LLM CASCADE WITH MULTI-OBJECTIVE OPTIMAL CONSIDERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Language Models (LLMs) have demonstrated exceptional capabilities in understanding and generating natural language. However, their high deployment costs often pose a barrier to practical applications, especially. Cascading local and server models offers a promising solution to this challenge. While existing studies on LLM cascades have primarily focused on the performance-cost trade-off, real-world scenarios often involve more complex requirements. This paper introduces a novel LLM Cascade strategy with Multi-Objective Optimization, enabling LLM cascades to consider additional objectives (e.g., privacy) and better align with the specific demands of real-world applications while maintaining their original cascading abilities. Extensive experiments on three benchmarks validate the effectiveness and superiority of our approach.

1 INTRODUCTION

As Large Language Models (LLMs) continue to evolve rapidly (Touvron et al., 2023; Achiam et al., 2023; Reid et al., 2024), they are increasingly being integrated into real-world applications, enhancing the intelligence of a wide range of systems. At the same time, mobile devices have become indispensable in everyday life. The emergence of on-device intelligence—such as Apple Intelligence (Gunter et al., 2024) and Gemini Live (Reid et al., 2024)—which embeds LLMs directly into devices for more personalized and intelligent user interactions, is gaining traction but remains relatively underexplored (Xu et al., 2024). A major challenge in this area is the hardware limitations of mobile devices, including constraints on compute power, battery life, and storage capacity. As a result, only smaller LLMs, such as Gemma-2B (Team et al., 2024), can be deployed on these devices, leading to trade-offs in performance compared to larger, more powerful models like Gemini. This raises a critical question for the research community: how can we optimize on-device intelligence given these size constraints? The LLM cascade method presents a solution for this challenge.

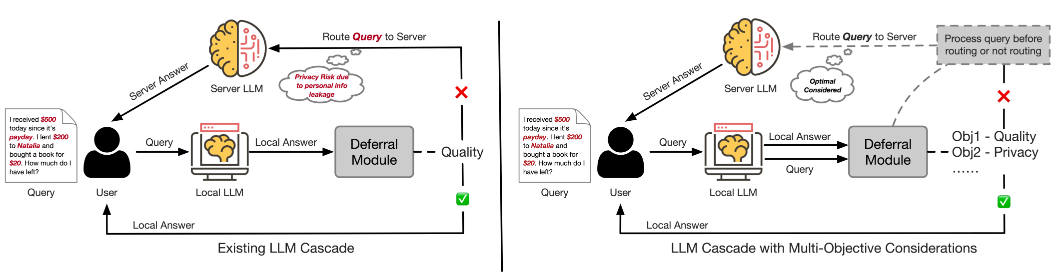


Figure 1: On the right is the existing LLM cascade, where the deferral module makes decisions solely based on the quality of the local answer, potentially leading to privacy leakage. On the left is our proposed LLM cascade with multi-objective considerations, where deferral decisions are more aligned with the needs of real-world applications.

In an LLM cascade system, a query is usually first processed by a smaller, weaker local LLM and is only escalated to a larger, stronger server LLM if the local model's output is deemed insufficient by a deferral module, as shown in Figure 1. This paradigm has garnered significant attention recently

(Chen et al., 2023a; Gupta et al., 2024; Yue et al., 2023; Wang et al., 2024). As larger LLMs are often substantially more expensive than their smaller counterparts (e.g., Gemini-1.5 Pro (Reid et al., 2024) costs up to 35 times more than Gemini-Flash[1]), most existing LLM cascade works focused on the exploration of optimal trade-offs between cost and performance. However, real-world applications can be more complicated and requires the cascade system to make deferral decisions beyond just performance-cost consideration. For instance, privacy concerns may arise if personal data is routed to the server LLM where decisions are made based solely on the local answer's quality, as illustrated in Figure 1. Unfortunately, few studies have explored the LLM cascade with multi-objective consideration. To address this, we propose to endorse multi-objective optimal considerations into the decision making by the LLM cascade system where the deferral module may hesitated to route the user query not only considering the local answer's quality but also with other considerations (e.g., privacy) as depicted in Figure 1.

One key focus of LLM cascade research is the design of deferral criteria, which determine whether a query needs to be routed to the server model. Ideally, the deferral criteria should identify queries that the local LLM is unlikely to handle effectively, sending them to the server to significantly improve performance while keeping costs manageable. Conversely, sending queries that the local LLM can address with high quality to the server can result in unnecessary costs. Intuitively, model confidence could serve as a good indicator, with queries routed to the server when the local model is not confident with its response. For instance, Zhu et al. (2024) explored a self-critique strategy to leverage the local model's intelligence to produce a confidence level in terms of the local answer and make decisions based on the confidence level. However, Jitkrittum et al. (2024) noticed the weakness of confidence-based deferral rule in cases where distribution shifts occur between the training and test datasets. Logit-based methods step further by using the generated token logits of the local answer as features to make deferral decisions. For example, Gupta et al. (2024) found the length bias and token uncertainty problems in cascading by relying on the mean logits and proposed to leverage quantile logits as features to mitigate this problem. Additionally, Wang et al. (2024) introduced cascade-aware training, which incorporates both the local and server LLM's logits into the loss function during local model training, helping the local LLM become more aware of which queries should be deferred to the server. Unfortunately, none of these works explored deferral decision making with respects to other objectives such as privacy. To address this gap, we propose incorporating multi-objective optimization into the LLM cascade system. The key is to enable the local LLM to better understand multi-objective deferral logic, rather than focusing solely on the cost-performance trade-off. Intuitively, we can utilize the in-context learning abilities of the local model by designing appropriate instructional prompts to help it understand the cascade logic with multi-objective considerations (Sordoni et al., 2024; Hartmann et al., 2024). However, this approach is limited by the size and corresponding in-context learning capacity of the local LLM. Another option is training the local LLM to incorporate multi-objective considerations. Instruction tuning has proven highly effective at improving LLM performance across specific tasks, as well as enhancing its ability to follow instructions (Zhao et al., 2024; Chen et al., 2024; Ma et al., 2024), aligning well with our goal of embedding cascade logic into the local model. Moreover, incorporating the more powerful server LLM's capabilities into the customized loss function during local LLM training penalizes the local model for producing high logits associated with poor-quality outputs(Wang et al., 2024). In tandem, we explore both training-based methods (i.e., instruction tuning, loss tuning) and training-free approaches (i.e., prompt engineering) to enable the local LLM to account for multi-objective considerations when deciding whether to invoke the server model. The contributions of this study are three-fold:

- We extend the current focus of LLM cascading beyond the traditional cost-performance trade-off to include multi-objective considerations, better aligning with the needs of real-world applications.
- We explore both training and training-free methods to enable local LLMs to comprehend complex cascade logic with multi-objective considerations.
- Extensive experiments on three benchmarks have validated the necessity and superiority of incorporating multi-objective considerations into LLM cascading, rather than relying solely on cost-performance trade-offs[2].

[1]https://ai.google.dev/pricing

[2]To encourage further explorations by the community, we will open-source our implementations (a copy is attached with this submissions).

