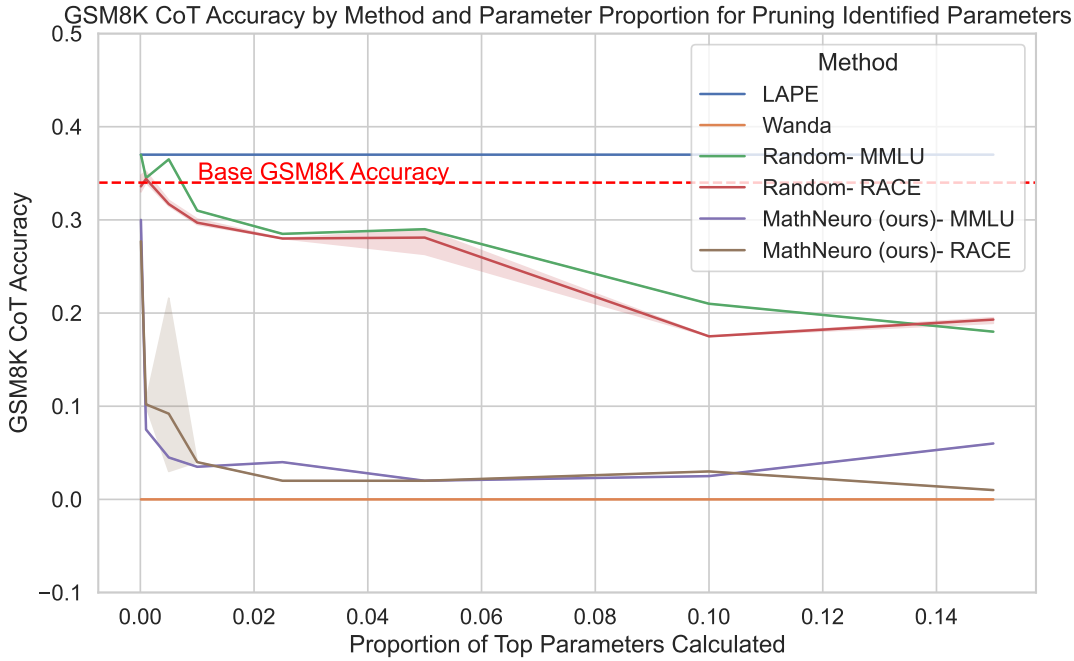

Figure 16: Impact of parameter proportion on GSM8K performance for ***pruning*** parameters identified by each method for Llama 3.2 1B IT.

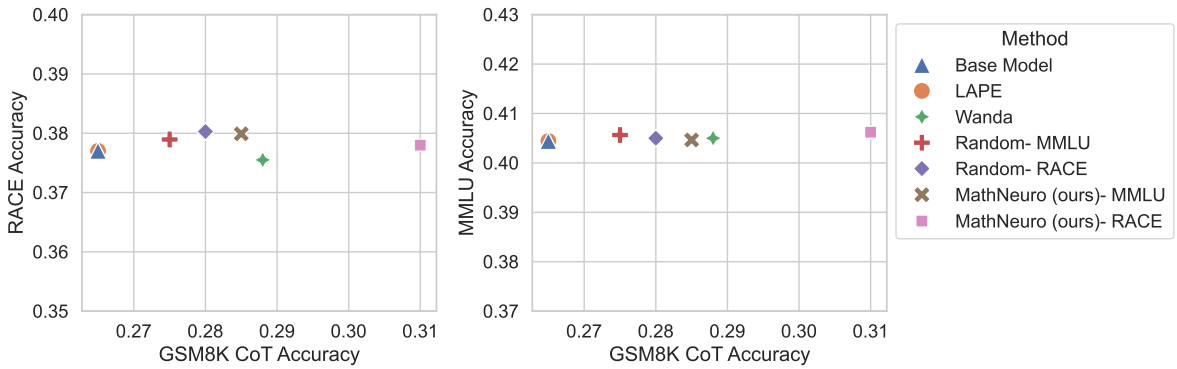

Figure 17: Effect of ***scaling*** identified parameters by 1.025 on math and non-math performance for Phi 1.5 based on calculating the top .1% of parameters. Ideal methods should fall in the top right of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$.

