MESS+: Dynamically Learned Inference-Time LLM Routing in Model Zoos with Service Level Guarantees

Herbert Woisetschläger
Technical University of Munich
h.woisetschlaeger@tum.de

Ryan Zhang
Horace Greeley High School
ryzhangofficial@gmail.com

Shiqiang Wang
IBM Research
shiqiang.wang@ieee.org

Hans-Arno Jacobsen
University of Toronto
jacobsen@eecg.toronto.edu

Abstract

Open-weight large language model (LLM) zoos provide access to numerous high-quality models, but selecting the appropriate model for specific tasks remains challenging and requires technical expertise. Most users simply want factually correct, safe, and satisfying responses without concerning themselves with model technicalities, while inference service providers prioritize minimizing operating costs. These competing interests are typically mediated through service level agreements (SLAs) that guarantee minimum service quality. We introduce MESS+, a stochastic optimization algorithm for cost-optimal LLM request routing while providing rigorous SLA compliance guarantees. MESS+ learns request satisfaction probabilities of LLMs in real-time as users interact with the system, based on which model selection decisions are made by solving a per-request optimization problem. Our algorithm includes a novel combination of virtual queues and request satisfaction prediction, along with a theoretical analysis of cost optimality and constraint satisfaction. Across a wide range of state-of-the-art LLM benchmarks, MESS+ achieves an average of $2\times$ cost savings compared to existing LLM routing techniques.

1 Introduction

As the number of open-weight large language models (LLMs), such as Llama [8], Granite [11] or Qwen [19], increases rapidly, deep learning infrastructure providers and end users are confronted with an abundance of models (model zoo) for their language processing tasks. Typically, each LLM family comes with at least three models, each with different capabilities and resource requirements (Figure 1). Sometimes, there is an update to the model weights that is released as a minor checkpoint (e.g., Llama 3.1 70B and Llama 3.3 70B). This leaves many users questioning what is the best model to use and whether common benchmark results apply to their specific needs [16]. Currently, the best way to approach model selection is educated guessing, using LLM benchmarks as a proxy to estimate model performance, or spending significant efforts to curate human-preference datasets for request routing [18]. Since working with LLMs can be expensive [22], minimizing costs is an equally high priority for end users and inference endpoint operators. This leaves us with the following tri-fold problem.

End-users primarily care about a factually correct and safe model output. When inquiring about text information, e.g., by asking questions or requesting language translation, end users are mostly interested in obtaining factually correct and sufficiently clear language output [26]. Additionally, many users are unfamiliar with the technical details of LLMs, making it challenging for them to select the right model for the job, i.e., their primary references are domain-specific benchmark

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

 https://doi.org/10.52202/085713-1804

rankings [9, 10]. However, there is no intuitive method to compare the complexity of individual requests with benchmark tasks. Thus, we require a method that learns over time whether a model can satisfy an incoming request as users interact with LLMs.

Inference endpoint providers prioritize low operating costs, and emerging AI legislation requires sustainable computing best practices. Operating infrastructure that can run state-of-the-art LLMs can be costly. Microsoft has announced it will acquire a stake in the Three Mile Island nuclear power plant in the United States to satisfy the energy demand of its planned data center in Pennsylvania, which has two reasons: consistent energy delivery and low energy cost [20]. At the same time, globally emerging AI regulation (e.g., EU AI Act Article 95) [6] emphasizes energy monitoring and compliance with sustainable computing best practices. Taken together, tighter regulation and ever increasing cost underpin the necessity of energy-optimal service operations. The current operating model of many inference service providers where only a single model is served per dedicated endpoint does not offer the necessary flexibility to keep cost in check. When aiming to provide cost optimal inference endpoints, we need a method that can choose the most efficient model that is likely to satisfy an incoming request.

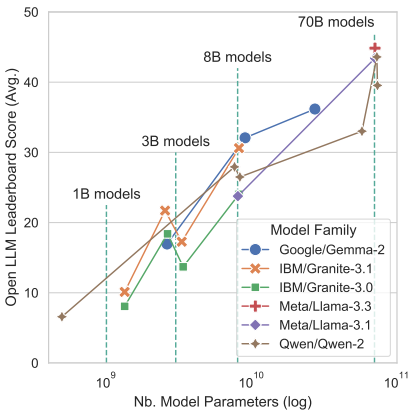


Figure 1: OpenLLM-Leaderboard performance comparison of popular LLM families. Each family typically consists of a minimum of three models with distinct capabilities and cost characteristics.

Enterprise use-cases require a consistently high-quality model inference output while keeping costs in check. This unites the requirement for high-quality model outputs and price sensitivity. Thus, commercial players typically rely on service-level agreements (SLAs) when sourcing services for their own products. An example of an SLA is one that specifies a percentage of requests to be automated by LLM-powered AI agents, while the remaining requests need to be handled by human experts. Typically, different SLA levels offer different quality standards at different prices. However, to date, it is usually up to end users to decide which models to use for their requests without any lower bounds on request satisfaction guarantees. As such, this is neither cost effective for the inference endpoint provider nor for the end user as both tend to overpay on operating costs and inference response costs, respectively. Thus, we require a method that minimizes operating costs while guaranteeing SLA compliance, i.e., a minimum request satisfaction rate over time.

