
LoRA Augmented Generation (LAG): Memory-efficient LoRA-Boosted Mixture of Experts LLM

Alfonsus Rodriques Rendy

School of Data Science
CUHK Shenzhen
121040014@link.cuhk.edu.cn

Nicholas Oh

School of Data Science
CUHK Shenzhen
121040006@link.cuhk.edu.cn

Zoe Emmanuel Halim

School of Data Science
CUHK Shenzhen
121040045@link.cuhk.edu.cn

Aaron

School of Data Science
CUHK Shenzhen
121040033@link.cuhk.edu.cn

Abstract

The current understanding in large language model (LLM) research is that achieving detailed knowledge and advanced reasoning capabilities necessitates a substantial number of model parameters. However, exceedingly large models face challenges such as slower inference speeds and prohibitively expensive deployment, as all model parameters must fit within the inference hardware (typically GPU) memory. Recent advancements, including Retrieval Augmented Generation (RAG) and Sparse Mixture of Experts (SMoE), have been proposed to address these concerns. Nonetheless, these techniques still demand considerable memory for parameter storage and do not significantly enhance reasoning capabilities. In response to these limitations, we propose LoRA Augmented Generation (LAG), a novel approach designed to improve reasoning and domain-specific knowledge while maintaining minimal memory and computational requirements. Empirical evaluations demonstrate that two versions of the LAG model significantly outperform the base Mistral 7B Instruct model. Performance benchmarking indicates that the incorporation of additional experts incurs only marginal additional costs.

1 Introduction

The field of natural language processing has seen significant advancements with the development of large language models (LLMs), where larger models have demonstrated superior performance due to their ability to encapsulate complex knowledge bases [1, 2]. However, the extensive computational and memory demands of these models limit their practicality and accessibility, particularly in resource-limited settings.

To mitigate these constraints, model fine-tuning such as Low-Rank Adaptation (LoRA) has been employed to enhance performance by adapting a small subset of parameters on small models, thereby maintaining efficiency while enhancing its ability on a specific tasks [3]. Although effective for specific tasks, this often restricts the model’s utility to narrow down-stream applications [4].

Therefore, there is a pressing need for innovative solutions that can provide the benefits of large models without their extensive resource requirements. In response to these challenges, we introduce the concept of LoRA Augmented Generation (LAG), a novel framework that integrates the Mixture of Experts (MoE) architecture with LoRA fine-tuning and prompt augmentation to boost the efficiency and capability of LLMs within resource-efficient and modularity principle. By employing LoRA mechanisms, LAG aims to enhance parameter efficiency, allowing the model to achieve higher performance without a proportional increase in computational overhead. This technique promises to bridge the gap between the need for powerful language processing tools and the limitations imposed by hardware capabilities, making advanced LLMs more accessible and practical for a wider range of applications and devices.

2 Related Works

Low Rank Adaptation Low Rank Adaptation (LoRA) optimizes low-rank matrix decomposition of the change in the weight matrix, while keeping the original weights frozen [3]. Specifically, let $\{W \in \mathbb{R}^{d \times k}\}$ denote the pre-trained weights and $\{\Delta W\}$ be its changes due to fine-tuning. LoRA aims to find $\{B \in \mathbb{R}^{d \times r}\}$ and $\{A \in \mathbb{R}^{r \times k}\}$ where r is usually much smaller than both d and k , such that $\{W + \Delta W\} = \{W + BA\}$ minimizes a loss function respective to the language modelling task. Due to the low rank matrices, LoRA is a parameter-efficient fine-tuning method that requires lower time and computing resources.

If we let the dimension of weights $\{W\}$ be $d \times d$ and denote L to be the number of weights to fine-tune, LoRA trains $2 \times L \times d \times r$ parameters as opposed to $L \times d \times d$, where $r \ll d$. LoRA remains as one of the most popular and strongest fine-tuning method for language models.

For a text generation task, given input sequence of tokens $\{x_1, \dots, x_N\}$, LoRA minimizes the cross-entropy loss:

$$\min_{\{B, A\}} - \sum_{t=1}^N \log (P_{\{W+BA\}} (x_t | x_1, x_2, \dots, x_{t-1}))$$

Mixture of Experts The Mixture of Experts (MoE) framework is designed to exploit sparse computing by dynamically selecting the appropriate network for processing inputs, akin to dynamic pruning [5, 6]. Building on this, the Mixtral $8 \times 7B$ model, an extension of the Mistral 7B, employs an advanced routing mechanism that directs inputs to two out of eight distinct neural networks, or ‘experts’, after each self-attention layer. Despite operating with only 13B active parameters per token—far fewer than dense models like Llama 2 70B and GPT-3.5—Mixtral achieves superior performance. Nonetheless, it retains a total of 47B parameters across all experts [7], which needs to be stored in whole within the execution hardware’s memory during inference. This limitation motivates our research towards a new architecture that reduces the parameter count while improving performance beyond existing benchmarks.