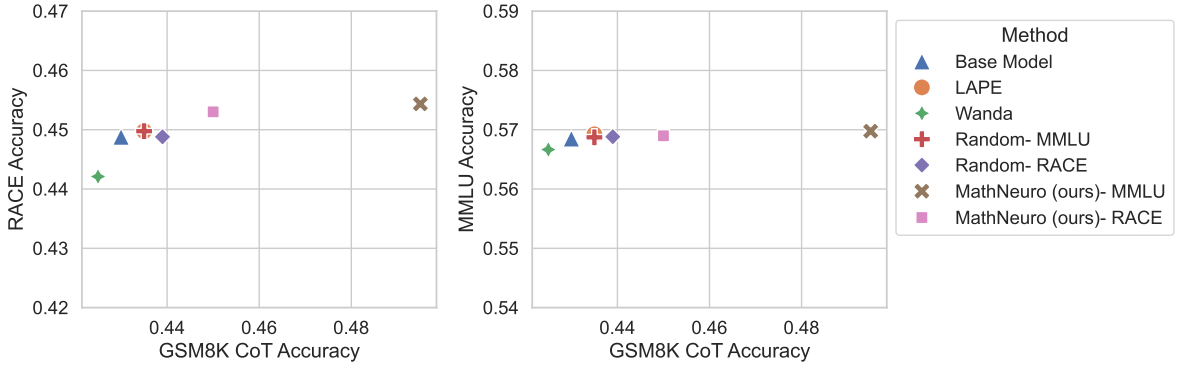

Figure 18: Effect of ***scaling*** identified parameters by 1.1 on math and non-math performance for Gemma 2 2B IT based on calculating the top 5% of parameters. Ideal methods should fall in the top right of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$.

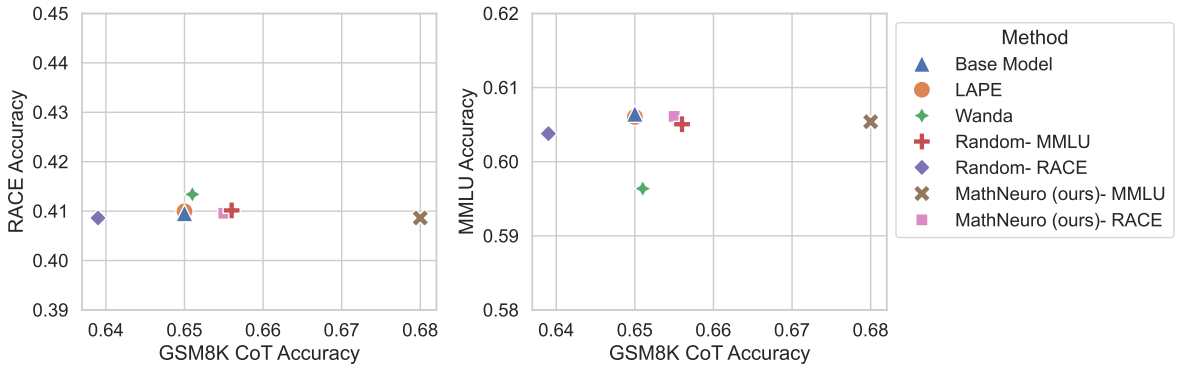

Figure 19: Effect of ***scaling*** identified parameters by 1.0625 on math and non-math performance for Llama 3.2 3B IT based on calculating the top 1% of parameters. Ideal methods should fall in the top right of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$.

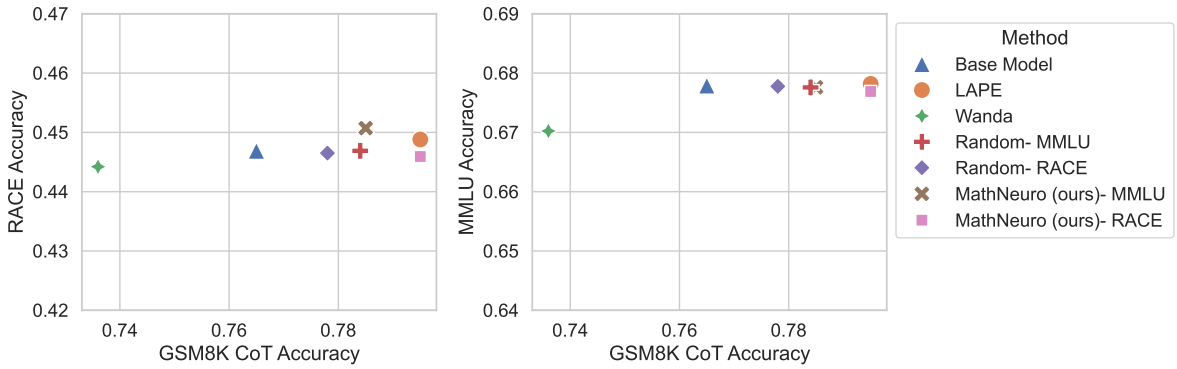

Figure 20: Effect of ***scaling*** identified parameters by 1.0625 on math and non-math performance for Llama 3.1 8B IT based on calculating the top 1% of parameters. Ideal methods should fall in the top right of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$.