In summary, we ask the fundamental question:

> *How can we design an algorithm for selecting LLMs from a model zoo that minimizes the operating cost while providing rigorous SLA compliance guarantees?*

This problem is challenging in several aspects. First, we need to find a way to learn whether an LLM can satisfy an incoming request as users interact with an inference endpoint over time. Second, we require a method to guarantee SLA compliance over time, while allowing the inference service provider to minimize their operating costs. Taken together, we have to guarantee that our per-request routing decisions rigorously satisfy an SLA requirement while ensuring cost optimality.

Contributions. Our paper introduces **MESS+** (**M**odel S**E**lection with Cost-optimal **S**ervice-level Guarantee**S**), a new stochastic optimization framework for minimizing operating costs and rigorously guaranteeing SLA compliance. Compared to existing LLM request routing techniques, we offer a solution to *cost optimal request routing* and provide a *lower bound request satisfaction guarantee*. Our approach dynamically learns request satisfaction probabilities in an online fashion as users interact with the system and makes routing decisions based on operating cost and request satisfaction over time. We provide *theoretical guarantees for cost-optimal SLA compliance with model zoos*.

Related Work. While prior works in LLM request routing have made significant contributions in various directions (Table 2), MESS+ distinguishes itself through its formal optimization approach to cost efficiency with rigorous SLA guarantees. Specifically, RouteLLM [18], Zooter [15], and RouterDC [3] focus on optimizing routing decisions based on model capabilities and query characteristics without formal SLA guarantees. LLM-Blender [12] and AutoMix [1] emphasize quality improvement through ensembling and self-verification approaches, respectively, without providing guarantees for a

https://doi.org/10.52202/085713-1804

Table 1: Related work overview. We are the first to introduce a cost-optimal stochastic optimization framework for LLM request routing with SLA guarantees.

Approach	Technique	# of LLMs in Zoo	Cost-aware	Service-Level Guarantee	Source
LLM-Blender	LLM Ensemble	> 2	–	–	Jiang et al. [12]
AutoMix	Self Verification	2	–	–	Aggarwal et al. [1]
Hybrid-LLM	Preference Data	2	–	–	Ding et al. [7]
Zooter	Reward-Model Labels	> 2	–	–	Lu et al. [15]
RouterDC	Contrastive Learning	> 2	–	–	Chen et al. [3]
TensorOpera Router	BertSim Scores	> 2	✓	–	Stripelis et al. [24]
RouteLLM	Preference Data	2	✓	–	Ong et al. [18]
MESS+ (ours)	Satisfaction Scores & Online Optimization	> 2	✓	✓	This paper

lower bound request satisfaction rate over time. Similarly, Hybrid-LLM [7] introduces quality-aware routing between models of different sizes, and TensorOpera Router [24] balances performance metrics empirically. In contrast, MESS+ uniquely formulates request routing as a stochastic optimization problem that minimizes operating costs while providing a minimum request satisfaction guarantee over time, adapting dynamically through online learning as users interact with the system. This optimization-driven approach with provable guarantees positions MESS+ as the first framework to deliver cost-optimal SLA compliance in LLM zoos.

2 MESS+: Model Selection with Cost-Optimal Service Level Guarantees

The overall goal of MESS+ is to find the most suitable LLM for each inference request $t \in \{1, 2, \ldots, T\}$ to minimize the operating cost $E_{m,t}$, while conforming to model performance constraints defined by an SLA over time to ensure contractual compliance.

2.1 Problem Formulation

Consider a language model zoo with M different LLMs. For every request t, each model is associated with a user satisfaction $s_{m,t} \in \{0, 1\}$ indicating whether model m can satisfy the t-th request, where $m \in \{1, 2, \ldots, M\}$. The value of $s_{m,t}$ is *unknown* before request t arrives.

Inference Cost (Objective). Each model in a zoo incurs a certain amount of cost $E_{m,t}$ (e.g., cost for an API call, energy consumption of an inference request), which can vary greatly based on the model and inference request size. For instance, a zoo can include models with 1B, 8B, and 70B parameters. These differences in size make costs volatile. In a scenario where users can choose the model and are unsure which model fits their request, they are likely to always choose the largest model, making model serving expensive, rendering costs even more volatile and overall hard to predict.

Service-Level Agreement (Constraint). Typically, contracts related to a service contain a list of requirements, including an SLA defining a minimum service quality. The SLA functions as a measurable control system with defined input variables, output metrics, and acceptable tolerance ranges. We define α to be the target request satisfaction rate over time[1], i.e., the relative share of requests that need to be served with a satisfactory LLM response over time.

