Firewall Routing: Blocking Leads to Better Hybrid Inference for LLMs

Anonymous ACL submission

Abstract

The rapid advancement of Large Language Models (LLMs) has significantly enhanced performance across various natural language processing (NLP) tasks, yet the high computational costs and latency associated with deploying such models continue to pose critical bottlenecks, limiting their broader applicability. To mitigate these challenges, we propose a dynamic hybrid inference framework, **Firewall Routing**, which efficiently selects between a strong and a weak LLMs based on the complexity of the query. A lightweight routing model is trained to optimize resource allocation by learning from response quality and preventing long-tail queries, which are often too hard to solve by LLMs, from being routed to the stronger model. Moreover, our method incorporates multiple sampling to enhance query evaluation reliability while leveraging **Hard Blocking** and **Soft Blocking** to handle long-tail queries along with refining labels for model selection. Extensive experiments show our method outperforms existing routing strategies by up to 5.29% in APGR, demonstrating state-of-the-art performance across multiple benchmarks.

1 Introduction

In recent years, we have witnessed the rapid advancement of artificial intelligence technologies, particularly the rise of large language models (LLMs) such as ChatGPT, which are reshaping the paradigms of our daily work. These models, often containing billions or even trillions of parameters, generate fluent and contextually appropriate responses, enabling natural interactions without requiring specialized user knowledge (OpenAI et al., 2024; Touvron et al., 2023; Grattafiori et al., 2024). However, such remarkable capabilities come at a significant cost: deploying LLMs demands expensive infrastructure, such as multi-GPU systems with high memory capacity, or incurs higher per-token charges in cloud-based LLM services for more capable models (Yu et al., 2022). Moreover, larger models often introduce higher latency, making them less suitable for real-time or resource-constrained applications. Striking a balance among strong model performance, high efficiency, and economical costs remains an "impossible triangle," yet it is precisely this challenge that drives ongoing research efforts in the field.

Making the "impossible triangle" possible requires a paradigm shift in how we allocate computational resources for language model inference. Extensive experiments have demonstrated that not all tasks require the full power of the largest models (Grattafiori et al., 2024). Simpler queries can often be handled effectively by smaller, lower-cost models without compromising quality, whereas more complex queries leverage the advanced capabilities of larger models. This principle forms the foundation of **Hybrid Inference**.

Given the promising potential, **Hybrid Inference** has garnered significant attention from both academia and industry. Existing strategies can be broadly categorized into two main types: **Cascade** methods (Chen et al., 2023; Gupta et al., 2024; Ramírez et al., 2024), and **Route** methods (Shnitzer et al., 2023; Šakota et al., 2024; Lu et al., 2023; Ong et al., 2024; Ding et al., 2024).

Cascade methods first process all queries using a weaker model. If the weaker model's confidence in its response is low, typically determined through an internal evaluation mechanism, the query is escalated to a stronger model for reprocessing. Although this approach is conceptually straightforward, it has several inherent limitations. On the one hand, evaluating response quality before completion in generative tasks is inherently difficult, leading to unreliable decision-making(Gupta et al., 2024). On the other hand, evaluating response quality after completion brings greatly increased latency. These factors make **Cascade** methods less

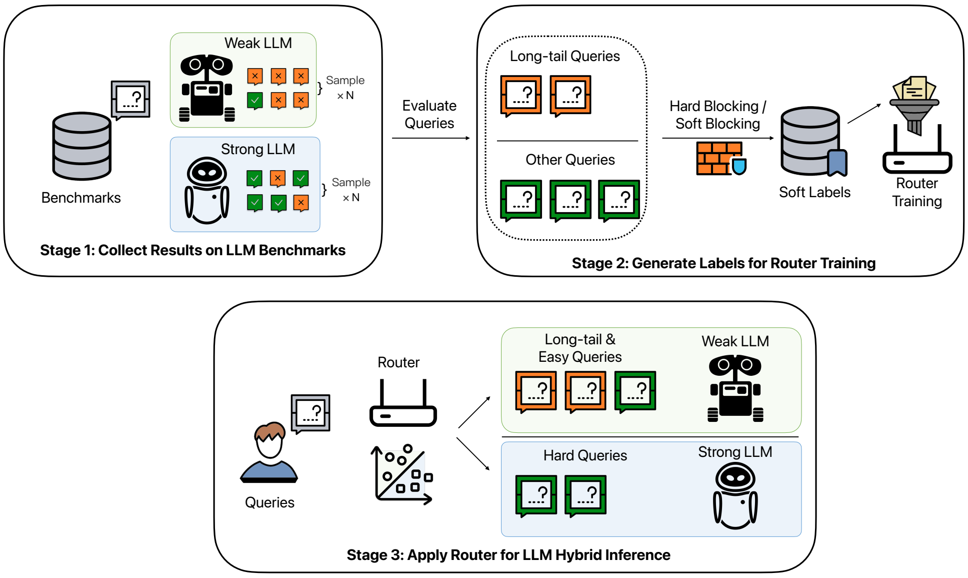

Figure 1: **Firewall Routing** framework for dual-model hybrid inference, comprising a strong model, a weak model, and a router model to balance performance and cost for LLM inference. By blocking long-tail queries from being routed to the strong model, the framework achieves state-of-the-art performance.

efficient in real-world applications.

Motivated by these considerations, we focus on **Route** methods, which leverage a lightweight router model to dynamically allocate queries to the most appropriate LLM under a given configuration. However, existing **Route** methods predominantly rely on collected preference data, which are often limited by strict domain-specific constraints (Shnitzer et al., 2023; Šakota et al., 2024; Lu et al., 2023), or heavily depend on model-generated scores (Ong et al., 2024; Ding et al., 2024). Moreover, these methods often depend on preference data or artificially generated labels based on model scoring. In the context of dual-model hybrid inference, where the strong model generally outperforms the weak model, they fail to address long-tail queries that challenge both models, highlighting opportunities for further optimization.

To address these challenges, we propose **Firewall Routing**, a dual-model hybrid inference system that builds on reliable benchmark results and manages to block long-tail queries, enhancing both performance and efficiency.

Specifically, we propose a novel paradigm for training the router model. Unlike existing methods, our approach utilizes multiple sampling during benchmark evaluations to obtain more accurate estimations of the capabilities of both the strong and weak models. These estimations are then used to construct soft labels for router training. Through mathematical derivations, this paradigm highlights the generality of soft label training in the domain of router optimization and demonstrates that the hard label approach is a specific instance of this broader framework.

To further address the challenge of long-tail queries, we propose two novel approaches—**Hard Blocking** and **Soft Blocking**—designed to effectively manage these cases. **Hard Blocking** utilizes statistical information to identify long-tail queries and assigns them the label "route to the weak model," minimizing unnecessary computational overhead. In contrast, **Soft Blocking** leverages the **Pass Rate** (pass@1) to generate refined soft labels with more precise routing conditions, further reducing computational inefficiencies.

To summarize, we make the following contributions:

1. We propose a novel router training paradigm leveraging multiple sampling to generate soft labels, which generalizes router optimization and demonstrates hard label training as a specific case within this framework.

2. We propose **Hard Blocking** and **Soft Blocking** as automated mechanisms to enable our approach to overcome the challenges associated with long-tail queries.

3. We validate our approach through extensive experiments across diverse configurations.

2 Related Works

Hybrid Inference balances response quality and inference cost by dynamically selecting models based on task complexity. For image classification, Kag et al. (2023) explored joint training of a small model, a large model, and a router, while in NLP tasks, the Tryage architecture (Hari and Thomson, 2023) employed a joint-trained router to optimize performance across domains. However, for LLMs, joint training is computationally expensive and deviates from the pre-training paradigm, leading to two main approaches: **Cascade Methods** and **Route Methods**.

Cascade Methods first query a weaker model and escalate the request to a stronger model only when necessary. FrugalGPT (Chen et al., 2023) estimates response confidence using an LLM-based heuristic to decide whether a query should be forwarded to a larger model. Similarly, Gupta et al. (2024) proposed a confidence estimation method based on the conditional probability of the generated response, serving as a reliability metric. By assessing the correctness of the weaker model's responses, these methods effectively reduce the number of strong model invocations while maintaining high response quality. However, this approach introduces significant response time overhead, as the weaker model must first generate an output before determining whether escalation is required.

Margin Sampling (Ramírez et al., 2024) is a different cascade approach without introducing extra response time. Only when the probability difference between the top two predicted tokens is small at the beginning of generation, indicating uncertainty, is the query escalated to the strong model.

Route Methods introduce a router model to determine which model should handle a given query. Some works focus on selecting the most effective model from a pool of equally scaled LLMs. For example, TensorOpera Router(Stripelis et al., 2024) proposes a complex and large-scale system that assigns each task to a specialized expert LLM. GraphRouter(Feng et al., 2025) builds upon existing routing strategies and combines multiple types of routers within a unified framework to jointly optimize both efficiency and performance in hybrid inference. Shnitzer et al. (2023) frame routing as an out-of-distribution (OOD) detection problem, predicting model response correctness using k-nearest embedded queries. Similarly, Šakota et al. (2024) train a model to determine whether a query can be correctly answered, incorporating a special token to indicate which LLM should be used. Lu et al. (2023) distill a reward model to predict the optimal expert LLM for a given query.

Many recent works focus on dual-model hybrid inference systems. For instance, RouteLLM (Ong et al., 2024) uses preference pairs from multiple LLMs in Chatbot Arena to train a Bradley-Terry model (Bradley and Terry, 1952) as the router. Hybrid LLM (Ding et al., 2024) derives Win Rates for queries through a biased comparison of response BARTScores, creating a desired label distribution to train the router. These approaches highlight the potential for training routers with more reliable evidence, such as pass@k (Chen et al., 2021), to improve model selection.

3 Method

3.1 Router Training Criterions

3.1.1 Train with Hard Label

Early works on building up hybrid inference systems usually train a system with the router model as a whole, where the router model learns how to route under a fixed configuration(Kag et al., 2023). Due to the high training costs associated with large-scale models, most works in LLM hybrid inference only train the router model.