2 METHODOLOGY

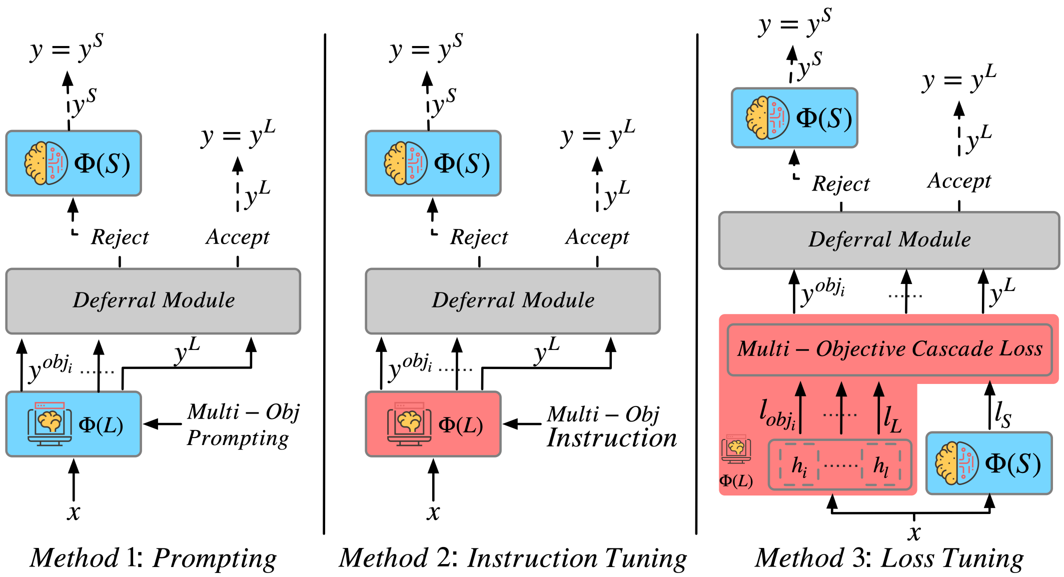


Figure 2: Overview of the proposed methods: $\Phi(L)$ and $\Phi(S)$ represent the local model and server model, respectively. The red box indicates trainable, while the blue box represents frozen. $\Phi(L)$ is tasked with generating responses y^L and y^{obj_i} for both the query x and the multi-objective considerations obj_i. For loss tuning, the generation tasks are handled by different heads h_i, and a combined cascade loss is utilized for tuning.

2.1 PRELIMINARY FORMULATION

Before proceeding, we will first present the preliminary concepts and formulations. Given a local LLM $\Phi(L)$ (smaller and weaker) and a server LLM $\Phi(S)$ (larger and stronger), when a user sends a query x to $\Phi(L)$, the local model generates an initial answer y^L. A deferral module $D(\cdot)$ then determines whether it is necessary to route the query x to $\Phi(S)$. If $D(\cdot)$ accepts y^L, it becomes the final answer y returned to the user. If rejected, the query x is routed to $\Phi(S)$, and the server-generated answer y^S serves as the final response y. Our objective in this study is to enable $\Phi(L)$ to be aware of multi-objective considerations $[obj_1, ..., obj_i]$ while generating y^L. The responses $[y^{obj_1}, ..., y^{obj_i}]$ corresponding to these considerations, along with y^L, can be utilized in $D([y^{obj_1}, ..., y^{obj_i}, y^L])$ to inform decision-making. In this study, we primarily focus on two objectives: privacy and quality. In the following sections, we will illustrate how to incorporate multi-objective considerations into both training methods (instruction tuning and loss tuning) and training-free methods (prompting).

2.2 MULTI-OBJECTIVE PROMPTING

Ideally, the $\Phi(L)$ can be taught multi-objective optimal cascade logic based on its own natural language understanding ability. Efforts have been made to enable the $\Phi(L)$ being aware of the confidence of generated responses via self-critique(Zhu et al., 2024), step-by-step prompting(Zhang & Gao, 2023) etc. We step further on the previous works and include the privacy concern (Hartmann et al., 2024) into prompt design. Specifically, we formulate an instructional prompt[3] which integrates query x and objective considerations (i.e., privacy consideration obj_p) to the $\Phi(L)$ to obtain response $[y^{obj_p}, y^L]$, and these response will further be sent to the $D(\cdot)$ where deferral decisions will be made. Further, we follow Deng et al. (2024)'s work and perform few-shot prompting to better activate the $\Phi(L)$'s in-context learning ability. However, with limited size, the Φ is inadequate[4] to understand the

[3]The prompts used can be seen in the appendix A

[4]Please refer to the appendix B for better understanding over the local llm's weakness.

multi-objective optimal cascade logic relying its own ability and the complicated logic may further hurt its ability to answer user's query and thus training is needed.

2.3 MULTI-OBJECTIVE INSTRUCTION TUNING

Previous studies have demonstrated the effectiveness of instruction tuning in enhancing downstream task performance and improving comprehension of given instructions (Zhu et al., 2024; Zhao et al., 2024; Ma et al., 2024; Li et al., 2023). This ability to understand instructions aligns well with our objective of grasping the deferral logic. Furthermore, the improvements in task performance help mitigate any negative impacts on generating y^L that may arise from producing y^{obj_i} during prompting. Similar to the prompting method, we utilize an instructional prompt that combines a step-by-step instruction with the user query x as input. The labeled text $\hat{y}$ corresponding to x, along with the labeled responses $\hat{y}^{obj_i}$ for the multi-objective considerations, serve as outputs for fine-tuning the model $\Phi(L)$. The responses generated by the tuned model will then be utilized by the deferral module $D(\cdot)$ to determine whether routing to the server model $\Phi(S)$ is necessary.

2.4 MULTI-OBJECTIVE LOSS TUNING

Stepping further over the methods that rely on the local model's intricate understanding ability, recent works have pointed out the superiority of distilling the server llm's ability on downstream tasks into the loss function for tuning the local model(Wang et al., 2024). Intuitively, our assumption is that the server llm is larger and more powerful(Hartmann et al., 2024) in terms of down-stream tasks, and thus the discrepancy between the generations of $\Phi(L)$ and $\Phi(S)$ can somehow be used for $\Phi(L)$ to indicate the confidence level. The larger the discrepancy is, the lower confidence level should the $\Phi(L)$ have. However, to enable $\Phi(L)$ being aware of multi-objective considerations, simply including the distillation loss from $\Phi(S)$ is inadequate. To this end, we decompose the overall task into several sub-tasks and use different heads to handle the different sub-tasks. Namely, given the multi-objective considerations $[obj_1, ..., obj_i]$ and the query x, we leverage multiple llm heads $[h_1, ..., h_i, h_L]$ to handle different considerations and the query. Each head will produce a loss and a distillation loss from $\Phi(S)$ will be optionally added. These losses will then be sent to a weighted-sum function to produce a multi-objective cascade loss for tuning $\Phi(L)$:

$$\begin{gathered} l = \sum_i^n w_i \cdot l_{obj_i} + w_L \cdot l_L + \alpha(t) \cdot w_S \cdot l_S \\ \sum_i^n w_i^n + w_L + w_S = 1, \alpha(t) = H(logit_{y^L}, t) \end{gathered} \tag{1}$$

where w_i denotes the weight for the loss associated with generating response y^{obj_i} for the objective obj_i, w_L is the weight for the loss of generating response y^L for x from $\Phi(L)$ and w_S is the weight for the loss of generating response y^S for x from $\Phi(S)$. n is the number of objectives that need to be considered. α is the factor for controlling if the knowledge from the server LLM $\Phi(S)$ is used depending on a logit threshold t. $H(\cdot, t)$ is a modified Heaviside Step function which returns 0 if $\cdot > t$ else returns 1. In the context of identifying privacy concern, the loss function we utilized for tuning $\Phi(L)$ is:

$$\begin{aligned} l = &- w_p \cdot (\hat{y}^p \cdot log(p_L(y^p|x)) + (1 - \hat{y}^p) \cdot log(1 - p_L(y^p|x)))+ \\ &w_L \cdot log(p_L(y^L|x)) + \alpha(t) \cdot w_S \cdot log(p_S(y^S|x)) \end{aligned} \tag{2}$$

where y^p, $\hat{y}^p$ are the predicted, golden binary predictions for privacy, respectively. Other terms remain the same as in formula 1. By incorporating multi-objective considerations into the loss function for tuning $\Phi(L)$, the model will generate answers with better awareness of these considerations. The corresponding logits of the generated answers by tuned $\Phi(L)$ can then be utilized by the deferral module to inform decision-making.

2.5 DEFERRAL MODULE

All the three methods are studying how to enable the local LLM to be aware of multi-objective considerations while generating the response to the query. And such considerations are presented as the logit distributions of the generated response, for example, higher logit may indicated higher performance and less privacy concern. Deferral module plays a pivotal role in the LLM cascade since it decides which query to send out to the server llm based on the logits. Following previous successes on using different logit (e.g., mean, quantile) of the generated response as the reference to decide if there is a need to route the query to the server LLM(Wang et al., 2024; Jitkrittum et al., 2024; Gupta et al., 2024), we also utilize the logit of generated response as indicators to make the routing decisions. Specifically, given a threshold $t \in (0, 1)$, if the logit of the generated response exceed t then it means the local LLM is confident with its response and no need to route, otherwise route the query x to the server LLM $\Phi(S)$.

3 Experimental Settings

3.1 Datasets

To validate the effectiveness of including multi-objective considerations into LLM cascade, we opt for three benchmarks to test our methods as below, more statistics can be seen in appendix C.2.

GSM8K(Cobbe et al., 2021) is a graduate student mathematical dataset consisting of mathematical questions and corresponding solutions, of which some questions contain personal information for privacy study(Hartmann et al., 2024).

MedQSum(Zekaoui et al., 2023) is a medical related dataset with a focus on summarizing the customer health question. The dataset contains customer health questions and corresponding summaries which contains personal healthcare information.

WMT22(Kocmi et al., 2022) is a sequence-to-sequence translation dataset consisting of source language sentences and corresponding target language sentences.

3.2 Tasks & Metrics

Dataset	Task Type	Privacy?	Measurement
GSM8K	Question Answering	✓	Accuracy, Privacy Leakage
MedQSum	Summarization	✓	ROUGE, Privacy Leakage
WMT22	Translation	✗	ROUGE

Table 1: Details of tasks and measurements.

We evaluate our proposed LLM cascade with multi-objective optimal considerations on three commonly used tasks: Question Answering, Summarization, and Translation, as indicated in Table 1. For datasets involving privacy concerns, we also incorporate the metric of privacy leakage (Hartmann et al., 2024), which calculates the average number of privacy tokens leaked when sending queries to the server LLM (Check more details in appendix C.2). This approach demonstrates the necessity and effectiveness of considering multi-objective factors in the LLM cascade.

3.3 Base Models & Implementation Details

For implementation details, we leverage the Transformers(Wolf et al., 2020) as the base code and conduct extensive experiments with the Gemma models(Team et al., 2024): **Gemma-2B** as the local LLM, **Gemma-7B** as the server LLM. Notably, the server LLM is fine-tuned on all datasets to reach reasonably great performance, of which the server LLM's ability on GSM8K, MedQSum and WMT22 are 52.85%, 61.22% and 36.51%, respectively. We use the AdamW optimizer(Loshchilov & Hutter, 2018; Paszke et al., 2017) with a learning rate of 5e-4 and also a linear warm-up scheduler initialized with 10% of the total training steps as warm-up steps and a weight decay of 1e-4 to avoid over-fitting for all the experiments. The batch size per device is set to 8. All the experiments are conducted on two computation nodes configured with eight 80G H100 GPUs.

4 EXPERIMENTAL RESULTS

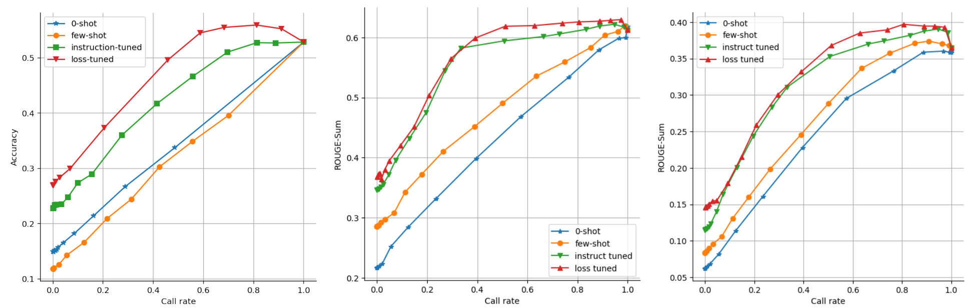
4.1 CASCADE STUDY

Dataset	Metric	%	Prompt Engineering 0-shot	Prompt Engineering few-shot	Instruction Tuning	Loss Tuning
GSM8K	CR		100	100	100	**81.2**
	SCR		28.13	28.13	28.13	**31.75**
	Acc	$\Phi(L)$	14.94	11.83	26.08	26.91
		$\Phi(L) + \Phi(S)$	52.85	52.85	52.85	**55.92**
		vs $\Phi(S)$	-	-	-	↑**3.07**
MedQSum	CR		99.3	96.2	**94.8**	97.3
	SCR		25.98	26.09	26.89	**26.92**
	R-S	$\Phi(L)$	21.69	28.55	34.61	36.77
		$\Phi(L) + \Phi(S)$	61.81	61.97	62.18	**62.95**
		vs $\Phi(S)$	↑0.59	↑0.75	↑0.96	↑**1.73**
WMT22	CR		100	90.9	94.7	**80.6**
	R-S	$\Phi(L)$	6.22	8.36	11.49	14.58
		$\Phi(L) + \Phi(S)$	36.51	37.39	39.04	**39.69**
		vs $\Phi(S)$	-	↑0.88	↑2.53	↑**3.18**

Table 2: Table 2 presents the best cascade performance of $\Phi(L)$ across three benchmarks. CR denotes the call rate, indicating the proportion of queries sent to the server. SCR represents the safe call rate, reflecting the number of queries that are safe (i.e., those sent to the server that do not contain privacy information) among the total sent queries. Acc refers to accuracy, while R-S indicates the ROUGE-Sum score. The symbol ↑ signifies an improvement compared to $\Phi(S)$.