Retrieval Augmented Generation Retrieval Augmented Generation (RAG) represents a sophisticated method to enhance large language models (LLMs) by granting them access to an external, non-parametric data source. For a given input query x , a pre-trained retriever analyzes x and retrieves a set of the top- K documents z which are the most relevant to the query. These documents z are then incorporated as additional context by concatenating them with x in the LLM’s input [8]. RAG excels in addressing knowledge-intensive tasks and presents a viable alternative to conventional fine-tuning methods. By preserving the model’s pre-trained capabilities and selectively integrating external context, RAG effectively mitigates *catastrophic forgetting*, where a pre-trained model loses general knowledge and abilities after fine-tuning [9].

3 Approach

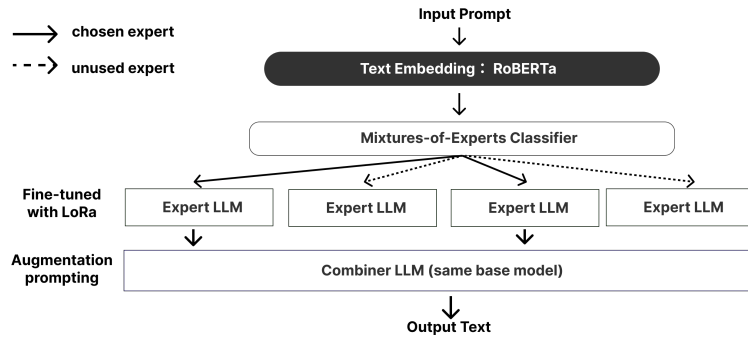


Figure 1: LoRA Augmented Generation Architecture

3.1 LoRA Augmented Generation Framework

The objective of LAG (LoRA Augmented Generation) is to construct a memory-efficient sequence-to-sequence language model that attains the performance of more parameter-intensive models on general tasks. The innovation of LAG stems from its unique augmentation method, which integrates a LoRA-fine-tuned model with the same base parameters and prompt augmentation techniques.

The architecture of LAG comprises three key components: (1) a RoBERTa-based experts classifier, (2) a LoRA-fine-tuned expert LLM, and (3) a combiner LLM. In practice, when provided with a user query sequence x , LAG will try to generate a relevant output sequence y (e.g., an answer to a question or execution of an instruction) by leveraging domain-specific sequences $z = \{z_i\}_{i=1}^d$ relevant to x . These domain-specific sequences equip LAG with additional knowledge tailored to the particular expert, enabling it to provide a response that surpasses the capabilities of general models. Ultimately, this framework allows the combiner LLM to assimilate diverse domain-specific insights, potentially enhancing its response generation capabilities.

3.2 Expert Classifier

The expert classifier is constructed by augmenting a RoBERTa text encoder with a classification head, which, when given an input query x , embeds the query semantics into a 768-dimensional vector and extracts relevant information pertinent to expert through a supervised multi-label classification task. The classifier’s output is a binary vector $y \in \{0, 1\}^d$, where d denotes the number of experts utilized, with each binary value indicating the expert’s relevancy. The primary purpose of this classifier is to identify which experts may be beneficial in addressing a given prompt, while requiring lower computational costs to maintain good performance compared to LLM-powered expert classification.

3.3 Expert LLM Fine-Tuning and Augmentation

The expert LLM in our framework is a fine-tuned model specialized in a specific domain. Our design principles for augmenting the expert LLM emphasize efficiency and modularity. To achieve efficient memory utilization and reduce inference time, we employed Low Rank Adaptation (LoRA) for fine-tuning. Notably, with the application of LoRA across d experts, less than 1% of all parameters are updated—equivalent to approximately 10M to 40M parameters for 7B base model. This approach not only enables parallel processing during inference by simply swapping the LoRA adapters but also maintains memory requirements at the base model size. Moreover, the modular nature of our augmentation allows for the addition of any number of experts based on user requirements, enhancing the scalability of the framework.

Additionally, the augmentation of the expert’s knowledge leverages a technique similar to Retrieval Augmented Generation (RAG)—namely, prompt augmentation. This method enriches the input to the expert LLM, enhancing its response quality. Analogous to RAG, our model retains the base model’s original parameters, which significantly reduces the risk of *catastrophic forgetting*, a common problem to most fine-tuned language systems. Furthermore, the streamlined process from the expert LLM to the combiner LLM introduces only one level of sequential bottleneck, which negligibly impacts the overall inference time.

4 Experiments

4.1 Dataset Collection and Pre-processing

The training data for the experts LLM and classifier were sourced from various open-source domain datasets, specifically, UltraInteract_sft [10], OpenMathInstruct [11], FinGPT [12], and Distilabel-ohp-writing which corresponds to four different field that our expert LLM specializes in: logic with chain of thoughts, math and coding, finance, and writing respectively.