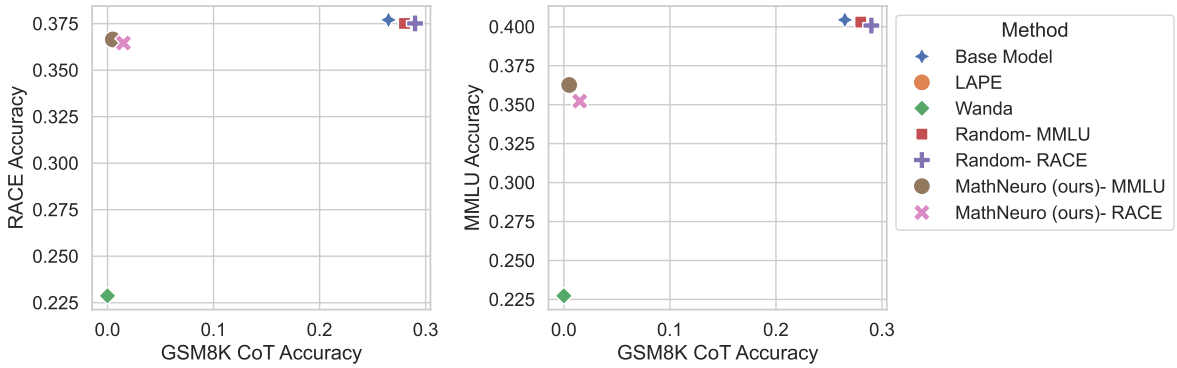

Figure 21: Effect of ***pruning*** identified parameters on math and non-math performance for Phi 1.5 for calculating the top .5% of parameters ***based on one sample***. Ideal methods should fall in the top left of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$.

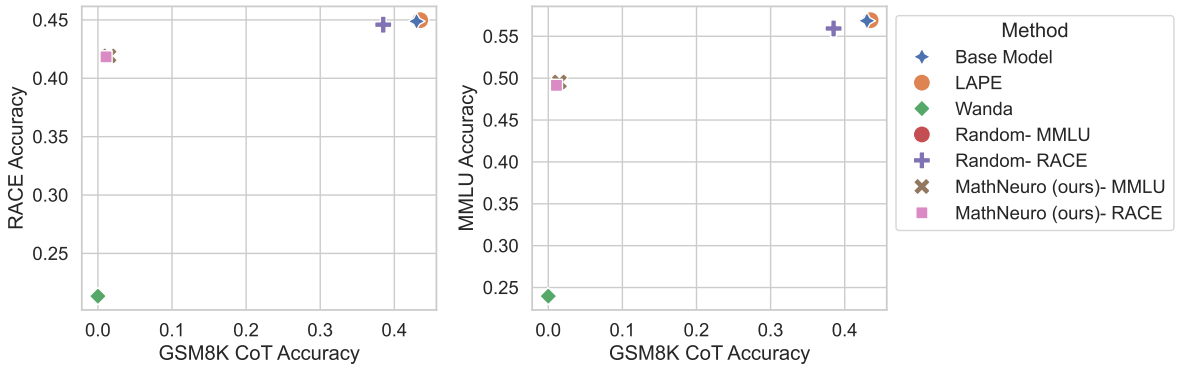

Figure 22: Effect of ***pruning*** identified parameters on math and non-math performance for Gemma 2 2B IT for calculating the top 2.5% of parameters ***based on one sample***. Ideal methods should fall in the top left of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$.