In existing evaluation frameworks for large language models, generative tasks typically follow a greedy decoding paradigm, where the model outputs the token with the highest probability while disregarding alternative token possibilities. Based on this setting, existing methods (Ding et al., 2024) adopt a "Hard Label" approach for router training.

Specifically, for a single query $x_i \in Q$, let $S(x_i)$ and $W(x_i)$ represent the responses generated by the strong model S and the weak model W, respectively, using greedy decoding. The correctness of these responses is denoted as $\delta(S(x_i))$ and $\delta(W(x_i))$, where $\delta(\cdot) \in \{0, 1\}$, with 1 indicating a correct response and 0 indicating an incorrect one. The decision on *whether to route the query to the weak model* is determined by the label y_i, defined as $y_i := \mathbb{I}[\delta(S(x_i)) \leq \delta(W(x_i))]$. Here, $y_i = 1$ implies the weak model is capable of performing at least as well as the strong model for query x_i, and thus the query should be routed to the weak model.

The hard-label router is trained by minimizing the binary cross-entropy loss:

$$\mathcal{L}(\theta) = -\frac{1}{|Q|}\sum_{i=1}^{|Q|}((1-y_i)log(1-p_\theta(x_i)) + y_i log(p_\theta(x_i))), \quad (1)$$

where $p_\theta(x)$ is output of router θ toward query x, where larger $p_\theta(x)$ indicates that the queries should more likely to be routed to the weak model.

The hard label approach is limited by its inability to account for the inherent variability in the responses of large models, thereby restricting the router's ability to make fine-grained decisions. This limitation becomes particularly apparent in scenarios where the smaller model's performance is often comparable to that of the larger model.

3.1.2 Train with Soft Label

To more objectively reflect the performance of large models, existing evaluations often involve multiple sampling of model outputs. Inspired by this approach, we extend our approach by incorporating multiple sampling, which allows us to evaluate the models more thoroughly and account for response variability. This enhancement aims to improve the robustness and efficiency of the routing decisions in our hybrid inference framework.

Specifically, for a single query x_i , let $S^1(x_i), \ldots, S^n(x_i)$ and $W^1(x_i), \ldots, W^n(x_i)$ denote the responses generated by the strong model S and the weak model W over n sampling iterations. The correctness of these responses is represented by $\delta(S^j(x_i))$ and $\delta(W^j(x_i))$, where $\delta(\cdot) \in \{0, 1\}$, with 1 indicating a correct response and 0 indicating an incorrect one. Each sampling iteration produces a noisy observation of y_i, denoted as $y_i^j = \mathbb{I}[\delta(S^j(x_i)) \le \delta(W^j(x_i))]$. In this setting, x_i is associated with n data pairs in the training set, denoted as $(x_i, y_i^1), (x_i, y_i^2), \ldots, (x_i, y_i^n)$.

Using this data, the router can still be trained with a hard label-based objective. However, this approach presents two significant challenges: first, the training cost scales proportionally with the number of sampling attempts n; second, a single input can correspond to varying labels, potentially misleading the router's behavior.

Thus, we introduce the concept of the weak-to-strong **Win Rate**, defined as $r_i := \frac{1}{n}\sum_{j=1}^{n} y_i^j$, which represents the probability that the weak model matches or exceeds the performance of the strong model. Furthermore, we demonstrate that optimization objectives based on **Win Rate** exhibit greater generality for router training. Notably, hard label training inherently captures the concept of **Win Rate**, which can be expressed in the following form:

$$\begin{aligned}
\mathcal{L}(\theta) &= -\frac{1}{n|Q|}\sum_{i=1}^{|Q|}\sum_{j=1}^{n}((1-y_i^j)log(1-p_\theta(x_i)) \\
&\quad + y_i^j log(p_\theta(x_i))) \\
&= -\frac{1}{n|Q|}\sum_{i=1}^{|Q|}((n-\sum_{j=1}^{n} y_i^j)log(1-p_\theta(x_i)) \\
&\quad + (\sum_{j=1}^{n} y_i^j)log(p_\theta(x_i))) \\
&= -\frac{1}{|Q|}\sum_{i=1}^{|Q|}((1-r_i)log(1-p_\theta(x_i)) \\
&\quad + r_i log(p_\theta(x_i))). \quad (2)
\end{aligned}$$

Here, $p_\theta(x)$ represents the output of the router θ for the query x, where a larger $p_\theta(x)$ indicates a higher likelihood that the query should be routed to the weak model.

This formulation naturally motivates the exploration of more refined soft labels that capture the nuanced behavior of large models through their win rates. In contrast to existing approaches (Ding et al., 2024), which adopt probabilistic label construction heuristically, we ground the transition from hard to soft labels in a principled formulation. This perspective sets the stage for our subsequent investigation into soft-label training strategies, where we aim to better leverage signals for more effective routing.

3.2 Blocking Long-tail Queries

Even for large models, there are instances where, despite multiple sampling attempts n, the model is still unable to resolve certain long-tail queries. This limitation arises from the inherent complexity and ambiguity in some queries, which even powerful models may struggle to address consistently, regardless of the number of samples taken. Consequently, such cases highlight the need for more sophisticated handling of long-tail queries in hybrid inference systems.

3.2.1 Hard Blocking

To automatically identify long-tail queries, we introduce multiple sample **Pass Rate** (pass@k when k=1) from Chen et al. (2021)'s work to substitute single sample correctness. For a single query $x_i \in$

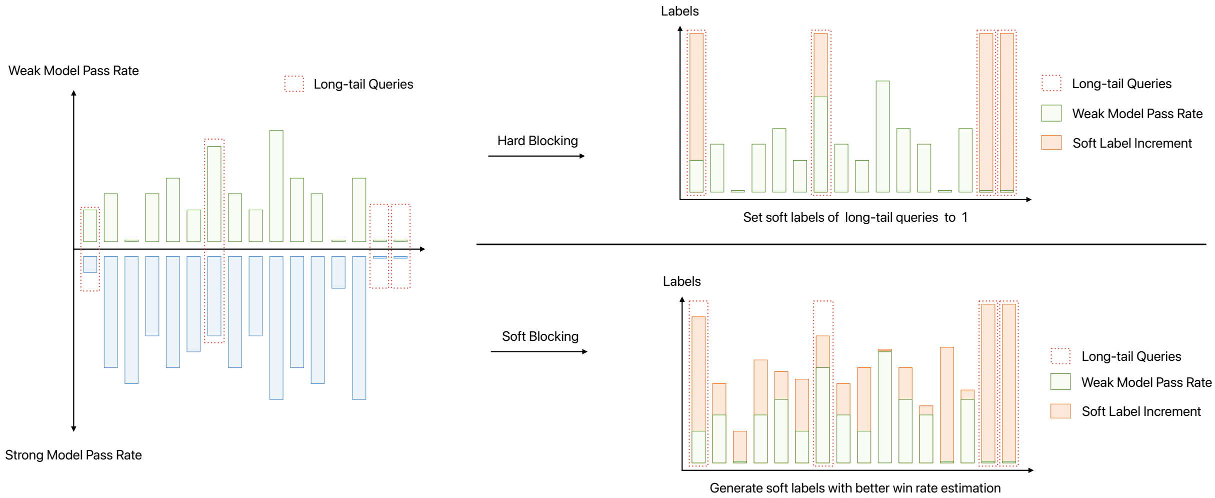

Figure 2: Hard Blocking and Soft Blocking facilitate the automatic handling of long-tail queries by generating reliable soft labels for router training. Queries assigned larger soft label values are more likely to be routed to the weak model.

Q with n sampled responses $R^1(x_i), ..., R^n(x_i)$ from model R, **Pass Rate** is defined as the average correctness of these responses:

$$pr(x_i) := \frac{1}{n}\sum_{j=1}^{n}\delta(R^j(x_i)). \tag{3}$$

We are able to split queries into two sets, Q_u and $Q_s = Q - Q_u$, representing long-tail and other queries, satisfying:

$$\begin{aligned} &\forall x^u \in Q_u, pr_s(x^u) \leq pr_w(x^u), \\ &\forall x^s \in Q_s, pr_s(x^s) > pr_w(x^s), \end{aligned} \tag{4}$$

here we identify long-tail queries as those on which the weak model outperform the strong model, which is also known as the **complementary behaviour** between LLMs (Chen et al., 2023).

By addressing long-tail queries through routing them to the weak model, the decision to route other queries similarly hinges entirely on the weak model's capability to handle these queries effectively:

$$label_i = \begin{cases} pr_w(x_i), x_i \in Q^s, \\ 1, x_i \in Q^u, \end{cases} \tag{5}$$

where $label_i$ is the soft label used in router training to substitute r_i in Eq.2.

To further reduce the cost associated with label collection in this method, it is also possible to split Q_u and Q_s using only the strong model's greedy-decoding responses, subject to the following restrictions:

$$\begin{aligned} &\forall x^u \in Q_u, \delta(S(x^u)) = 0, \\ &\forall x^s \in Q_s, \delta(S(x^s)) = 1. \end{aligned} \tag{6}$$

3.2.2 Soft Blocking

A closer examination of Eq.2 and the concept of the Pass Rate reveals that r_i functions as a noisy indicator, capturing the behaviors of the two models when processing the same query. A key insight is that the performance of the strong model is independent of whether the weak model answers correctly. Instead of treating the two models' performances as a joint distribution, we can more effectively leverage the distributional information obtained from multiple samplings. By treating the two independent events separately, we can more accurately estimate r_i through **Pass Rate**. To maximize the use of this information, we define the joint event for routing the query to the weak model by combining two conditions: *the weak model is correct* and *even if the weak model is incorrect, the strong model also fails*. This method allows us to offer a more refined and informative estimate of overall performance:

$$\begin{aligned} label_i &= pr_w(x_i) + (1 - pr_w(x_i))(1 - pr_s(x_i)) \\ &= 1 - (1 - pr_w(x_i))pr_s(x_i), \end{aligned} \tag{7}$$

where $label_i$ is the soft label used in router training to substitute r_i in Eq.2, and $label_i$ is the observed frequency that the strong model fail to overperform the weak model. As shown in Fig 2, this method also works well with long-tail queries.