Expert training After the data was filtered, it was transformed to Mistral instruction template to match the LLM base model specification: `<s>[INST] Question [/INST] Answer</s>`. For datasets with more than one feature relating to inputs, the entries were concatenated into a single entry before being formatted as the above.

Classifier training In the training of our classifier, we exclusively used user queries without incorporating any additional information (such as instructions or contextual details) that may be present in the dataset. This approach was adopted to minimize the risk of overfitting. We manually annotated each data point with the relevant expertise field label. Specifically, a label of an expert on a data was assigned as '1' if it (1) pertained to the dataset related to the domain or (2) contained overlapping information with another domain—for example, some Chain of Thought (CoT) datasets have overlapping data with the math and coding domain. To ensure a fair representation and eliminate bias toward any particular domain, the dataset was randomly selected and balanced across the domains.

4.2 Experiment Setup

We chose mistralai/Mistral-7B-Instruct-v0.2 model as the base model for both expert and combiner considering its performance to be one of the top among open-source LLM at the time of the experiment [13].

Classifier Training For the expert classifier, we trained a classification adapter with three FC layers with 520, 245, and 96 hidden units respectively. The classification head takes input of the pooled RoBERTa encoder representation of the query [14]. We trained the classifier using supervised setting as multi-label classification task on 22,910 training examples using multi-label cross-entropy loss. We tuned the hyper-parameter using K-fold cross validation.

Expert Fine-Tuning We fine-tuned four distinct Mistral 7B models using Unsloth—an optimized QLoRA fine-tuning framework powered by the Triton engine [15]—each specializing in different knowledge domains: (1) logic and chain of thought (CoT), (2) mathematics and coding, (3) finance, and (4) writing. To enhance efficiency, we employed 4-bit quantized weights with an update weight rank of 16.

Expert Augmentation on Combiner We augmented additional information given by the expert LLM through prompt augmentation. Specifically, if a particular expert is chosen by the classifier, the expert LLM must provide a domain-specific response to the input query which are then appended to the query for the combiner LLM.

4.3 Evaluation Method

We evaluated the quality of the outputs both qualitatively and quantitatively. Quantitative assessments aimed to establish an objective benchmark comparison with existing large language models (LLMs). For this purpose, we conducted evaluations using MT-Bench with LLAMA3-70b as a judge [16]. Additionally, we employed human annotators to assess the quality of the responses, comparing them to those generated by the base model without expert knowledge augmentation.

5 Results and Analysis

Inference Performance Evaluation The study examined model inference time across three scenarios: (1) varying the number of experts called while maintaining a constant input size, (2) varying the input size with a single expert call, and (3) varying the input size with all four experts called. It was observed that employing batched inference initially increased inference time with an increasing number of experts; however, beyond a certain point, the additional experts resulted in diminishing returns. This diminishing effect was limited, as evidenced by significantly increased inference times with very large batch sizes (e.g., 64 and 128), potentially indicating a hardware-dependent constraint [see Figures 2 and 3 in Appendix].

The primary outcome of this performance evaluation underscores that the inclusion of numerous experts does not linearly impact inference time. Under typical conditions, it can be anticipated that inference times with any number of experts will converge to approximately 2.5 times the required time.

MT-Bench Evaluation We evaluated our LAG model using LLAMA3-70b as a judge using two modes: (1) single evaluation where we obtained scoring for each responses in the benchmark prompt and (2) pairwise with baseline where we compare LAG with the base model mistral-7b.

Model	Turn 1 Score \uparrow	Turn 2 Score \uparrow	Average \uparrow
gpt-4*	7.594771	7.051020	7.328333
llama3-70b*	7.301325	6.881481	7.103147
lag-mistral-7b-v4	7.102564	6.658451	6.890940
lag-mistral-7b-v3	7.046250	6.556250	6.801250
gpt-3.5-turbo*	6.945513	6.478261	6.726190
mistral-7b	7.006494	6.392138	6.694409
lag-mistral-7b-v2	6.903846	6.148344	6.532248
lag-mistral-7b-v1	6.493750	6.068750	6.281250

Table 1: MT-Bench Single Evaluation in two rounds. Turn i score means the evaluation of i -th answer in the conversation. Model with (*) mark are just for reference

We observed that the performance of the LAG framework is highly sensitive to the prompting techniques employed. However, we can empirically demonstrate that with adjustments in prompt augmentation, the quality of the base LLM’s response can be significantly improved. Specifically, we identified two model outputs that surpass the performance of the base model when evaluated using LLAMA3-70b as a benchmark. These models utilized prompting augmentations that constrain the combiner to critically assess and selectively incorporate the expert’s knowledge, resulting in more detailed and focused responses that directly address the prompts.