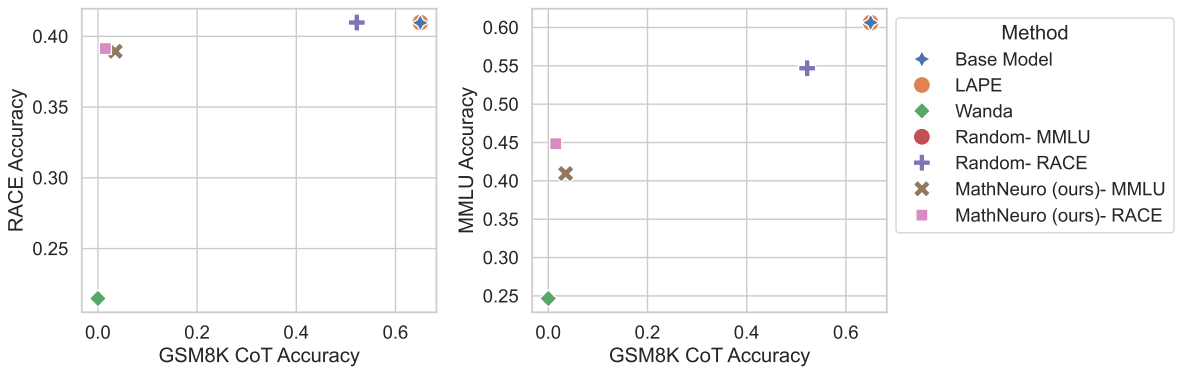

Figure 23: Effect of ***pruning*** identified parameters on math and non-math performance for Llama 3.2 3B IT for calculating the top 10% of parameters ***based on one sample***. Ideal methods should fall in the top left of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$.

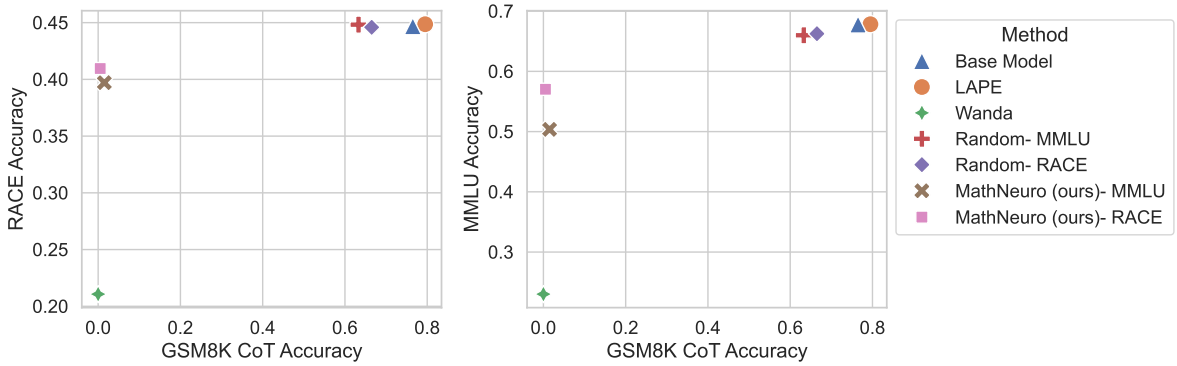

Figure 24: Effect of ***pruning*** identified parameters on math and non-math performance for Llama 3.1 8B IT for calculating the top 5% of parameters ***based on one sample***. Ideal methods should fall in the top left of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$.

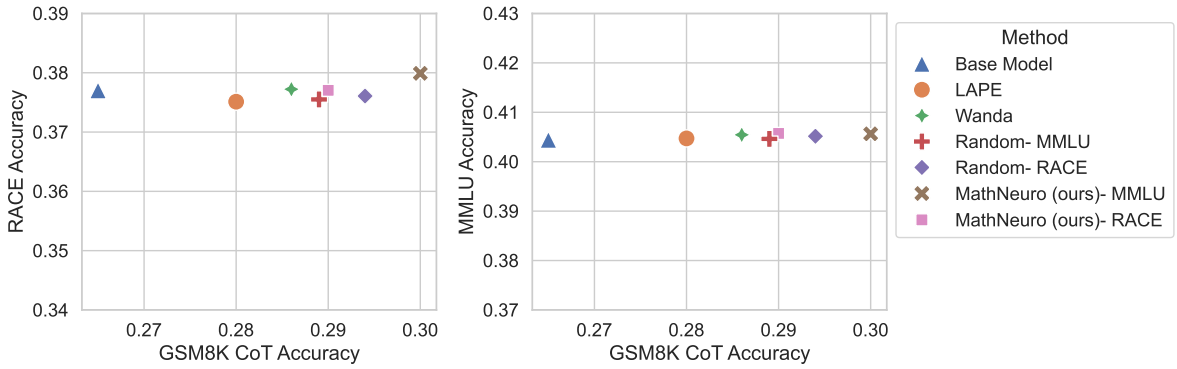

Figure 25: Effect of ***scaling*** identified parameters by 1.025 on math and non-math performance for Phi 1.5 for calculating the top 2.5% of parameters ***based on one sample***. Ideal methods should fall in the top right of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$.