Cascade Performance As shown in Table 2, the cascade approach significantly enhances the performance of the local model $\Phi(L)$, even surpassing the server model $\Phi(S)$. For instance, by routing 81.2% of queries to the server, the loss-tuned $\Phi(L)$ achieves a 55.92% accuracy on the GSM8K dataset, reflecting a 3.07% improvement over $\Phi(S)$. On the MedQSum dataset, improvements in rouge-sum scores of 0.59%, 0.75%, 0.96%, and 1.73% are observed for 0-shot prompting, few-shot prompting, instruction tuning, and loss tuning, respectively, with routing rates of 99.3%, 96.2%, 94.8%, and 97.3%. A similar pattern is noted on the WMT22 dataset, further validating the advantages of LLM cascade for the local model $\Phi(L)$. However, the cost of cascading remains a critical concern in real-world applications. The goal of the cascade is to enhance the local model's performance while maintaining a reasonable server call rate. We observe that training-based methods, such as instruction tuning and loss tuning, yield larger performance gains at lower call rates, indicating the necessity of training the local model to optimize cost-performance trade-offs. In contrast, the performance of training-free methods (e.g., prompt engineering) heavily depends on the server model $\Phi(S)$, rather than the cascade itself. For example, on the GSM8K dataset, the best performance of training-free methods coincides with sending all queries to the server, a pattern is also seen on the WMT22 dataset. This suggests that the local model struggles to identify which queries should be routed to the server. Furthermore, training methods demonstrate a more favorable "safe call" rate compared to training-free methods, highlighting the local model's inability to incorporate multi-objective considerations during cascading. This underscores the need to include multi-objective optimization strategies in LLM cascading.

Performance vs Cost To further understand how the call rate impacts on the local LLM's performance, we set different thresholds t ranging from 0 to 1 with a step of 0.05 to see the performance trends on three datasets. As can be observed in Figure 3, both 0-shot prompting and few-shot prompting exhibit a roughly linear performance improvement as the call rate increases on the GSM8K and MedQSum datasets, suggesting that the prompting methods tend to route queries randomly. However, on the WMT22 dataset, the performance curve for the prompting methods suggests that the local LLM struggles to grasp cascade logic when considering other objectives. In contrast, training methods, especially loss tuning, display a performance increase curve as the number of calls rises, with specific inflection points indicating the optimal trade-off between performance and cost. For instance, when constrained to a 50% call rate, loss tuning demonstrates the best performance, even matching the capabilities of the server LLM, which is quite promising. These

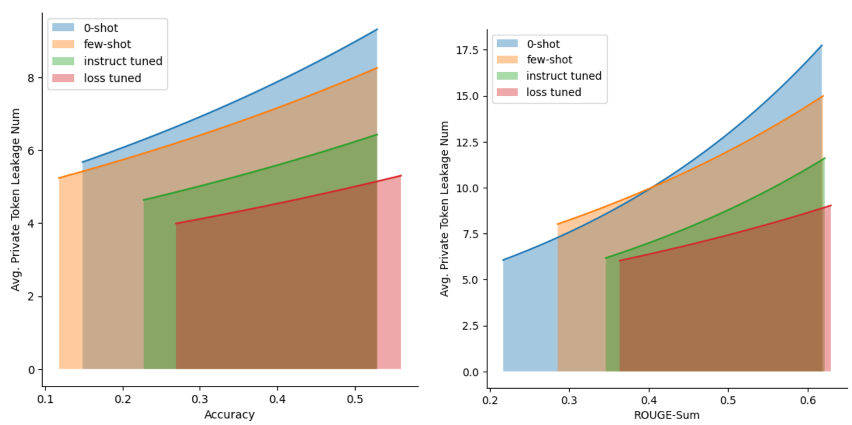


(a) GSM8K (b) MedQSum (c) WMT22

Figure 3: Curves depicting cascade performance versus call rate for different methods across all three datasets: (a) GSM8K, (b) MedQSum, and (c) WMT22.

observations reinforce the necessity for training the local model to effectively understand cascade logic, particularly when incorporating multi-objective considerations.

4.2 Privacy Study



(a) GSM8K (b) MedQSum

Figure 4: The curves illustrating the relationship between the number of privacy tokens leaked and performance are shown for (a) GSM8K and (b) MedQSum.

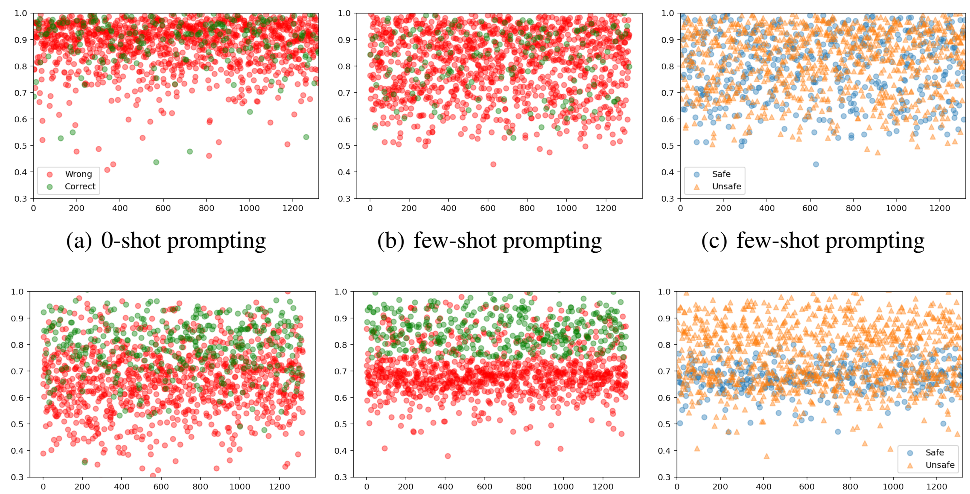
Dataset	Metric	Prompt Engineering		Instruction Tuning	Loss Tuning
		0-shot	few-shot		
GSM8K	precision	0	64.17	82.95	**91.79**
	recall	0	44.20	72.89	**87.24**
MedQSum	precision	48.06	68.85	85.62	**90.10**
	recall	8.44	42.99	68.84	**82.99**

Table 3: Privacy identification by different models.

One of the key contributions of our study is the incorporation of multi-objective optimal considerations (e.g., privacy) into the LLM cascade, distinguishing our work from previous approaches. In this section, we demonstrate how these multi-objective considerations help mitigate privacy concerns within the LLM cascade while preserving its ability to enhance performance.