Baseline	Model	Win Rate \uparrow	Loss Rate \downarrow	Adjusted Win Rate \uparrow
Mistral-7b	lag-mistral-7b-v3	35.27%	17.12%	59.08%
	lag-mistral-7b-v4	28.57%	24.67%	51.95%
	lag-mistral-7b-v2	20.69%	48.27%	36.20%
	lag-mistral-7b-v1	15.21%	49.13%	33.04%

Table 2: Pairwise Evaluation. Adjusted win rate is percentage of win and tie while win rate only considers the percentage of win

Qualitative Evaluation We found a writing prompt where the base Mistral 7B model failed to adhere to user instructions, resulting in inaccurate task execution. In contrast, our LAG Mistral model exhibited a superior understanding of prompts and consistently executed tasks successfully [Sample 1]. Additionally, there was an input prompt about quadratic formula calculation where the base Mistral 7B arrived at a wrong answer to the input question while LAG Mistral was able to solve the math question correctly while giving detailed steps [Sample 2]. This shows that the LAG framework can contribute positively to making a base model more theoretically adept and capable of solving technical questions and following user instructions accurately.

However, it is noteworthy that in some specific cases, the LAG Mistral model made errors where the Mistral 7B model succeeded. Specifically, in the first instance, which involved a word-counting task, the LAG Mistral inaccurately summoned a writing expert. In the second instance, related to a logic riddle, the model inappropriately engaged logic and math experts. These misaligned expert consultations misled the combiner, thereby negatively impacting performance. These issues can be attributed to inadequate training data for the classifier, which failed to recognize situations where no expert input was necessary. To address this, enhancing the training set with such examples could improve performance by leveraging the base model’s strengths in more general prompt types. Alternatively, adding specialized experts and refining the expert classifier for these specific task types could also effectively resolve the problem [see Appendix A.2].

6 Conclusion and Limitations

Conclusion We introduced LAG, a novel LLM framework designed for memory efficiency and scalability, aimed at enhancing the performance of a base LLM model in specialized domains while maintaining inference latency low through batched expert inference. This framework leverages advanced techniques such as LoRA, MoE, and RAG. Our findings indicate that the LAG framework has the potential to improve the base model’s performance even when applied to models with relatively few parameters. Further, by employing a minimally fine-tuned combiner, LAG mitigates the risk of catastrophic forgetting through maintaining the integrity of the base model’s parameters. However, it is noted that the LAG framework’s efficacy is sensitive to the prompting strategies employed within the combiner as well as the quality of the responses generated by the expert LLM.

Limitations The experiments conducted in this study utilized an untrained combiner and text embedder. Additionally, the limited size and quality of the dataset, along with the computational costs associated with training the expert LLMs and the selection of the base model, were detrimental to our model’s experimental performance. The proposed architecture could significantly benefit from a larger and higher-quality dataset, increased training epochs, and possibly a fine-tuned embedder and combiner. Future research should investigate the impact of these enhancements on the model’s performance.

References

- [1] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Nee-lakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.
- [2] Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. Language models as knowledge bases?, 2019.
- [3] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.
- [4] Sreyan Ghosh, Chandra Kiran Reddy Evuru, Sonal Kumar, Ramaneswaran S, Deepali Aneja, Zeyu Jin, Ramani Duraiswami, and Dinesh Manocha. A closer look at the limitations of instruction tuning, 2024.
- [5] Zhengyan Zhang, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. Moefication: Transformer feed-forward layers are mixtures of experts, 2022.
- [6] Zichang Liu, Jue Wang, Tri Dao, Tianyi Zhou, Binhang Yuan, Zhao Song, Anshumali Shrivastava, Ce Zhang, Yuandong Tian, Christopher Re, and Beidi Chen. Deja vu: Contextual sparsity for efficient llms at inference time, 2023.
- [7] Antoine Roux Arthur Mensch Blanche Savary Chris Bamford Devendra Singh Chaplot Diego de las Casas Emma Bou Hanna Florian Bressand Gianna Lengyel Guillaume Bour Guillaume Lample L  lio Renard Lavaud Lucile Saulnier Marie-Anne Lachaux Pierre Stock Sandeep Subramanian Sophia Yang Szymon Antoniak Teven Le Scao Th  ophile Gervet Thibaut Lavril Thomas Wang Timoth  e Lacroix William El Sayed Albert Q. Jiang, Alexandre Sablayrolles. Mixtral of experts, 2024.
- [8] Aleksandra Piktus   Fabio Petroni Vladimir Karpukhin Naman Goyal   Heinrich K  ttler   Mike Lewis Wen-tau Yih Tim Rockt  schel Sebastian Riedel Douwe Kiela Patrick Lewis, Ethan Perez. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2020.
- [9] Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning, 2024.
- [10] Lifan Yuan, Ganqu Cui, Hanbin Wang, Ning Ding, Xingyao Wang, Jia Deng, Boji Shan, Huimin Chen, Ruobing Xie, Yankai Lin, Zhenghao Liu, Bowen Zhou, Hao Peng, Zhiyuan Liu, and Maosong Sun. Advancing llm reasoning generalists with preference trees, 2024.
- [11] Shubham Toshniwal, Ivan Moshkov, Sean Narenthiran, Daria Gitman, Fei Jia, and Igor Gitman. Openmathinstruct-1: A 1.8 million math instruction tuning dataset. *arXiv preprint arXiv: Arxiv-2402.10176*, 2024.
- [12] Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. Fingpt: Open-source financial large language models. *FinLLM Symposium at IJCAI 2023*, 2023.
- [13] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L  lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth  e Lacroix, and William El Sayed. Mistral 7b, 2023.
- [14] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.
- [15] Unsloth, 2023.
- [16] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023.