Datasets	TriviaQA				GSM8K				HumanEval			
Metrics	APGR↑	Pass Rate↑			APGR↑	Pass Rate↑			APGR↑	Pass Rate↑		
		20%	50%	80%		20%	50%	80%		20%	50%	80%
Linear Interpolation	50.00	19.95	34.97	49.99	50.00	11.69	17.16	22.62	50.00	8.12	10.10	12.08
Hybrid LLM	49.17	18.98	34.38	49.99	62.08	14.38	20.75	24.79	51.94	8.10	10.50	**12.35**
RouteLLM (MF)	51.58	20.69	36.27	51.09	49.39	11.37	17.13	22.27	47.08	7.81	9.95	12.23
Margin Sampling	50.02	19.78	35.01	50.15	46.01	10.85	16.02	21.70	44.88	7.74	9.81	11.53
Ours (Hard Block)	53.16	22.09	37.85	50.96	**67.37**	**16.34**	**22.46**	24.67	**54.36**	8.17	**10.77**	12.27
Ours (Soft Block)	**55.00**	**22.48**	**38.99**	**52.88**	66.65	15.53	22.15	**25.32**	53.13	**8.23**	10.69	12.27

Table 1: Zero-shot performance of different methods across selected datasets. The weak model is Llama3.2-1B, and the strong model is Llama3.1-70B. Linear Interpolation represents the combined performance of the two LLMs to simulate random routing. **Bolded values** indicate the best-evaluated results. Note that Pass Rates at 0% and 100% correspond to using only the weak or strong model, respectively, and thus remain identical across all methods.

4 Experiments

4.1 Settings

Datasets We evaluate our method on generative tasks commonly used to assess the capabilities of large language models (LLMs). Following prior work (Ong et al., 2024), we adopt three benchmarks: TriviaQA (Joshi et al., 2017) for commonsense question answering, GSM8K (Cobbe et al., 2021) for mathematical reasoning, and HumanEval (Chen et al., 2021) for code generation. The training set is constructed from the training splits of TriviaQA and GSM8K, totaling over 68K examples, while HumanEval is used solely as a test set to evaluate the router's out-of-domain (OOD) generalization capability. Across all datasets, we use a simple zero-shot prompt format without system prompts, where each input is structured as: `"Question: {question}\nAnswer:"`. Generating training labels in such generative settings is computationally intensive, as it requires producing $n = 32$ response samples per query. In our setup, these labels are derived from LLaMA3.2-1B, 3B, and LLaMA3.1-70B models, making the data collection process particularly expensive, which also limits us to conduct experiments on more LLMs.

Models In this study, we utilize two large language models (LLMs) from the Llama family (Grattafiori et al., 2024) for our experiments: Llama3.2-1B serves as the weak model, while Llama3.1-70B is employed as the strong model for training the router. Furthermore, to assess the generalizability of the trained router, we test it on an alternative model pair, substituting Llama3.2-3B as the weak model.

Routers Aligned with prior studies (Ding et al., 2024), we adopt DeBERTa-v3-large (He et al., 2023) as the backbone for the router model, augmented with an additional linear layer to output the probability of assigning each query to either the weak or strong model. The router is trained for 10 epochs using the designated loss function, and the final evaluation is based on the checkpoint that achieves the best performance on the validation set. Since our configuration largely follows that of prior works, and it is worth noting that, compared to the strong model (LLaMA3.1-70B), the computational cost of both the weak model and the router is negligible. As a result, the overall latency, and the reciprocal of speedup rate closely approximate the routing ratio. Therefore, we report these values in Appendix A for completeness.

Baselines We compare our approach with several state-of-the-art methods, including Hybrid LLM (Ding et al., 2024), RouteLLM (Ong et al., 2024), and Margin Sampling (Ramírez et al., 2024). For Hybrid LLM, we reproduce the best methodology and hyperparameter selection as outlined in the original paper, "the probabilistic router with data transformation." The determistic variant is reproduced as **Hard Label** in Table 3. For RouteLLM, we employ the best practices with downloadable pre-trained weights, utilizing Matrix Factorization (MF) with OpenAI's text-embedding-3-small to embed the queries. For Margin Sampling, we treat it as a train-free baseline. We also adopt Random Routing (i.e., linear interpolation) as a baseline, which approximates the expected performance between always routing to the weak or strong model. Full results are reported in Appendix A.

Metrics We evaluate the performance of the hybrid inference system using the **Pass Rate**, defined as pass@1 (Chen et al., 2021), based on $n = 32$ sampling iterations. The system's performance is

Datasets	TriviaQA				GSM8K				HumanEval			
Metrics	APGR↑	Pass Rate↑			APGR↑	Pass Rate↑			APGR↑	Pass Rate↑		
		20%	50%	80%		20%	50%	80%		20%	50%	80%
Linear Interpolation	50.00%	25.75	38.59	51.44	50.00%	12.60	17.72	22.85	50.00%	10.27	11.44	12.61
Hybrid LLM	49.15%	24.86	38.04	51.50	61.09%	14.38	20.75	24.79	51.94%	**10.42**	11.47	12.63
RouteLLM (MF)	51.22%	26.14	39.45	52.28	49.11%	15.11	20.72	24.66	50.33%	10.10	11.26	12.65
Margin Sampling	51.21%	26.07	39.15	52.49	43.82%	11.71	16.11	21.42	44.88%	9.97	11.41	12.42
Ours (Hard Block)	53.29%	27.61	41.15	52.28	**65.97%**	**16.68**	**22.30**	24.54	50.86%	10.04	11.62	12.63
Ours (Soft Block)	**55.38%**	**27.91**	**42.28**	**54.28**	65.48%	16.01	22.04	**25.24**	**52.37%**	10.33	**11.72**	**12.80**

Table 2: Zero-shot performance of various methods across selected datasets, generalizing to different model pairs. Trained on the hybrid inference system of Llama3.2-1B and Llama3.1-70B, and evaluated on the hybrid inference system of Llama3.2-3B and Llama3.1-70B. Linear Interpolation simulates random routing by combining the performance of the two LLMs. **Bolded values** indicate the best-evaluated results. Note that Pass Rates at 0% and 100% correspond to using only the weak or strong model, respectively, and thus remain identical across all methods.

reported at different proportions (20%, 50%, 80%) of queries routed to the strong model. Furthermore, we incorporate the **Average Performance Gap Recovered (APGR)** metric from RouteLLM (Ong et al., 2024), which quantifies the system's ability to recover the performance gap between two LLMs. APGR is computed across a range of routing ratios (0%, 10%, . . . , 100%) and yields values between 0% and 100%, reflecting how much of the performance discrepancy is resolved through dynamic routing. While APGR serves as a robust and interpretable metric, another metric introduced in the same work—**Call-Performance Threshold (CPT)**—is less reliable. In particular, closing the bottom-n% performance gap is substantially easier than the top-n%, making CPT prone to inflation. Although our method still achieves state-of-the-art CPT results, we include this metric only in Appendix B. It is also important to emphasize that **existing route methods are commonly evaluated on a per-task basis**, often using task-specific thresholds and evaluation metrics.

4.2 Main Results

4.2.1 Overall Performance

Table 1 summarizes the overall performance of various routing methods within a hybrid inference system utilizing Llama3.2-1B and Llama3.1-70B. Methods achieving higher APGR also exhibit improved performance across different proportions of queries routed to the strong model. Our proposed methods outperform existing approaches, with a notable improvement of **3.72% on TriviaQA, 5.29% on GSM8K, and 2.42% on HumanEval**, demonstrating robustness across diverse query scenarios. Additional visualizations of these results are provided in Appendix D.

On TriviaQA, Soft Blocking achieves the best performance, with Hard Blocking also outperforming all baselines. Hybrid LLM performs poorly, likely due to its reliance on BartScore-based win rates, which—according to Appendix C—do not reliably reflect response quality across datasets. Other methods consistently outperform random routing, confirming their effectiveness. On GSM8K and HumanEval, baseline methods show consistent trends—either strong or weak on both—whereas our methods consistently yield state-of-the-art results. Despite using the same training data, Hybrid LLM underperforms due to its less effective objective formulation.

RouteLLM and Margin Sampling struggle to generalize. For RouteLLM, the drop may stem from domain shifts and OOD routing issues; its original paper also reports weak GSM8K performance without additional data, which we could not obtain. Margin Sampling also suffers on reasoning tasks like math, where its core assumption—based on output margin—is challenged by the presence of multiple valid solutions, especially when using smaller LLMs.

4.2.2 Generalizing to Different Model Pairs

Generalizing to different model pairs is not a mandatory property for router models. However, considering that the training cost of a router is often dominated not by the router architecture itself, but by the label collection process—which can be computationally expensive—it is desirable to examine whether the router can generalize across similar model combinations. In this work, we explore a mild generalization setting by replacing the weak model with a nearby alternative (e.g., swapping the

Datasets	TriviaQA				GSM8K				HumanEval				Sample Cost
Metrics	APGR↑	Pass Rate↑			APGR↑	Pass Rate↑			APGR↑	Pass Rate↑			
		20%	50%	80%		20%	50%	80%		20%	50%	80%	
Weak Model Pass Rate	50.96	20.00	35.88	50.94	51.17	12.18	17.48	22.63	53.42	8.19	10.56	12.31	32+0
Strong Model Pass Rate	54.31	21.44	38.53	**53.28**	65.98	15.24	21.82	**25.35**	49.16	7.87	9.95	12.27	0+32
Hard Label	52.05	20.80	36.51	51.51	63.26	14.68	21.02	24.91	50.63	8.61	10.12	11.97	32+32
Hard Blocking w/o SMS	54.48	22.23	38.62	52.61	63.43	14.95	21.09	25.05	**54.44**	**8.86**	10.48	**12.60**	32+1
Hard Block	53.16	22.09	37.85	50.96	**67.37**	**16.34**	**22.46**	24.67	54.36	8.17	**10.77**	12.27	32+32
Soft Block	**55.00**	**22.48**	**38.99**	52.88	66.65	15.53	22.15	25.32	53.13	8.23	10.69	12.27	32+32

Table 3: Zero-shot performance of various label designs across selected datasets. All models were trained and evaluated using Llama3.2-1B as the weak model and Llama3.1-70B as the strong model. **Bolded values** indicate the best results. Note that Pass Rates at 0% and 100% correspond to using only the weak or strong model, respectively, and thus remain consistent across all methods. **Sample Cost** denotes the number of sampling process required to get **Pass Rate** for a single training example, represented as $\{a + b\}$, where a is the number of samples drawn from the weak model and b from the strong model.