As can be seen in Figure 4, by incorporating privacy considerations into the cascade, the local LLM tends to route a greater proportion of safe queries to the server, as evidenced by the smaller area under the curves for few-shot prompting compared to the area for zero-shot prompting, even when only a few examples are provided. However, the number of privacy tokens leaked increases at a faster rate compared to the training methods, indicating that relying on the local LLM's in-context ability to identify multiple objectives in cascading is not trustworthy. The privacy identification results presented in Table 3 further validate this claim, as the precision and recall metrics for identifying privacy concerns in queries using prompting methods are not comparable to those of training-based methods. Interestingly, the local LLM $\Phi(L)$ (Gemma-2B) does not recognize personal information, such as names or account details, as privacy concerns, even when explicitly prompted. This oversight could pose risks when the local LLM is applied in real-world financial applications (specific cases can be found in Appendix B). In contrast, the trained $\Phi(L)$ shows significant improvement in identifying private queries, as indicated in Table 3. The gradual increase in performance, illustrated in Figure 4, suggests that the trained $\Phi(L)$ is less likely to route private queries to the server, reinforcing the importance and necessity of incorporating privacy considerations into cascading.

4.3 Logits Distribution Study



(a) 0-shot prompting (b) few-shot prompting (c) few-shot prompting

(d) instruction tuning (e) loss tuning (f) loss tuning

Figure 5: Logits scatter distribution produced by different methods on GSM8K dataset. (e) and (f) are logits for privacy concerns; y-axis is the logits, x-axis is the data index.

To further understand the effectiveness of our proposed LLM cascade with multi-objective considerations, we visualize the logit distributions for both training and training-free methods. As shown in Figure 5 and 8, the logits become more decentralized when a few examples are provided for $\Phi(L)$ to learn the cascade logic, in contrast to 0-shot prompting. Additionally, the signals within the distributions for prompting methods are not distinctly separable, which accounts for the randomness observed in routing queries, as discussed in previous sections. In contrast, training methods demonstrate more distinct distributions, where concentrated red points represent the reflection points noted in Figure 3. This indicates that training-based methods better grasp the cascade logic; answers with higher logits are correlated with more correct responses, suggesting that the trained $\Phi(L)$ is more confident in its correct answers and more likely to route difficult queries to the server. Furthermore, the trained model tends to send fewer unsafe queries to the server, as the logits for unsafe responses are generally higher, making them less likely to be sent. These observations reaffirm the effectiveness and necessity of incorporating multi-objective optimal considerations into cascading, highlighting the superiority of our proposed loss function for training the local LLM compared to existing prompting and instruction tuning methods.

5 Conclusion & Furture Work

In this study, we advance the LLM cascade by incorporating multi-objective optimization, moving beyond existing approaches that primarily emphasize cost-performance trade-offs. This enhancement aligns more closely with the demands of real-world applications. We utilize three methods to assess the necessity and effectiveness of embedding multiple objectives into the cascade. Extensive experiments demonstrate that training is essential for local LLMs to grasp the intricate cascade logic while maintaining their cascading capabilities.

While this work represents the first effort to introduce multi-objective considerations into LLM cascades, future research will explore how the number and complexity of objectives influence the cascade performance of local LLMs. We also aim to develop more sophisticated techniques for integrating these objectives and investigate memory-based methods to sustain favorable cost-performance trade-offs while accommodating a wider array of objectives.

6 Related Work

LLM Cascade Cascading has been extensively studied and applied across various domains due to its ability to enhance system performance in downstream tasks by selecting appropriate models (Hu et al., 2023; Li et al., 2019; Karlos et al., 2016; Viola & Jones, 2001). Recently, this approach has garnered increasing attention for improving the performance of large language models (LLMs). For instance, Agrawal et al. (2024); Xu et al. (2023); Chen et al. (2024) have explored speculative decoding, which leverages a larger and more powerful LLM to verify token-level accuracy during the inference of a smaller LLM, thereby accelerating the overall process. Despite the success of cascading, researchers have observed that larger, more capable LLMs (e.g., GPT-4 (Achiam et al., 2023)) can be expensive, while smaller LLMs (e.g., GPT-2 (Radford et al., 2019)) may not always meet performance requirements. This has led to the emergence of the deferral rule—determining when to invoke the larger LLM—as a critical area of exploration for balancing performance and cost in LLM cascading (Shekhar et al., 2024; Chen et al., 2023a;b). There are two primary approaches to deferral: confidence-based methods and router-based methods. Confidence-based methods leverage the LLM's confidence in its generated answers to inform deferral decisions. Ideally, an LLM exhibits higher confidence for higher-quality answers, and vice versa. A straightforward approach involves asking the LLM to provide a confidence score alongside its answers, invoking the stronger LLM when the score is low (Zhu et al., 2024). Another prevalent method utilizes the logits of generated tokens to represent the LLM's confidence, with recent studies exploring operations on logits, such as mean (Gupta et al., 2024) and quantile (Jitkrittum et al., 2024). Wang et al. (2024) extended this concept by incorporating the logits of the stronger LLM into the loss function for tuning the weaker LLM, enhancing its understanding of the cascade logic and enabling deferral decisions based on logit indicators. In contrast, router-based methods use a routing mechanism to determine whether to invoke the stronger LLM. Typically, the router selects the most suitable LLM for different tasks. Non-predictive routing evaluates the outputs of multiple LLMs to select the best one, but this can be costly due to the need to assess all involved models (Madaan et al., 2023; Lee et al., 2023; Wang et al., 2023). Predictive routing systems, however, employ reward functions that allow the router to anticipate which LLM to select, thus avoiding the latency associated with extensive evaluations (Shnitzer et al., 2023; Šakota et al., 2024; Hari & Thomson, 2023). Nonetheless, router-based methods require prior knowledge of each LLM's capabilities and may incur significant costs when trying to enhance performance, compared to confidence-based methods (Hu et al., 2024b;a). In this study, we adopt confidence-based methods for LLM cascading.

Privacy-preservation Privacy has always been a core concern in LLM research (Kim et al., 2024; Zhang et al., 2024b; Das et al., 2024; Janryd & Johansson, 2024; Feng et al., 2024), particularly for on-device LLM applications (Zhang et al., 2024a; Peng et al., 2024; Yuan et al., 2024). LLMs have been shown to inadvertently reveal sensitive information, such as personal names (Evertz et al., 2024; Kim et al., 2024). To address these privacy issues, Liu et al. (2024a;b;c); Kassem et al. (2023) proposed machine unlearning techniques that enable LLMs to forget sensitive information, thus mitigating the risk of generating harmful or biased content. Another approach is differential privacy, which adds noise to the training data, making it more difficult to identify individual data points (Hartmann et al., 2024). Additionally, Zhang et al. (2024c) suggested using contrastive learning to erase an LLM's memory of user information. While these methods have shown success across diverse user bases, our objective is to enhance the sensitivity of our LLM cascade framework to privacy concerns in single-user settings. To achieve this, we aim to leverage in-context learning and integrate binary privacy identification into the loss function, allowing the local LLM to be more attuned to privacy considerations during the cascading process.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Amey Agrawal, Nitin Kedia, Jayashree Mohan, Ashish Panwar, Nipun Kwatra, Bhargav Gulavani, Ramachandran Ramjee, and Alexey Tumanov. Vidur: A large-scale simulation framework for llm inference. *Proceedings of Machine Learning and Systems*, 6:351–366, 2024.