A Appendix:

A.1 Prompting Techniques

We iteratively improve the prompt augmentation techniques to achieve better performance. The prompt augmentation determines several hyper-parameters (1) how much should the combiner rely on the expert's answer (2) how detailed should the combiner answer (3) in what way the expert's knowledge should be incorporated. The four prompts that we experimented on are:

Here are some references from various domain experts.
↳ according to an expert in logical reasoning and chain of thought: <CoT Augmentation>
↳ according to an expert in math and calculations: <Math Augmentation>
↳ according to an expert in computer programming (in any language): <Coding Augmentation>
↳ according to an expert in writing: <Writing Augmentation>
Answer the user's question or prompt: <User's Prompt>

Prompt 1: lag-mistral-7b-v1

We observed that the multiple experts sometimes give conflicting information. Therefore, we need the combiner to not blindly accept the expert's information. Thus, the second prompt adds a new constraint for the combiner to give additional judgement to the expert's additional information.

Here are some references from various domain experts. Note that the experts may not always be correct, and the user does not see any of these opinions. Therefore, use your judgment in using an expert's opinion, and rephrase it as your own answer
↳ according to an expert in logical reasoning and chain of thought: <CoT Augmentation>
↳ according to an expert in math and calculations: <Math Augmentation>
↳ according to an expert in computer programming (in any language): <Coding Augmentation>
↳ according to an expert in writing: <Writing Augmentation>
Here is the user's question or prompt: <User's Prompt>

Prompt 2: lag-mistral-7b-v2

We observed that the prompt in v2 still results in the model mentioning the existence of the experts making the result unnatural and often shorter than it should be. Therefore we add a new constraint in the prompt to make the combiner answer in a more detailed way without mentioning the experts' result.

Here are some references from various domain experts. Note that the experts may not always be correct, and the user does not see any of these opinions. Therefore, use your judgment in using an expert's opinion, and rephrase it as your own answer. Do NOT let the user know regarding the existence of experts
↳ according to an expert in logical reasoning and chain of thought: <CoT Augmentation>
↳ according to an expert in math and calculations: <Math Augmentation>
↳ according to an expert in computer programming (in any language): <Coding Augmentation>
↳ according to an expert in writing: <Writing Augmentation>
Here is the user's question or prompt: <User's Prompt>
Be extremely detailed and thorough in your responses. Elaborate on your answers. Focus on addressing the user's prompt effectively, directly, and helpfully.

Prompt 3: lag-mistral-7b-v3

The last version is just a paraphrase of the third prompt to test the consistency of the combiner's performance to incorporate the experts' information using similar prompting.

Here are some opinions to take into consideration. The opinions may not always be correct. The user does not see any of these opinions. Use your judgment to determine whether an opinion is helpful for the user's prompt, and give the user YOUR informed answer. Do NOT let the user know regarding the existence of these opinions.

↳ according to an expert in logical reasoning and chain of thought: <CoT Augmentation>

↳ according to an expert in math and calculations: <Math Augmentation>

↳ according to an expert in computer programming (in any language): <Coding Augmentation>

↳ according to an expert in writing: <Writing Augmentation>

Although the opinions may guide you, Do NOT mention receiving any opinions. Be very detailed, thorough, yet appropriately concise in your responses. Focus and place utmost priority in addressing the user's prompt directly, and helpfully. Here is the user's question or prompt: <User's Prompt>

Prompt 4: lag-mistral-7b-v4

A.2 Qualitative Evaluation Samples

Sample 1: Writing with Alphabetical Constraint

Prompt: Write a descriptive paragraph about a bustling marketplace, incorporating sensory details such as smells, sounds, and visual elements to create an immersive experience for the reader.

Follow-Up Prompt: Rework your previous response. Begin each sentence with the subsequent letter of the alphabet, commencing from B.