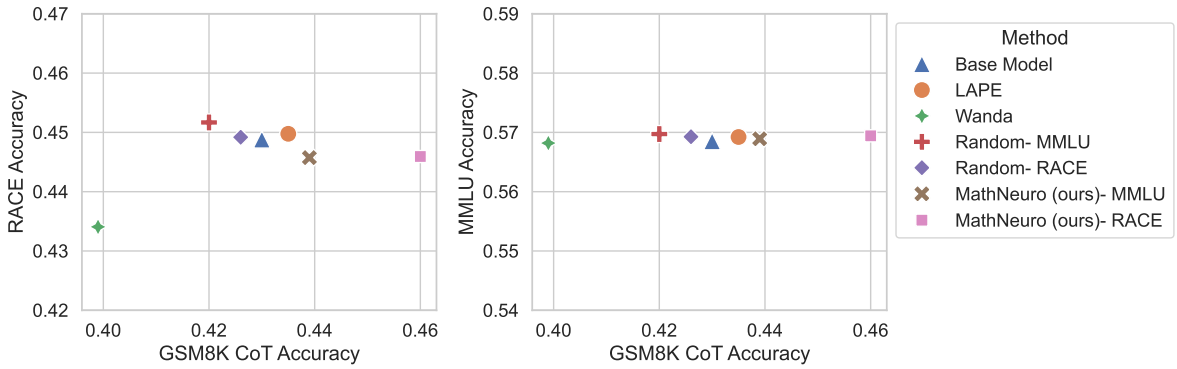

Figure 26: Effect of ***scaling*** identified parameters by 1.1 on math and non-math performance for Gemma 2 2B IT for calculating the top 2.5% of parameters ***based on one sample***. Ideal methods should fall in the top right of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$.

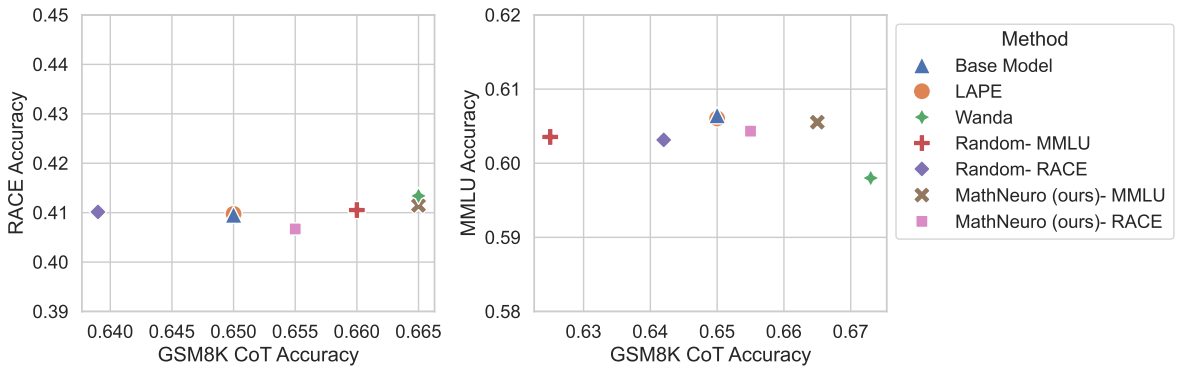

Figure 27: Effect of ***scaling*** identified parameters by 1.0625 on math and non-math performance for Llama 3.2 3B IT for calculating the top 10% (left) and 15% (right) of parameters ***based on one sample***. Ideal methods should fall in the top right of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$.