Boyuan Chen, Mingzhi Zhu, Brendan Dolan-Gavitt, Muhammad Shafique, and Siddharth Garg. Model cascading for code: Reducing inference costs with model cascading for llm based code generation. *arXiv preprint arXiv:2405.15842*, 2024.

Lingjiao Chen, Matei Zaharia, and James Zou. Frugalgpt: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*, 2023a.

Lingjiao Chen, Matei Zaharia, and James Zou. Less is more: Using multiple llms for applications with lower costs. In *Workshop on efficient systems for foundation models@ ICML2023*, 2023b.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems, 2021. *URL https://arxiv. org/abs/2110.14168*, 2021.

Badhan Chandra Das, M Hadi Amini, and Yanzhao Wu. Security and privacy challenges of large language models: A survey. *arXiv preprint arXiv:2402.00888*, 2024.

Keqi Deng, Guangzhi Sun, and Philip C Woodland. Wav2prompt: End-to-end speech prompt generation and tuning for llm in zero and few-shot learning. *arXiv preprint arXiv:2406.00522*, 2024.

Jonathan Evertz, Merlin Chlosta, Lea Schönherr, and Thorsten Eisenhofer. Whispers in the machine: Confidentiality in llm-integrated systems. *arXiv preprint arXiv:2402.06922*, 2024.

Qizhang Feng, Siva Rajesh Kasa, Hyokun Yun, Choon Hui Teo, and Sravan Babu Bodapati. Exposing privacy gaps: Membership inference attack on preference data for llm alignment. *arXiv preprint arXiv:2407.06443*, 2024.

Tom Gunter, Zirui Wang, Chong Wang, Ruoming Pang, Andy Narayanan, Aonan Zhang, Bowen Zhang, Chen Chen, Chung-Cheng Chiu, David Qiu, et al. Apple intelligence foundation language models. *arXiv preprint arXiv:2407.21075*, 2024.

Neha Gupta, Harikrishna Narasimhan, Wittawat Jitkrittum, Ankit Singh Rawat, Aditya Krishna Menon, and Sanjiv Kumar. Language model cascades: Token-level uncertainty and beyond. *arXiv preprint arXiv:2404.10136*, 2024.

Surya Narayanan Hari and Matt Thomson. Tryage: Real-time, intelligent routing of user prompts to large language model. *arXiv preprint arXiv:2308.11601*, 2023.

Florian Hartmann, Duc-Hieu Tran, Peter Kairouz, Victor Cărbune, et al. Can llms get help from other llms without revealing private information? *arXiv preprint arXiv:2404.01041*, 2024.

Qitian Jason Hu, Jacob Bieker, Xiuyu Li, Nan Jiang, Benjamin Keigwin, Gaurav Ranganath, Kurt Keutzer, and Shriyash Kaustubh Upadhyay. Mars: A benchmark for multi-llm algorithmic routing system. In *ICLR 2024 Workshop: How Far Are We From AGI*, 2024a.

Qitian Jason Hu, Jacob Bieker, Xiuyu Li, Nan Jiang, Benjamin Keigwin, Gaurav Ranganath, Kurt Keutzer, and Shriyash Kaustubh Upadhyay. Routerbench: A benchmark for multi-llm routing system. *arXiv preprint arXiv:2403.12031*, 2024b.

Shengkai Hu, Haoyu Wang, and Basel Halak. Cascaded machine learning model based dos attacks detection and classification in noc. In *2023 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5. IEEE, 2023.

Balder Janryd and Tim Johansson. Preventing health data from leaking in a machine learning system: Implementing code analysis with llm and model privacy evaluation testing, 2024.

Wittawat Jitkrittum, Neha Gupta, Aditya K Menon, Harikrishna Narasimhan, Ankit Rawat, and Sanjiv Kumar. When does confidence-based cascade deferral suffice? *Advances in Neural Information Processing Systems*, 36, 2024.

Stamatis Karlos, Nikos Fazakis, Sotiris Kotsiantis, and Kyriakos Sgarbas. A semisupervised cascade classification algorithm. *Applied Computational Intelligence and Soft Computing*, 2016(1): 5919717, 2016.

Aly Kassem, Omar Mahmoud, and Sherif Saad. Preserving privacy through dememorization: An unlearning technique for mitigating memorization risks in language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4360–4379, 2023.

Siwon Kim, Sangdoo Yun, Hwaran Lee, Martin Gubri, Sungroh Yoon, and Seong Joon Oh. Propile: Probing privacy leakage in large language models. *Advances in Neural Information Processing Systems*, 36, 2024.

Tom Kocmi, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Thamme Gowda, Yvette Graham, Roman Grundkiewicz, Barry Haddow, et al. Findings of the 2022 conference on machine translation (wmt22). In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pp. 1–45, 2022.

Chia-Hsuan Lee, Hao Cheng, and Mari Ostendorf. Orchestrallm: Efficient orchestration of language models for dialogue state tracking. *arXiv preprint arXiv:2311.09758*, 2023.

Ang Li, Xue Yang, and Chongyang Zhang. Rethinking classification and localization for cascade r-cnn. *arXiv preprint arXiv:1907.11914*, 2019.

Ming Li, Lichang Chen, Jiuhai Chen, Shwai He, Heng Huang, Jiuxiang Gu, and Tianyi Zhou. Reflection-tuning: Data recycling improves llm instruction-tuning. *arXiv preprint arXiv:2310.11716*, 2023.

Susan Lincke. Complying with hipaa and hitech. In *Information Security Planning: A Practical Approach*, pp. 345–365. Springer, 2024.

Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Xiaojun Xu, Yuguang Yao, Hang Li, Kush R Varshney, et al. Rethinking machine unlearning for large language models. *arXiv preprint arXiv:2402.08787*, 2024a.

Zhenhua Liu, Tong Zhu, Chuanyuan Tan, and Wenliang Chen. Learning to refuse: Towards mitigating privacy risks in llms. *arXiv preprint arXiv:2407.10058*, 2024b.

Zheyuan Liu, Guangyao Dou, Zhaoxuan Tan, Yijun Tian, and Meng Jiang. Towards safer large language models through machine unlearning. *arXiv preprint arXiv:2402.10058*, 2024c.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

Zeyuan Ma, Hongshu Guo, Jiacheng Chen, Guojun Peng, Zhiguang Cao, Yining Ma, and Yue-Jiao Gong. Llamoco: Instruction tuning of large language models for optimization code generation. *arXiv preprint arXiv:2403.01131*, 2024.

Aman Madaan, Pranjal Aggarwal, Ankit Anand, Srividya Pranavi Potharaju, Swaroop Mishra, Pei Zhou, Aditya Gupta, Dheeraj Rajagopal, Karthik Kappaganthu, Yiming Yang, et al. Automix: Automatically mixing language models. *arXiv preprint arXiv:2310.12963*, 2023.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

Dan Peng, Zhihui Fu, and Jun Wang. Pocketllm: Enabling on-device fine-tuning for personalized llms. *arXiv preprint arXiv:2407.01031*, 2024.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Marija Šakota, Maxime Peyrard, and Robert West. Fly-swat or cannon? cost-effective language model choice via meta-modeling. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pp. 606–615, 2024.