Response (LAG on left, Mistral-7B on right):

↳ **Expert(s) called:** Writing

Beneath the vibrant canopy of the marketplace, a symphony of sounds filled the air. Clamoring vendors called out to passersby, their voices rising above the chatter of shoppers and the laughter of children. Drums beat rhythmically, their sounds echoing through the narrow alleys. Echoes of clanging pots and pans added to the cacophony, their metallic resonance mingling with the drums. Freshly baked bread wafted through the air, tantalizing the senses and beckoning visitors to sample the warm loaves. Glistening in the sunlight, the deep reds and yellows of ripe fruits and vegetables beckoned. Hues of spices from nearby stalls added a piquant aroma, their fragrances weaving together in a complex tapestry of scent. Intriguing smells of exotic spices and herbs tantalized the nose, inviting exploration. Jewels of gold and silver adorned the wares of the jewelry vendors, their brilliance reflecting off the polished surfaces. Kaleidoscopic colors of fabrics and pottery shimmered in the dappled shade, their vibrancy a feast for the eyes. Lively banter between vendors and customers added to the marketplace's infectious energy. Marketgoers haggling over prices, their voices rising and falling, created a lively atmosphere. Nimble fingers of vendors deftly weaving baskets, their skill on display for all to see. Overhead, the sun cast dappled shadows on the cobblestone streets, adding to the marketplace's charm. Pottery stalls showcased intricately designed bowls and vases, their beauty a testament to the artisans' craftsmanship. Quaint shops selling handmade trinkets and souvenirs lined the alleys, their wares a reflection of the local culture. Rows of spice jars, their labels bearing intricate designs, beckoned to curious visitors. Suddenly, the sound of a lute being played filled the air, its melodic strains adding to the marketplace's enchanting ambiance. Traditional dancers twirled and swayed, their graceful movements a sight to behold. Unforgettable experiences awaited around every corner, each sensory detail adding to the marketplace's rich tapestry. Vibrant colors, tantalizing smells, and the infectious energy of the marketplace created an atmosphere unlike any other. Wandering through the marketplace, visitors were transported to a world of sights, sounds, and smells, a world where the senses came alive. Xanadu, a place of wonder and delight, was the marketplace's true name. Yearning for just one more taste, one more smell, visitors left the marketplace with memories they would cherish forever. Zealously, they looked forward to their next visit, the marketplace's magic leaving an indelible mark on their souls.

Beneath the ancient city's towering structures, the bustling marketplace comes alive. Cobblestones, worn smooth by time, weave intricately through the labyrinthine streets. Delightful aromas waft through the air: cardamom, cinnamon, and saffron. Echoes of merchants haggling over prices fill the space. Fruits and vegetables, ripe and succulent, display their vibrant colors.

Glistening spices, heaped high in mounds, entice the senses. Jasmine flowers, their fragrance sweet and heady, add to the intoxicating blend. Kaleidoscopic fabrics, draped artfully over stalls, catch the sunlight. Laughter and lively banter, the sounds of the marketplace, create a joyful symphony.

Melodic tones of traditional instruments, the music of the street performers, add to the sensory experience. Nestled among the stalls, the people, a rich tapestry of cultures and traditions, mingle and converse. Overhead, the sun casts dappled shadows on the uneven ground. Painted signs, bold and colorful, beckon the curious traveler. Quiet moments of reflection are found amidst the bustling scene.

Rays of sunlight, filtering through the canopy of fabric, bathe the marketplace in a warm glow. Sensory delights, a feast for the senses, await the intrepid explorer. Traditional dishes, steaming and savory, tantalize the taste buds. Unique treasures, hidden among the stalls, beckon the collector. Vendors, their faces etched with the lines of time, share their stories and wisdom.

Woven baskets, filled with fragrant spices, are offered for sale. Xylophones, their notes resonating through the air, add to the lively atmosphere. Yearning for adventure, the traveler delves deeper into the marketplace, eager to discover its hidden gems. Zealously, they immerse themselves in the rich tapestry of sights, sounds, and smells.

Sample 2: Evaluating the result of a quadratic function given an input

Prompt: Given that $f(x) = 4x^3 - 9x - 14$, find the value of $f(2)$

Response (LAG on left, Mistral-7B on right):

↳ **Expert(s) called:** Logic, Math

To find the value of the function $f(x)$ at $x = 2$, we substitute x with 2 in the given function and perform the necessary calculations.

First, we substitute $x = 2$ in the function $f(x) = 4x^3 - 9x - 14$:

$$f(2) = 4(2)^3 - 9(2) - 14$$

Next, we calculate the values inside the parentheses:

$$(8) = 32, 9(2) = 18$$

Then, we perform the arithmetic operations:

$$32 - 18 - 14 = 0$$

Therefore, the value of $f(2)$ is 0.