Datasets	TriviaQA	GSM8K	HumanEval
Metrics	APGR↑		
Hard Blocking (Causal)	51.78%	57.16%	54.44%
Hard Blocking (DeBERTa)	53.16%	**67.37%**	54.36%
Soft Blocking (Causal)	52.44%	58.55%	**55.31%**
Soft Blocking (DeBERTa)	**55.00%**	66.65%	53.13%

Table 4: Zero-shot performance of different backbone models (DeBERTa-v3-large, Llama3.2-1B) across selected datasets. Trained and evaluated within the hybrid inference system of Llama3.2-1B and Llama3.1-70B. **Bolded values** indicate the best-evaluated results.

1B model with a 3B variant).

In Table 2, we evaluate the performance of the hybrid inference system configured with Llama3.2-3B and Llama3.1-70B, utilizing routers trained in prior experiments without any additional retraining. Our methods, particularly Soft Blocking, consistently demonstrate superior performance in this configuration, achieving an APGR improvement of **4.16% on TriviaQA, 4.88% on GSM8K, and 0.43% on HumanEval**, which highlights the generalization capability of our method, where routers trained on one model pair exhibit consistent performance when applied to another, confirming its adaptability. Additional visualizations of these results are provided in Appendix D.

4.3 Ablation Study

4.3.1 Router Models

An alternative choice for the router model backbone is causal LLMs (Ong et al., 2024). However, we argue that using a router model larger than the weak model incurs unnecessary computational costs and impacts response time. As a result, we train the weak model as the router for comparison. As shown in Table 4, DeBERTa-v3-large (300M) outperforms Llama3.2-1B, despite its smaller size, demonstrating better performance. Llama3.2-1B performs better on HumanEval, indicating potential generalization ability.

4.3.2 Label Designs

We conduct an ablation study on various label strategies, as shown in Table 3. Training the router solely with the weak model's pass rate yields performance only marginally above random routing, indicating that the weak model alone provides limited routing signal. In contrast, using the strong model's pass rate leads to better results, as it implicitly reflects query difficulty—queries that challenge the strong model tend to be universally hard. Nonetheless, both strategies are outperformed by our proposed methods. Hard labels derived from greedy decoding offer additional improvements, suggesting that the router benefits from discrete supervision and can learn beyond merely detecting weak model failures. Lastly, a cost-efficient variant— **Hard Blocking without Strong Model Sampling**, which is described in Eq 6—replaces full sampling of the strong model with a single greedy decoding step and achieves comparable performance, making it a practical alternative under constrained computational budgets.

5 Conclusions

In this work, we propose **Firewall Routing**, a dual-model hybrid inference framework that leverages multiple sampling and innovative blocking techniques to optimize query routing. Through extensive experiments across various benchmarks, our approach demonstrates state-of-the-art performance, significantly reducing computational costs while maintaining high response quality.

Limitations

The generalization of the proposed hybrid inference system across different model pairs and datasets remains an area for further exploration. Future work should include a broader evaluation across diverse models and datasets to assess the scalability and applicability of the proposed approach in real-world, heterogeneous settings.

References

Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

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

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code. *Preprint*, arXiv:2107.03374.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Ruhle, Laks V. S. Lakshmanan, and Ahmed Hassan Awadallah. 2024. Hybrid llm: Cost-efficient and quality-aware query routing. *Preprint*, arXiv:2404.14618.

Tao Feng, Yanzhen Shen, and Jiaxuan You. 2025. Graphrouter: A graph-based router for llm selections. *Preprint*, arXiv:2410.03834.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal

Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

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

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

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *Preprint*, arXiv:2111.09543.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *Preprint*, arXiv:1705.03551.

Anil Kag, Igor Fedorov, Aditya Gangrade, Paul Whatmough, and Venkatesh Saligrama. 2023. Efficient edge inference by selective query. In *The Eleventh International Conference on Learning Representations*.

Keming Lu, Hongyi Yuan, Runji Lin, Junyang Lin, Zheng Yuan, Chang Zhou, and Jingren Zhou. 2023. Routing to the expert: Efficient reward-guided ensemble of large language models. *Preprint*, arXiv:2311.08692.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M Waleed Kadous, and Ion Stoica. 2024. Routellm: Learning to route llms with preference data. *Preprint*, arXiv:2406.18665.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Guillem Ramírez, Alexandra Birch, and Ivan Titov. 2024. Optimising calls to large language models with uncertainty-based two-tier selection. *Preprint*, arXiv:2405.02134.

Marija Šakota, Maxime Peyrard, and Robert West. 2024. 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*, pages 606–615.

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

Dimitris Stripelis, Zijian Hu, Jipeng Zhang, Zhaozhuo Xu, Alay Dilipbhai Shah, Han Jin, Yuhang Yao, Salman Avestimehr, and Chaoyang He. 2024. Tensoropera router: A multi-model router for efficient llm inference. *Preprint*, arXiv:2408.12320.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Gyeong-In Yu, Joo Seong Jeong, Geon-Woo Kim, Soojeong Kim, and Byung-Gon Chun. 2022. Orca: A distributed serving system for transformer-based generative models. In *16th USENIX Symposium on Operating Systems Design and Implementation (OSDI 22)*, pages 521–538.

A System Metrics of the Router Model

Regarding the choice of router model, we follow established practices in prior work and provide latency comparisons to contextualize its overhead. For instance, the appendix of HybridLLM (Ding et al., 2024) (Table 5) reports that the latency introduced by the router model is negligible:

Model	Latency (seconds)
Router	0.036 ± 0.002
FLAN-T5 (800M)	0.46 ± 0.039
LLaMA-2 (7B)	7.99 ± 0.15
LLaMA-2 (13B)	14.61 ± 0.27

Table 5: Latency values for different models reported in HybridLLM (Ding et al., 2024).

We report our empirical latency values in Table 6. Taking GSM8K as an example—which has the shortest average input and output lengths among our benchmarks (approximately 60 tokens for input and 35 tokens for generation)—we observe that longer sequences significantly increase LLM latency. In contrast, the router latency remains negligible and is unaffected by input or output length.

Model	Latency (seconds)
DeBERTa-v3-large (300M)	0.024 ± 0.002
LLaMA-3.2 (1B) – First Token	0.012 ± 0.001
LLaMA-3.2 (1B) – Finish Generation	0.890 ± 0.086
LLaMA-3.2 (3B) – First Token	0.048 ± 0.003
LLaMA-3.2 (3B) – Finish Generation	3.56 ± 0.103
LLaMA-3.1 (70B) – First Token	0.845 ± 0.005
LLaMA-3.1 (70B) – Finish Generation	57.85 ± 0.872

Table 6: Latency values for different models in our experiments.

Even for small-scale LLMs such as LLaMA-3.2 (1B), the latency introduced by router-based methods remains negligible compared to cascade-based approaches, which require waiting until the full generation is completed. This gap is further amplified in real-world scenarios, where modern LLMs often operate with long system prompts and extensive contexts. In such settings, cascade models experience even greater latency due to the need to process full outputs before making downstream decisions, whereas router models remain lightweight and unaffected by sequence length.

It is worth noting that, compared to the strong model (LLaMA3.1-70B), the computational overhead introduced by both the weak model and the router (DeBERTa-v3-large) is negligible. This is particularly evident in our setting, where the weak model (either LLaMA3.2-1B or 3B) requires only 0.9s or 3.56s to complete generation, and the router takes merely 0.024s per query. In contrast, the strong model takes approximately 57.85s to finish generation, meaning that the total latency in our method is overwhelmingly dominated by the fraction of queries dispatched to the strong model.

Consequently, the overall latency of the routing system closely approximates a linear interpolation between the weak and strong model latencies, weighted by the routing ratio. This also implies that the speedup over full strong model inference is roughly proportional to the percentage of queries filtered away from the strong model. For example, at a routing ratio of 50%, our method achieves a latency of 29.38s with the 1B weak model (compared to 57.85s with full strong model usage), leading to a nearly 1.97× speedup. Full results for both latency and speedup under different routing ratios, for both 1B and 3B weak models, are summarized in Table 7.

Routing Ratio	Avg. Latency (s)		Speedup	
	1B	3B	1B	3B
0%	0.902	3.560	64.14×	16.25×
20%	12.619	14.442	4.58×	4.01×
50%	29.377	30.729	1.97×	1.88×
80%	46.135	47.016	1.25×	1.23×
100%	57.850	57.850	1.00×	1.00×

Table 7: Overall latency (in seconds) and relative speedup under different routing ratios, using either LLaMA-3.2 1B or 3B as the weak model, and LLaMA-3.1 70B as the strong model. Latency includes the cost of DeBERTa router, weak model generation, and strong model generation. Speedup is computed as the ratio of 70B-only latency to current latency.

In our study, we adopt a linear interpolation baseline—i.e., *Random Routing*—which serves as a reference point that approximates the expected performance between two extremes: always routing to the weak model and always routing to the strong model. This baseline provides a meaningful point of comparison for evaluating the effectiveness of various routing strategies. For completeness, we summarize the zero-shot performance of the constituent LLMs in Table 8.

Model	TriviaQA	GSM8K	HumanEval
LLaMA3.2-1B	9.94%	8.05%	6.80%
LLaMA3.2-3B	17.18%	9.18%	9.49%
LLaMA3.1-70B	60.00%	26.27%	13.39%

Table 8: Zero-shot pass rates of the weak and strong models. These serve as endpoints for evaluating the effectiveness of routing policies under a linear interpolation baseline.