Shivanshu Shekhar, Tanishq Dubey, Koyel Mukherjee, Apoorv Saxena, Atharv Tyagi, and Nishanth Kotla. Towards optimizing the costs of llm usage. *arXiv preprint arXiv:2402.01742*, 2024.

Tal Shnitzer, Anthony Ou, Mírian Silva, Kate Soule, Yuekai Sun, Justin Solomon, Neil Thompson, and Mikhail Yurochkin. Large language model routing with benchmark datasets. *arXiv preprint arXiv:2309.15789*, 2023.

Alessandro Sordoni, Eric Yuan, Marc-Alexandre Côté, Matheus Pereira, Adam Trischler, Ziang Xiao, Arian Hosseini, Friederike Niedtner, and Nicolas Le Roux. Joint prompt optimization of stacked llms using variational inference. *Advances in Neural Information Processing Systems*, 36, 2024.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, volume 1, pp. I–I. Ieee, 2001.

Congchao Wang, Sean Augenstein, Keith Rush, Wittawat Jitkrittum, Harikrishna Narasimhan, Ankit Singh Rawat, Aditya Krishna Menon, and Alec Go. Cascade-aware training of language models. *arXiv preprint arXiv:2406.00060*, 2024.

Yiding Wang, Kai Chen, Haisheng Tan, and Kun Guo. Tabi: An efficient multi-level inference system for large language models. In *Proceedings of the Eighteenth European Conference on Computer Systems*, pp. 233–248, 2023.

Albert Webson and Ellie Pavlick. Do prompt-based models really understand the meaning of their prompts? *arXiv preprint arXiv:2109.01247*, 2021.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In Qun Liu and David Schlangen (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL `https://aclanthology.org/2020.emnlp-demos.6`.

Daliang Xu, Wangsong Yin, Xin Jin, Ying Zhang, Shiyun Wei, Mengwei Xu, and Xuanzhe Liu. Llmcad: Fast and scalable on-device large language model inference. *arXiv preprint arXiv:2309.04255*, 2023.

Jiajun Xu, Zhiyuan Li, Wei Chen, Qun Wang, Xin Gao, Qi Cai, and Ziyuan Ling. On-device language models: A comprehensive review. *arXiv preprint arXiv:2409.00088*, 2024.

Yizhen Yuan, Rui Kong, Yuanchun Li, and Yunxin Liu. Wip: An on-device llm-based approach to query privacy protection. In *Proceedings of the Workshop on Edge and Mobile Foundation Models*, pp. 7–9, 2024.

Murong Yue, Jie Zhao, Min Zhang, Liang Du, and Ziyu Yao. Large language model cascades with mixture of thoughts representations for cost-efficient reasoning. *arXiv preprint arXiv:2310.03094*, 2023.

Nour Eddine Zekaoui, Siham Yousfi, Mounia Mikram, and Maryem Rhanoui. Enhancing large language models' utility for medical question-answering: A patient health question summarization approach. In *2023 14th International Conference on Intelligent Systems: Theories and Applications (SITA)*, pp. 1–8. IEEE, 2023.

Shiquan Zhang, Ying Ma, Le Fang, Hong Jia, Simon D'Alfonso, and Vassilis Kostakos. Enabling on-device llms personalization with smartphone sensing. *arXiv preprint arXiv:2407.04418*, 2024a.

Xiaojin Zhang, Yulin Fei, Yan Kang, Wei Chen, Lixin Fan, Hai Jin, and Qiang Yang. No free lunch theorem for privacy-preserving llm inference. *arXiv preprint arXiv:2405.20681*, 2024b.

Xuan Zhang and Wei Gao. Towards llm-based fact verification on news claims with a hierarchical step-by-step prompting method. *arXiv preprint arXiv:2310.00305*, 2023.

Zhaohan Zhang, Ziquan Liu, and Ioannis Patras. Get confused cautiously: Textual sequence memorization erasure with selective entropy maximization. *arXiv preprint arXiv:2408.04983*, 2024c.

Jin Zhao, Chao Liu, Jiaqing Liang, Zhixu Li, and Yanghua Xiao. A novel cascade instruction tuning method for biomedical ner. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 11701–11705. IEEE, 2024.

Yun Zhu, Yinxiao Liu, Felix Stahlberg, Shankar Kumar, Yu-Hui Chen, Liangchen Luo, Lei Shu, Renjie Liu, Jindong Chen, and Lei Meng. Towards an on-device agent for text rewriting. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 2535–2552, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-naacl.163. URL `https://aclanthology.org/2024.findings-naacl.163`.

A PROMPTS

The design of prompts plays a crucial role in activating the LLM's capabilities for downstream tasks. Following the findings of Webson & Pavlick (2021) on prompt design, we first assume a persona for the LLM, then provide task instructions and ask the model to generate outputs in a fixed style. For few-shot prompting, we provide task examples along with their corresponding outputs; details are shown in Fig. 6. Interestingly, we observed that as the number and complexity of tasks in the instructions increased, the model's performance on the target task declined, as demonstrated in Table 2. The prompts presented here yielded the best performance among all the variations we tested.

B PRELIMINARY RESULTS

Metric %	Cascade	Prompt Engineering				Instruction Tuning
		0-shot	1-shot	2-shot	5-shot	
Call Rate		0	70.43	48.98	67.43	42.76
Safe Call Rate		0	2.05	2.94	2.13	**27.61**
Accuracy	✗	14.94	10.08	**11.83**	10.68	**26.08**
	✓	14.94	42.91	37.30	42.61	42.29

Table 4: Preliminary results on GSM8K.

Following the approach of Hartmann et al. (2024), we initially attempted to use self-critique and rely on the in-context learning capabilities of the local LLM to implement the deferral function. Specifically, we instructed the model to handle the task while simultaneously outputting a confidence level, which would determine whether the query should be deferred to the server. However, preliminary results revealed limitations in this design. As shown in Table 4, without examples, the local model

```
gsm8k_prompt = r'''Assume you're a student working on some mathematical
problems. Now, you'll be giving mathematical problems, you need to do two
tasks: a. Check if the question contains personal information (e.g., names
etc.), output Yes or No only;\n
b. Solve this question; \n

Here are some examples:
Question: Hector purchased a container of gumballs. He gave 4 to Todd,
then he gave twice as many as he had given Todd to Alisha, and then he gave
5 less than four times as many to Bobby as he had given to Alisha. If
Hector had 6 gumballs remaining, what is the total number of gumballs that
Hector purchased? \n
Output:
a. Contains Personal Information: Yes
b. Answer: Hector gave to Alisha twice as many as he had given Todd, for a
total of 4*2=<<4*2=8>>8 gumballs, Hector gave 5 less than four times as
many to Bobby as he had given to Alisha, or a total of
(8*4)-5=<<8*4-5=27>>27 gumballs. If Hector had 6 gumballs remaining, he
originally purchased 4+8+27+6=<<4+8+27+6=45>>45 gumballs. #### 45

Question: A garden produced 237 potatoes, 60 fewer cucumbers and twice as
many peppers than the cucumbers. How many vegetables did the garden
produce? \n
Output:
a. Contains Personal Information: No
b. Answer: 237 potatoes + 60 cucumbers + 2*60 peppers = 237 + 60 + 120 =
317 vegetables.<eos>.

Case,
Question: {question}\n
Output:'''
```