So, the answer is: 0

To find the value of $f(2)$, we need to substitute $x = 2$ into the given function $f(x) = 4x^3 - 9x - 14$.

$$\text{So, } f(2) = 4(2)^3 - 9(2) - 14$$

First, calculate 2 raised to the power of 3:

$$2^3 = 8$$

Now, substitute 8 for x in the function:

$$f(2) = 4(8)^3 - 9(2) - 14$$

Next, calculate 8 cubed:

$$8^3 = 512$$

Now, substitute 8^3 for x^3 in the function:

$$f(2) = 4(512) - 9(2) - 14$$

Calculate 4 times 512:

$$2048$$

Now, calculate -9 times 2:

$$-18$$

Subtract -18 from 2048:

$$2066$$

So, the value of $f(2)$ is 2066

Sample 3: Counting Words Problem

Prompt: Please read the paragraph below and count how many times the words "Amazon", "river", and "you" appear. Please present the results in the format of "word, number of appearances" with each word on a separate line. Sort the lines in order of the number of appearances.

The Amazon, a mesmerizing expanse of nature's wonders, is home to the legendary Amazon River. Flowing through awe-inspiring landscapes like the Amazon rainforest, the river weaves its way through Brazil, Colombia, and Peru, giving life to countless creatures. From the mighty jaguars prowling the Amazon jungle to the vibrant macaws soaring above the canopy, this remarkable region teems with biodiversity. Deep within the river's currents, magnificent pink river dolphins gracefully glide alongside piranhas and electric eels. Along the riverbanks, you'll find bustling cities like Manaus, where the urban meets the wild, and Iquitos, a gateway to the heart of the Amazon rainforest. As you venture further, the Amazon River reveals hidden gems like the captivating Anavilhanas Archipelago, a mosaic of islands brimming with rare species. Embark on an adventure, explore the enchanting Amazon River, and immerse yourself in a world teeming with life and untamed beauty.

Response (LAG on left, Mistral-7B on right):

↳ **Expert(s) called:** Writing

Amazon, 5
river, 10
you, 1

Amazon, 7
river, 5
you, 1

The Amazon is a vast and breathtaking expanse of nature, home to the world-renowned Amazon River. This remarkable river flows through awe-inspiring landscapes such as the Amazon rainforest, traversing countries like Brazil, Colombia, and Peru. The river is the lifeblood of countless creatures, from the powerful jaguars that roam the Amazon jungle to the colorful macaws that soar above the canopy. ...

Here is a breakdown of the word appearances in the paragraph:

- Amazon, 7: The word "Amazon" appears 7 times in the paragraph.
- river, 5: The word "river" appears 5 times in the paragraph.
- you, 1: The word "you" appears only once in the paragraph.

Prompt 2: Please repeat the same task using the words 'the', 'and', and 'to'

Response:

Based on the given paragraph, the words "the," "and," and "to" appear as follows:

* "the" appears 10 times. For example: "The Amazon, a mesmerizing expanse of nature's wonders, is home to the legendary Amazon River."

* "and" appears 3 times. For example: "Flowing through awe-inspiring landscapes like the Amazon rainforest, the river weaves its way through Brazil, Colombia, and Peru."

* "to" appears 2 times. For example: "From the mighty jaguars prowling the Amazon jungle to the vibrant macaws soaring above the canopy."

Therefore, the answer is:

* "the": 10 appearances * "and": 3 appearances * "to": 2 appearances the, 13 and, 3 to, 3

Here is a breakdown of the word appearances in the paragraph:

the, 13: The word "the" appears 13 times in the paragraph.

and, 3: The word "and" appears 3 times in the paragraph.

to, 3: The word "to" appears 3 times in the paragraph.

Sample 4: Logic on Number of Brothers

Prompt: David has three sisters. Each of them has one brother. How many brothers does David have?

Response (LAG on left, Mistral-7B on right):

↳ **Expert(s) called:** Logic, Math

To find the number of brothers David has, we need to consider the number of brothers each of his sisters has. Since David has three sisters, and each of them has one brother, it means David also has one brother for each of his sisters. Therefore, David has a total of three brothers.

Answer: 3

David has one brother. Here's the reasoning behind this:

1. We know that David has three sisters.
2. We are then told that each of his sisters has one brother.
3. Since all of David's sisters share the same brother (David), this means that David is the brother of all three sisters.
4. Therefore, David has one brother.

Prompt 2: If we change the previous question and assume that each sister of David has two brothers, how many brothers would David have?

Response (LAG on left, Mistral-7B on right):

↳ **Expert(s) called:** Logic, Math

To determine the number of brothers David would have if each of his sisters has two brothers, let's analyze the given information:

We know that David has three sisters. Each of his sisters has two brothers.

To find the number of brothers David has, we need to consider the number of brothers each of his sisters has. Since each sister has two brothers, we might initially think that David would have 2 * 3 = 6 brothers. However, this is not the case because David is already counted as one of the brothers for each of his sisters. To find the correct answer, we need to subtract the number of brothers that David is counted as (for his sisters) from the total number of brothers each sister has:

1. Each sister has 2 brothers.
2. However, David is one of those brothers for each sister.
3. So, for each sister, there are 2 brothers - 1 (David) = 1 additional brother.
4. Since David has three sisters, he would have 3 * 1 = 3 additional brothers.
5. In total, David would have 3 brothers (from his sisters) + 3 brothers (additional brothers) = 6 brothers.