B Call-Performance Threshold (CPT)

Call-Performance Threshold (CPT) (Ong et al., 2024) measures the minimum percentage of queries that need to be routed to the strong model in order to achieve a certain percentage (e.g., 20%, 50%, or 80%) of the full performance gap between the weak and strong models. However, this formulation suffers from an inherent bias: closing the bottom-$n\%$ of the performance gap is significantly easier than closing the top-$n\%$. This is because a large fraction of "easy" queries can be accurately predicted by even simple heuristics (e.g., margin-based uncertainty), enabling models to quickly reduce the apparent performance gap with relatively few strong model calls. In contrast, the remaining "hard" queries require deeper reasoning or more expressive models and are disproportionately challenging to resolve.

As a result, CPT is highly sensitive to the distribution of query difficulty, and improvements on CPT can often be inflated by correctly routing trivial or low-complexity queries while failing to address more meaningful or representative cases. Moreover, CPT provides no insight into whether the selected routing decisions generalize well or preserve robustness across datasets and tasks.

Although our method still achieves state-of-the-art CPT scores, we do not adopt it as a primary metric for evaluation. Instead, we include it in the appendix for completeness and reproducibility, and base our main analysis on metrics like Pass Rate and APGR, which better reflect the trade-off between quality and efficiency in practical deployments.

Datasets	TriviaQA	GSM8K	HumanEval
Metrics	CPT↓ 20% / 50% / 80 %		
Hybrid LLM	20 / 50 / 80	10 / 30 / 62	20 / 44 / 76
RouteLLM (MF)	18 / 46 / 76	20 / 50 / 82	22 / 54 / 78
Margin Sampling	20 / 48 / 78	24 / 54 / 84	26 / 50 / 88
Ours (Hard Block)	16 / **42** / **72**	10 / 28 / 58	**10** / 46 / **72**
Ours (Soft Block)	**14** / **42** / **72**	**8** / **24** / **54**	18 / **40** / **72**

Table 9: Zero-shot CPT performance of different methods across selected datasets. The weak model is Llama3.2-1B, and the strong model is Llama3.1-70B. **Bolded values** indicate the best-evaluated results.

C Is BartScore a Reliable Metric of Response?

We calculate the BartScore for the responses of different LLMs on TriviaQA and GSM8K. The responses are sorted based on their BartScore, and the sorted responses are grouped into bins. Average accuracy is then calculated within each bin to assess the performance of the models at different levels of response correctness.

As shown in Figure 3, 4, a correlation between BartScore and accuracy is only observed on TriviaQA with Llama3.1-70B. In other cases, no consistent or discernible pattern is evident.

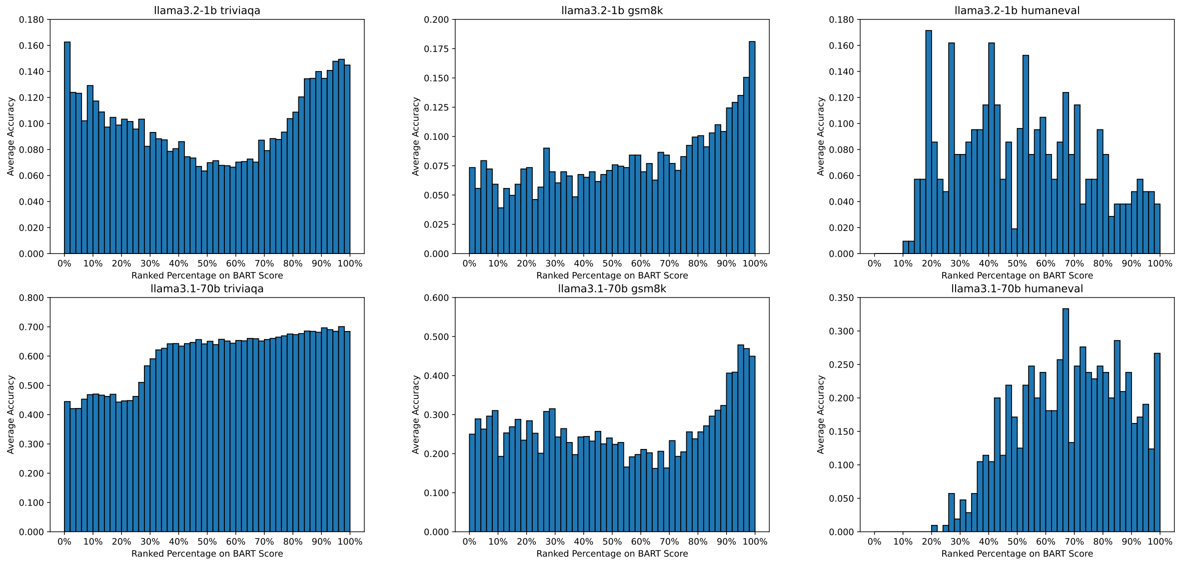

Figure 3: BartScore analysis of LLM responses on TriviaQA, GSM8K, and HumanEval. The responses are sorted by BartScore and grouped into bins, with accuracy calculated within each bin to evaluate performance at varying levels of response quality.

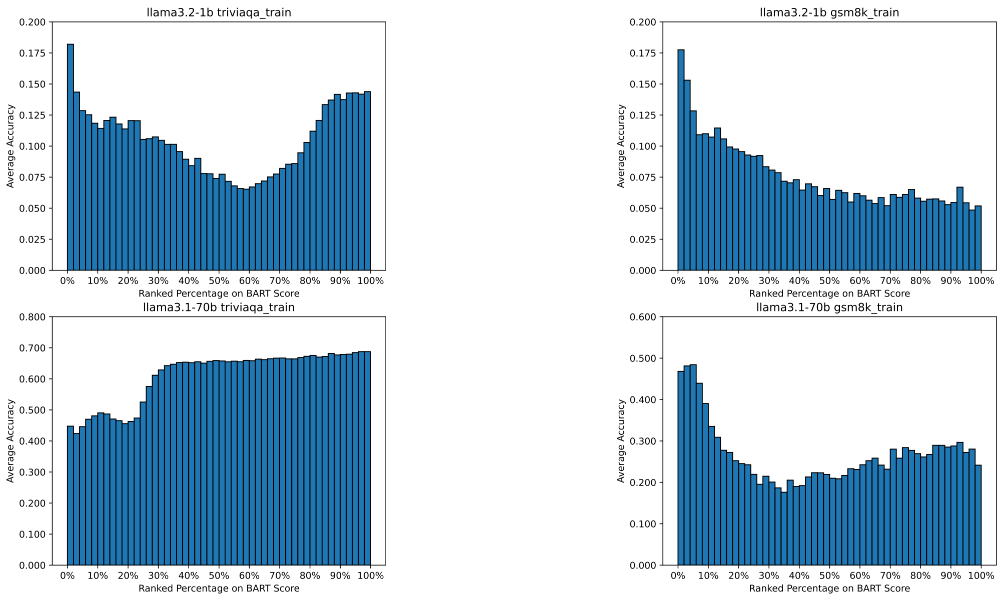

Figure 4: BartScore analysis of LLM responses on training set of TriviaQA and GSM8K. The responses are sorted by BartScore and grouped into bins, with accuracy calculated within each bin to evaluate performance at varying levels of response quality.

D Visualization of Route Method Performance

Similarly, we rank all queries based on the values predicted by the models, and patch them into distinct bins. For each bin, we compute the average pass rate of the strong model and the weak model. Additionally, we evaluate the improvement in pass rate achieved by routing the queries in each bin to the strong model, rather than to the weak model.

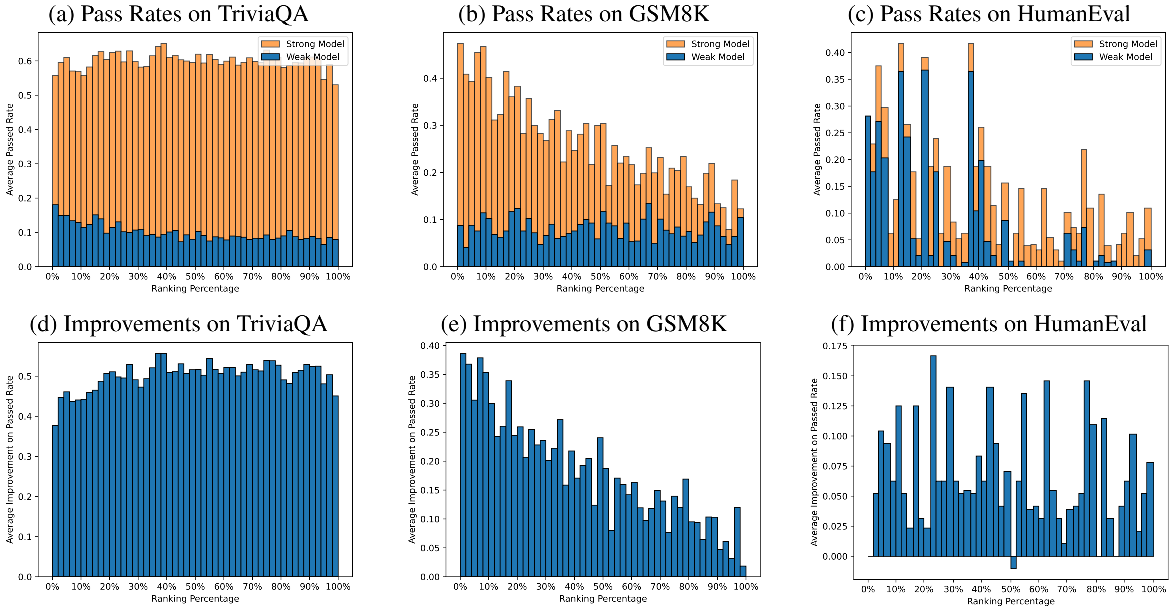

Figure 5: Performance evaluation of reproduced Hybrid LLM on selected datasets. The system utilizes Llama3.2-1B as weak model and Llama3.1-70B as strong model. Results are presented in a zero-shot setting.