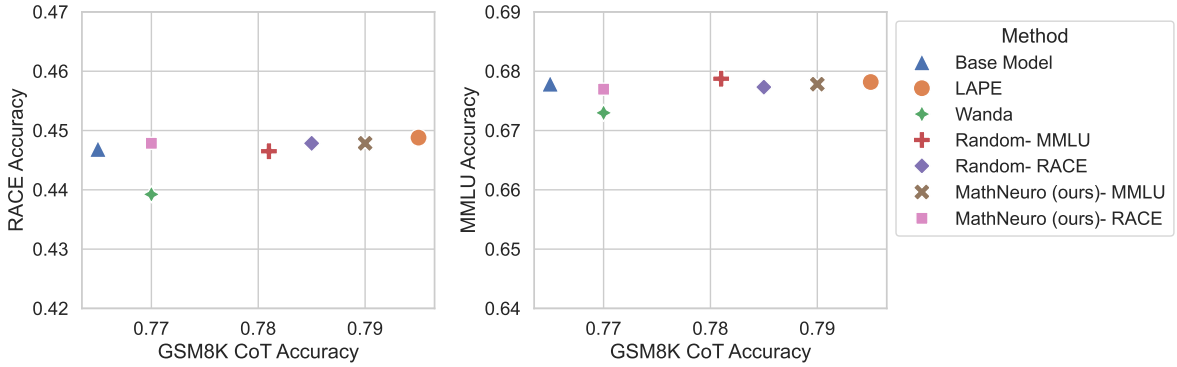

Figure 28: Effect of ***scaling*** identified parameters by 1.0625 on math and non-math performance for Llama 3.1 8B IT for calculating the top 15% of parameters ***based on one sample***. Ideal methods should fall in the top right of the plot. MMLU and RACE denote that a point was calculated using MMLU or RACE, respectively, as $\mathcal{D}_{\text{non-math}}$.

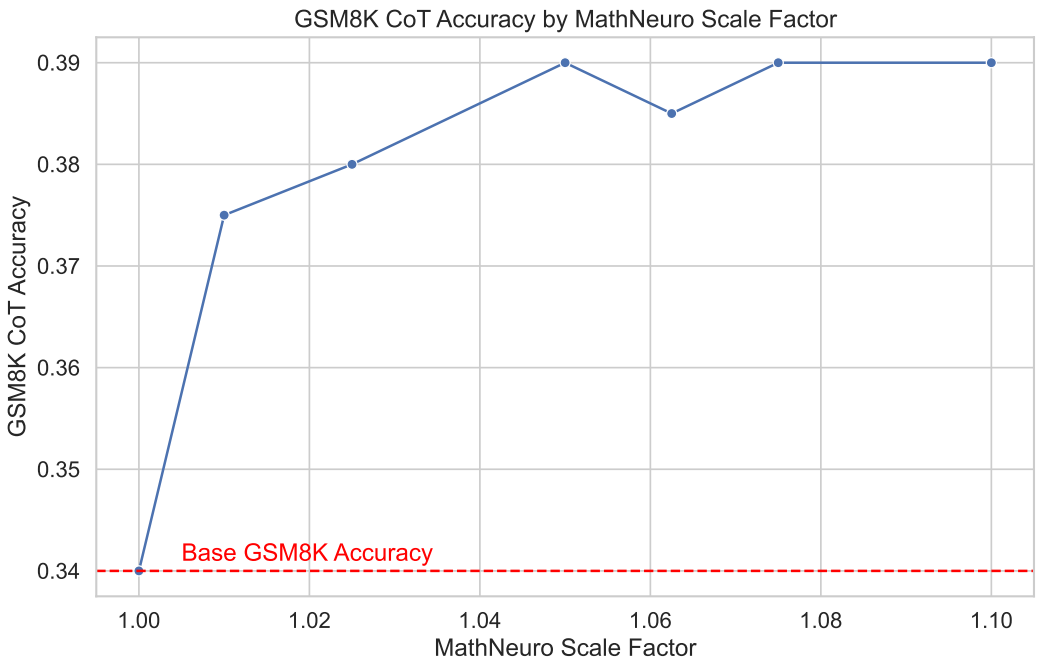

Figure 29: Impact of MathNeuro scale factor on GSM8K performance for Llama 3.2 1B.

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 (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 as a gift. Betty's grandparents gave her $15
Response After Pruning Based on MMLU (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.

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 Pruning (incorrect) Answer: 'can see more film stars'
Response After Pruning Based on RACE (incorrect) Answer: 'can see more film stars'
Response After Pruning Based on MMLU (incorrect) Answer: A. 'can write anything they like'

Table 2: Responses to a RACE 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 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 (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 (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 3: Responses to a MMLU question before and after pruning parameters identified by MathNeuro for Llama 3 1B IT.