Therefore, the answer is: 6

It's important to note that this answer contradicts the initial opinion given, which incorrectly states that David would have 6 brothers based on the number of brothers each of his sisters has without considering that David is already counted as one of his sisters' brothers

In this case, if each of David's sisters has two brothers, it means that David has one sibling brother and another sibling brother. So, David would have two brothers.

A.3 Inference Time Graphs

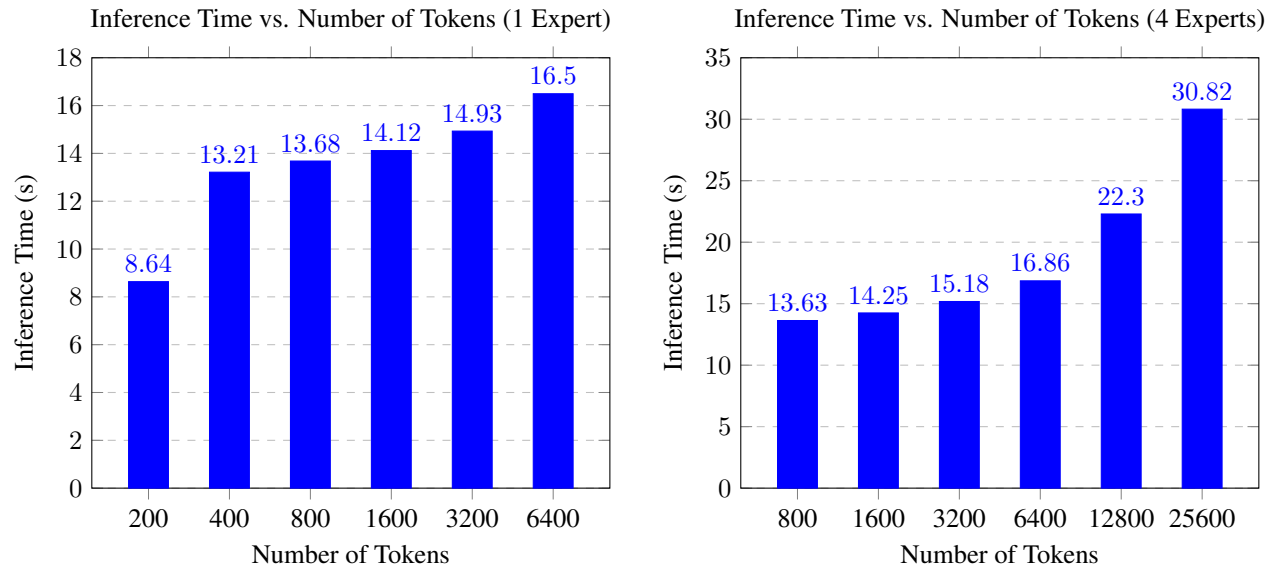


Figure 2: Inference Time vs. Number of Tokens Plots With Varying Number of Experts

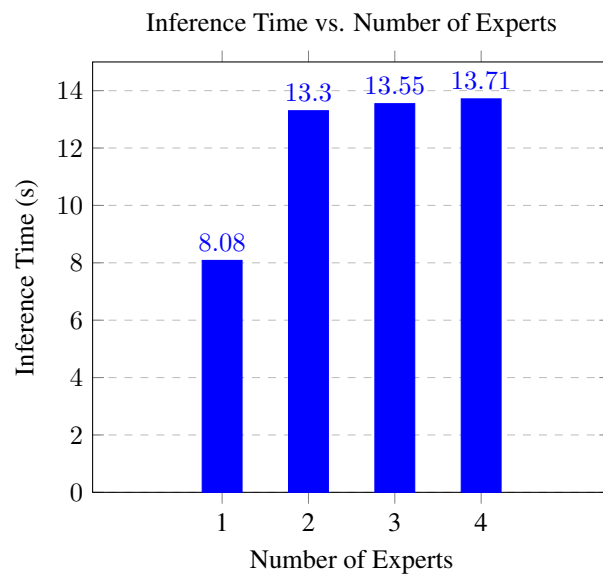


Figure 3: Inference Time vs. Number of Experts.

A.4 Classifier Training Performance

K-fold	Validation Loss	Training Loss
1	0.001488	0.000395
2	0.000755	0.000381
3	0.001062	0.000411
4	0.003054	0.000310
5	0.000969	0.000377

Table 3: K-fold cross validation results of the experts classifier (K=5) trained on 22,910 examples. Number of training epochs is 15 with learning rate 10^{-4} set to random seed 42.