(a) GSM8K Prompt

```
meqsum_prompt = r'''
Please summarize the below consumer health question (CHQ) following the
given examples, output the summarized question only:\n

Examples:
CHQ: SUBJECT: who and where to get cetirizine - D
MESSAGE: I need/want to know who manufscturs Cetirizine. My Walmart is
looking for a new supply and are not getting the recent
Summary: Who manufactures cetirizine?

CHQ: who makes bromocriptine
i am wondering what company makes the drug bromocriptine, i need it for a
mass i have on my pituitary gland and the cost just keeps raising. i
cannot ever buy a full prescription because of the price and i was told if
i get a hold of the maker of the drug sometimes they offer coupons or
something to help me afford the medicine. if i buy 10 pills in which i
have to take 2 times a day it costs me 78.00. and that is how i have to buy
them.  thanks.
Summary: Who manufactures bromocriptine?

Case:
CHQ: {chq}\n
Summary:
'''
```

(b) MedQSum Prompt

```
instruction_prompt_wmt = r'''
Assume you're a professional translator, now please translate the
following sentence into English and check if the sentence contains privacy
leakage. Output the translated sentence and privacy check only.\n
Source sentence: {src}\n
Target sentence:
Privacy:
'''
```

(c) WMT22 Prompting

Figure 6: Prompts Used for Prompt Engineering and Instruction Tuning on three datasets.

tends to be overly confident in every generated response. Moreover, even when provided with several examples, the model treats confidence as a classification task, rather than correlating it with the quality of its generated responses. Consequently, we opted to use logits for more effective LLM cascading. Further, as indicated in section A, as the number and the complexity of tasks within the instruction increase, the model tend to have worse performance on the downstream task. As such, we propose to decompose the tasks within the instruction to several tasks and use different heads to handle it for achieving LLM cascade.

C Supplementary Results

C.1 Supplementary Cascade Results

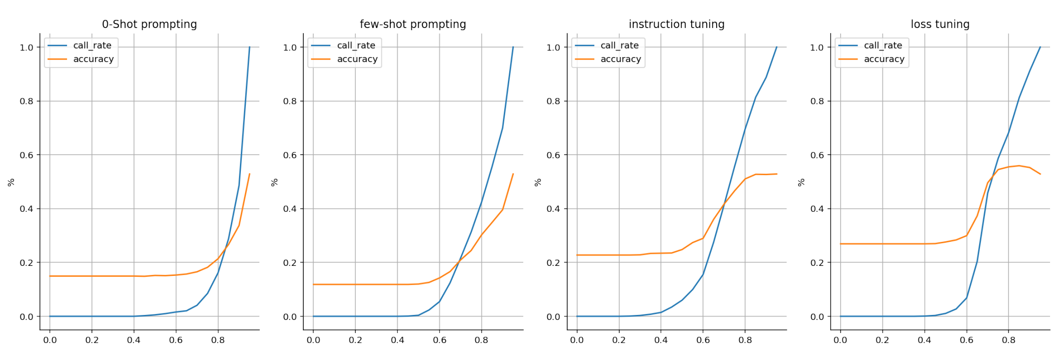
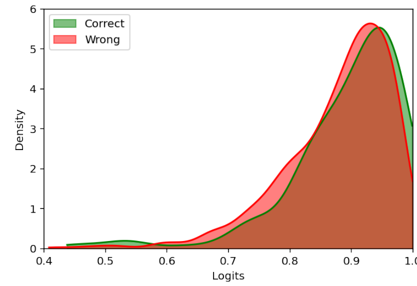
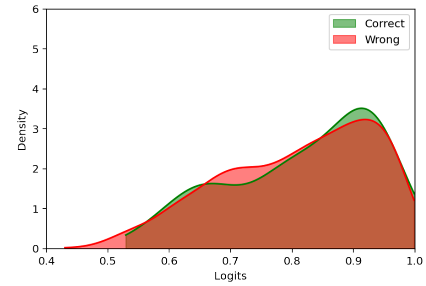


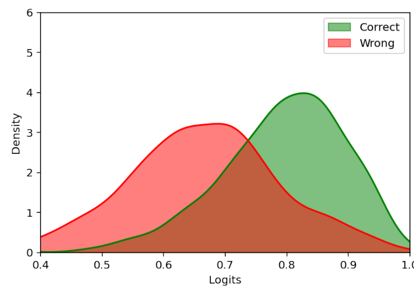
Figure 7: The curve of performance and call rate vs threshold on GSM8K dataset



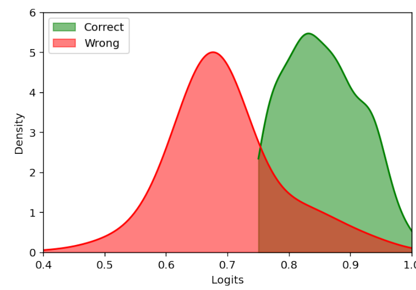
(a) 0-shot prompting



(b) few-shot prompting



(c) instruction tuning



(d) loss tuning

Figure 8: Logits distribution curve by different methods on GSM8K dataset: (a) 0-shot prompting, (b) few-shot prompting, (c) instruction tuning, (d) loss tuning.

As shown in Figure 8, training-based methods have a direct impact on distinguishing between correct and incorrect answers using logits (i.e., the separation between the green and red areas). This aligns with the scatter distribution in Figure 5, further validating the necessity of training in LLM cascading. Additionally, the higher peak in the red area indicates a faster performance improvement, as depicted in Figures 3 and 7. These findings explain the effectiveness and intuition of our approach.

C.2 Datasets

Dataset	Task Type	Avg. Input Length	Avg. Output Length	Avg. Leakage Tokens
GSM8K	Question Answering	52.56	83.60	5.19
MedQSum	Summarization	70.51	11.49	11.27
WMT22	Translation	101.67	95.19	-

Table 5: Statistics of datasets.

Table 5 provides detailed statistics for all datasets. Following the privacy research by Hartmann et al. (2024), we extracted tokens with privacy concerns (e.g., names and other personal identifiers), as the number of such privacy-leakage tokens is critical for evaluating our methods. The extraction was based on PII rules (Kim et al., 2024) and HIPAA regulations (Lincke, 2024), achieving extraction accuracies of 99.1% for GSM8K and 99.7% for MedQSum. A subset of 100 samples was manually verified by a highly educated PhD student, and the p-value score between human and machine extractions was less than 0.05, further validating the effectiveness of our proposed methods.