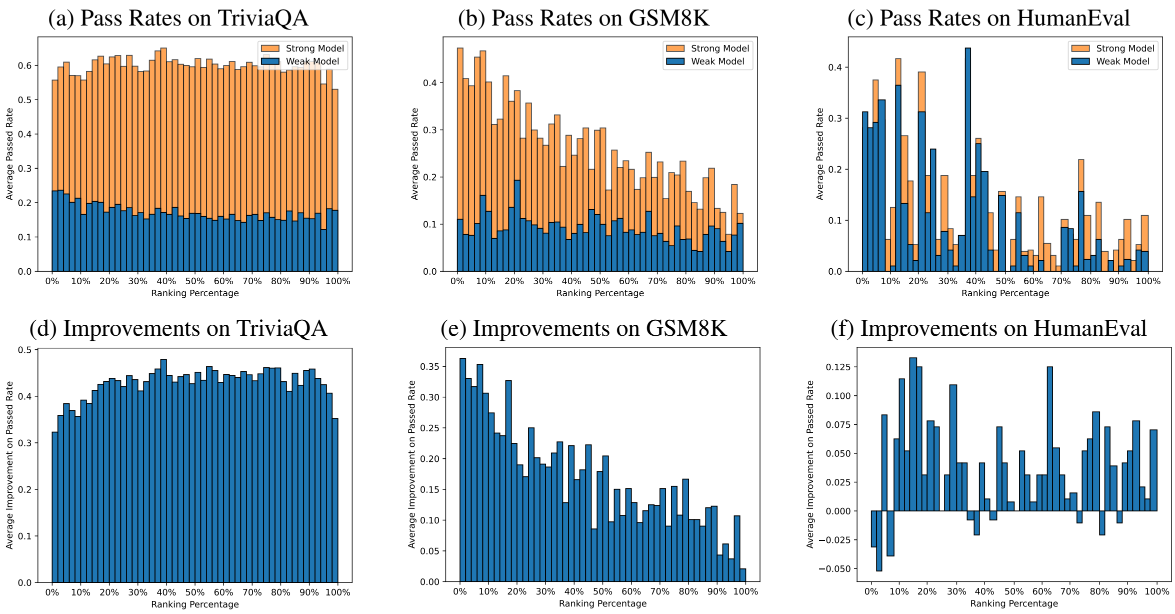

Figure 6: Performance evaluation on generalization of reproduced Hybrid LLM on selected datasets. Evaluated on a system with Llama3.2-3B as weak model and Llama3.1-70B as strong model. Results are presented in a zero-shot setting.

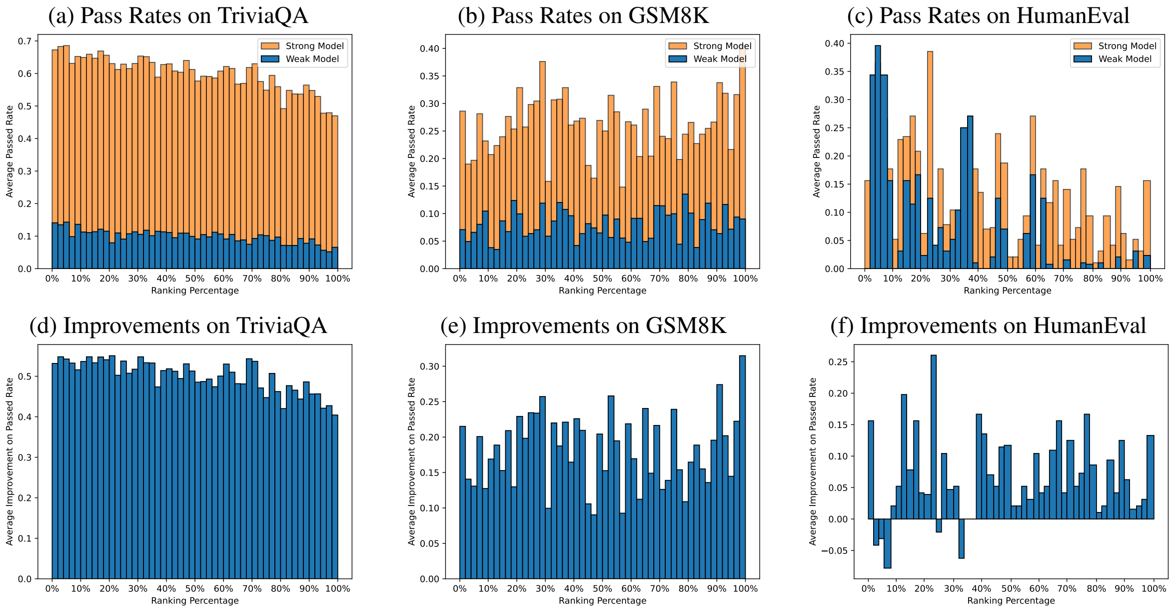

Figure 7: Performance evaluation of Matrix Factorization from RouteLLM on selected datasets. The system utilizes Llama3.2-1B as weak model and Llama3.1-70B as strong model. Results are presented in a zero-shot setting.

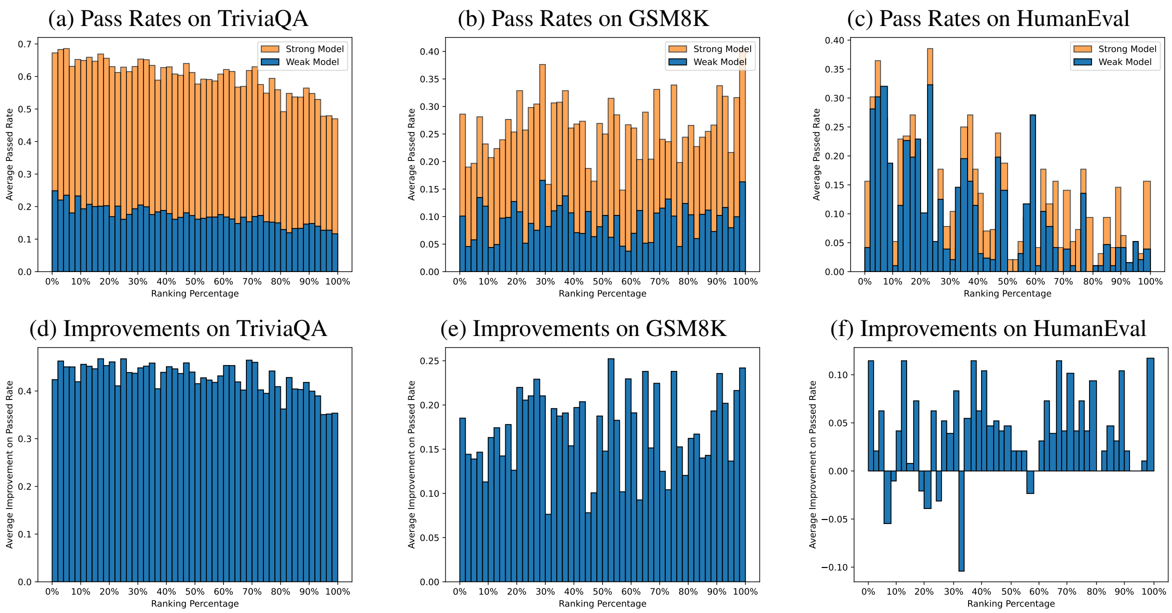

Figure 8: Performance evaluation on generalization of Matrix Factorization from RouteLLM on selected datasets. Evaluated on a system with Llama3.2-3B as weak model and Llama3.1-70B as strong model. Results are presented in a zero-shot setting.

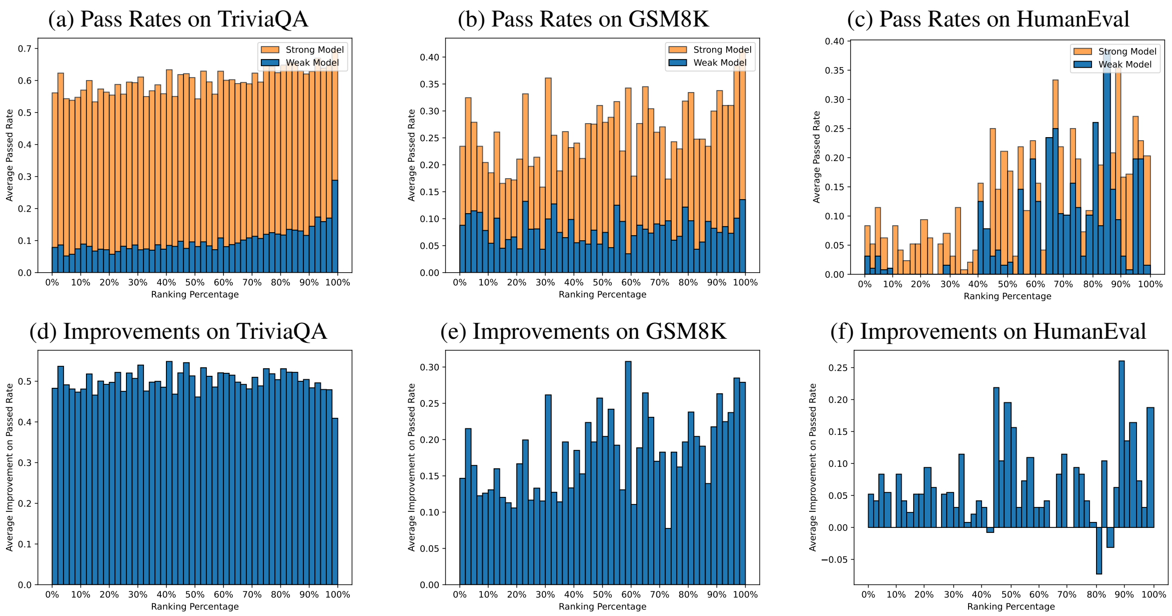

Figure 9: Performance evaluation of Margin Sampling on selected datasets. The system utilizes Llama3.2-1B as weak model and Llama3.1-70B as strong model. Results are presented in a zero-shot setting.

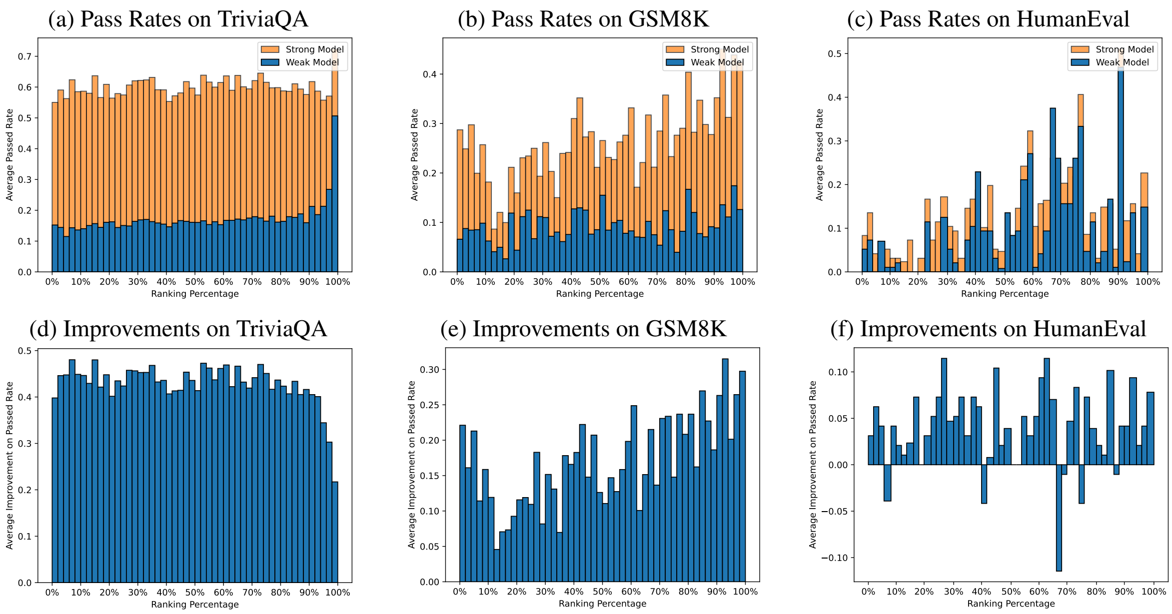

Figure 10: Performance evaluation on generalization of Margin Sampling on selected datasets. Evaluated on a system with Llama3.2-3B as weak model and Llama3.1-70B as strong model. Results are presented in a zero-shot setting.

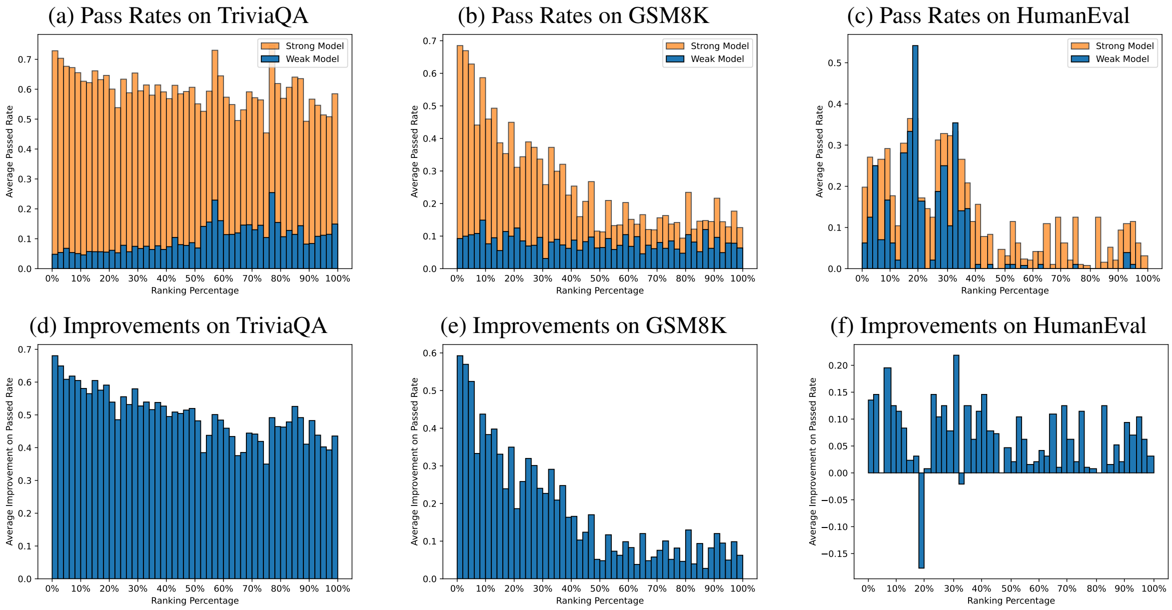

Figure 11: Performance evaluation of Hard Blocking on selected datasets. The system utilizes Llama3.2-1B as weak model and Llama3.1-70B as strong model. Results are presented in a zero-shot setting.

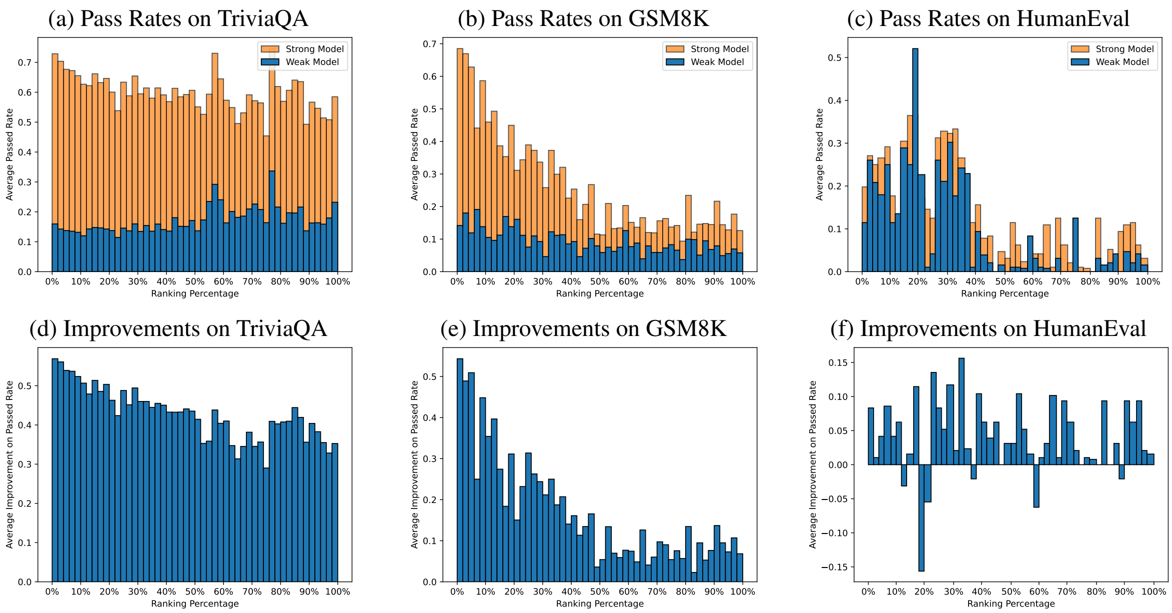

Figure 12: Performance evaluation on generalization of the Hard Blocking on selected datasets. Evaluated on a system with Llama3.2-3B as weak model and Llama3.1-70B as strong model. Results are presented in a zero-shot setting.

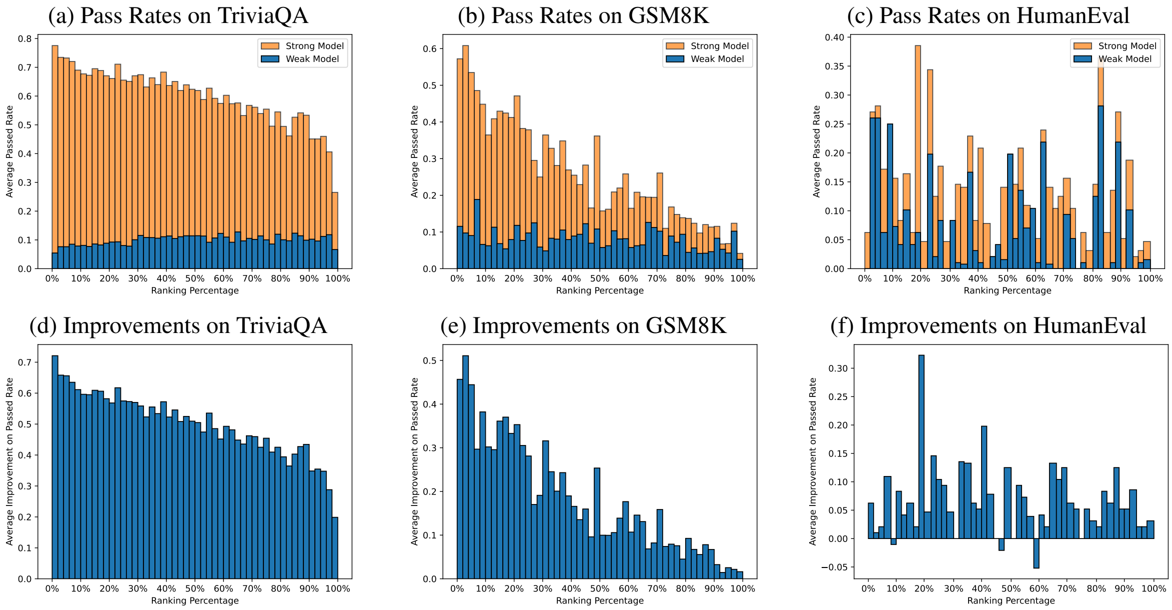

Figure 13: Performance evaluation of Soft Blocking on selected datasets. The system utilizes Llama3.2-1B as weak model and Llama3.1-70B as strong model. Results are presented in a zero-shot setting.

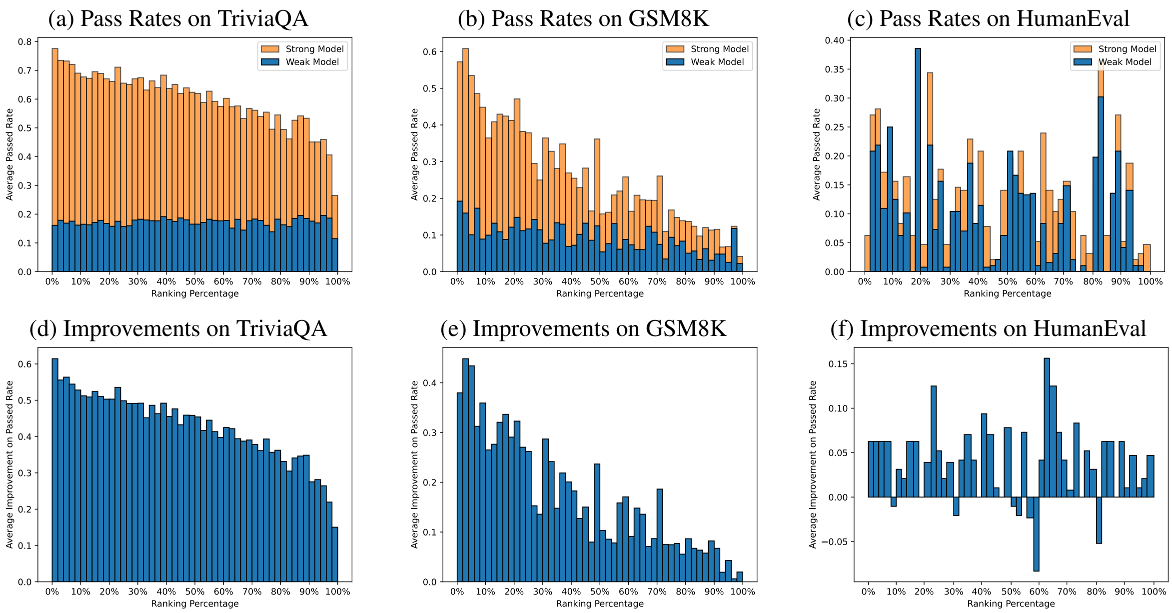

Figure 14: Performance evaluation on generalization of the Soft Blocking on selected datasets. Evaluated on a system with Llama3.2-3B as weak model and Llama3.1-70B as strong model. Results are presented in a zero-shot setting.

Figure 15: Performance evaluation of the router trained on Weak Model's Pass Rates across selected datasets. The system utilizes Llama3.2-1B as weak model and Llama3.1-70B as strong model. Results are presented in a zero-shot setting.

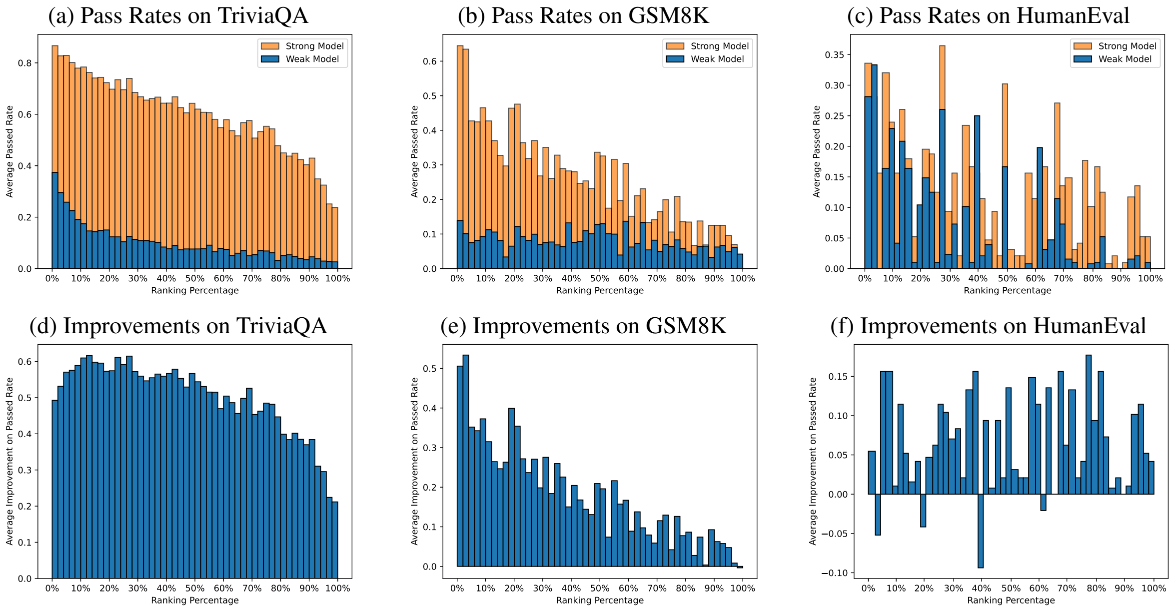

Figure 16: Performance evaluation of the router trained on Strong Model's Pass Rates across selected datasets. The system utilizes Llama3.2-1B as weak model and Llama3.1-70B as strong model. Results are presented in a zero-shot setting.

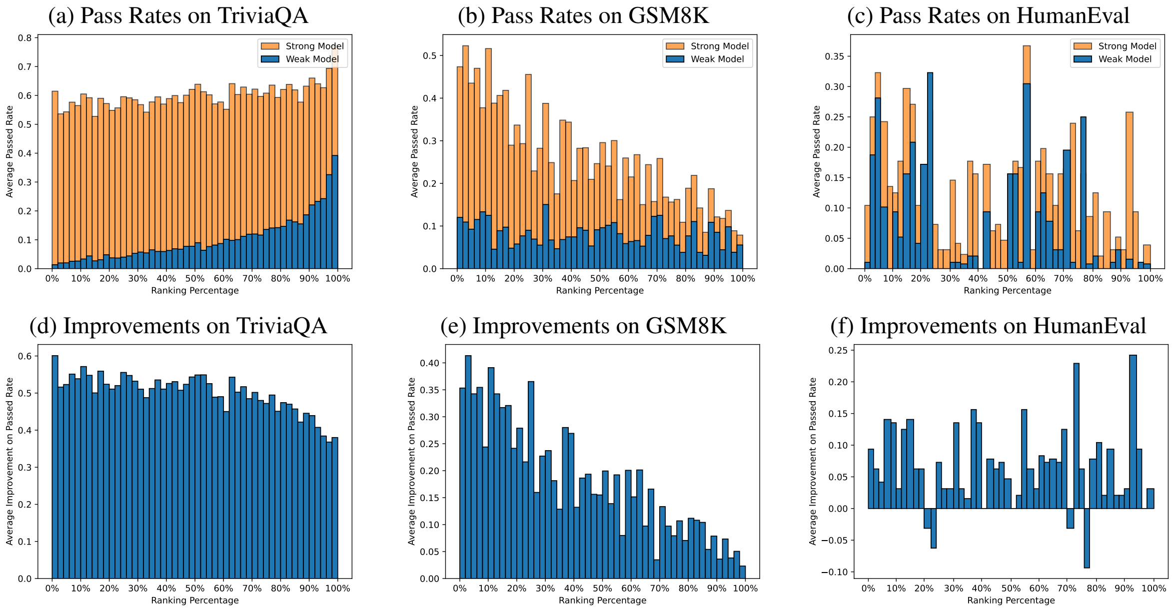

Figure 17: Performance evaluation of the router trained on Hard Labels attained with greedy decoding across selected datasets. The system utilizes Llama3.2-1B as weak model and Llama3.1-70B as strong model. Results are presented in a zero-shot setting.

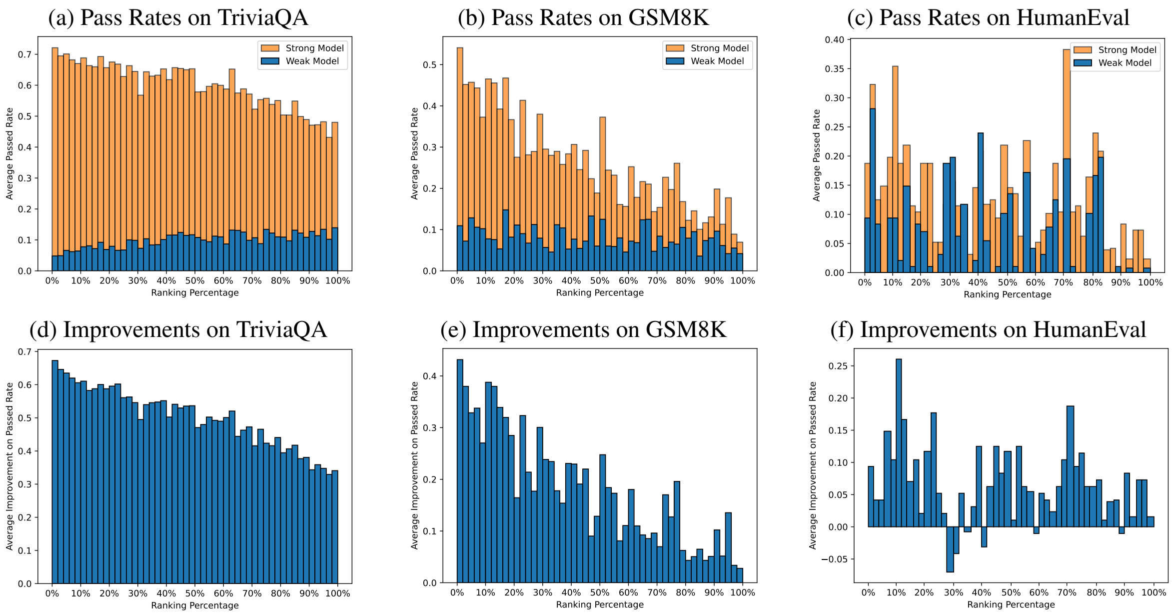

Figure 18: Performance evaluation of the router trained using Hard Blocking without conducting sampling on the strong model across selected datasets The system utilizes Llama3.2-1B as weak model and Llama3.1-70B as strong model. Results are presented in a zero-shot setting.