Control Problem. Taking the objective and constraint together, we can formalize the following problem of minimizing the average operating cost per request under a minimum performance requirement defined via an SLA over time:

$$\min_{\{y_{m,t}:\forall t,m\}} \quad \frac{1}{T}\sum_{t=1}^{T}\sum_{m=1}^{M}\mathbb{E}\left[y_{m,t}E_{m,t}\right], \tag{1a}$$

$$\text{s.t.} \quad \frac{1}{T}\sum_{t=1}^{T}\sum_{m=1}^{M}\mathbb{E}\left[y_{m,t}s_{m,t}\right] \geq \alpha, \tag{1b}$$

$$\sum_{m=1}^{M} y_{m,t} = 1, \quad \forall t \in \{1, \ldots, T\}, \tag{1c}$$

$$y_{m,t} \in \{0, 1\}, \quad \forall t \in \{1, \ldots, T\}, m \in \{1, \ldots, M\}, \tag{1d}$$

where $y_{m,t} = 1$ if model m is chosen and $y_{m,t} = 0$ otherwise.

[1] In practice, α should be chosen with a certain safety margin from the SLA requirement such that we do not violate the SLA even if the average request satisfaction is slightly below α.

https://doi.org/10.52202/085713-1804

Challenges. Our optimization problem involves an inherent trade-off between request satisfaction and operating cost, since larger LLMs, and thus more capable ones, typically yield higher satisfaction rates while consuming more resources at the same time. As such, we see a *correlation over time between the objective and constraints*. Further, optimizing the operating cost involves a time average that is hard to predict a priori as the properties of future requests are generally *unknown* and *heterogeneous*.

2.2 Online Decision Making

We introduce an online decision making process that addresses the aforementioned challenges. The quantities $E_{m,t}$ and $s_{m,t}$ are captured for every request without knowledge of future statistics. The full procedure of MESS+ is described in Algorithm 1.

Algorithm 1: **M**odel S**E**lection with Cost-optimal **S**ervice-level Guarantee**S** (MESS+)

Input: T, V, α, c, learning rate $\eta_t > 0, \forall t$
Output: $\{y_{m,t} : \forall m,t\}$, outputs of chosen models for all t

Initialize $Q_1 \leftarrow 0$, random vector $\mathbf{z}_1$;
for $t \leftarrow 1$ **to** T **do**
  Compute $p_t \leftarrow \min\left(1, {}^{c}/\sqrt[4]{t}\right)$;
  Sample $X_t \sim \text{Bernoulli}(p_t)$;
  if $X_t = 1$ *or* $t = 1$ **then**
    `// Explore model zoo`
    $y_{m,t} \leftarrow 1, \forall m$; `// all models queried`
    foreach $m \in \{1, 2, ..., M\}$ **do**
      Obtain true request satisfaction $s_m(t)$;
    $\mathbf{z}_{t+1} \leftarrow \mathbf{z}_t - \eta_t \nabla F(\mathbf{z}_t, \mathbf{a}_t)$; `// SGD using request t's content` $\mathbf{a}_t$
    $m^* \leftarrow \arg\max_m s_{m,t}$;
  else
    `// Online decision making with user satisfaction predictor`
    Predict request satisfaction probability $\hat{s}_{m,t}, \forall m$;
    $m^* \leftarrow \arg\min_m VE_{m,t} + Q_t(\alpha - \hat{s}_{m,t})$; `// Solve Problem` (3)
    $y_{m^*,t} \leftarrow 1, y_{m',t} \leftarrow 0, \forall m' \neq m^*$;
    $\mathbf{z}_{t+1} \leftarrow \mathbf{z}_t$;
  Get output from model m^* and its accuracy $s_{m^*,t}$;
  $Q_{t+1} \leftarrow \max\{0, Q_t + \alpha - s_{m^*,t}\}$;
    `// Virtual queue update`

Methodology. Our approach is inspired by the Lyapunov drift-plus-penalty framework [17], with *novel extensions* to support a request satisfaction predictor that is learned in an online manner and used in per-request model selection decisions.

Request Satisfaction Prediction. Since $s_{m,t}$ is only known after invoking model m, we need a mechanism that predicts whether a model m will meet the request satisfaction requirements, so that we can select an appropriate LLM for every incoming request *before* the request is sent to any LLM. We learn a predictor $\hat{s}_{m,t} \in [0, 1]$ online that predicts the *probability* that model m can satisfy an incoming request t for all request-model combinations. The predictor takes the request as input and extracts useful information of the request using a lightweight model to make the prediction. For different requests, the prediction $\hat{s}_{m,t}$ is usually different. The detailed procedure of this prediction is described in Section 2.3.

Virtual Queues. We capture SLA violations, i.e., accumulated undershooting of α, in a *virtual queue* Q with the following update procedure after receiving and processing request t:

$$Q_{t+1} = \max\left\{0, Q_t + \alpha - \textstyle\sum_{m=1}^{M} y_{m,t} s_{m,t}\right\}, \tag{2}$$

where for $t = 1$, we initialize $Q_1 = 0$. Intuitively, this captures the cumulative constraint violations, which can be seen by comparing (1b) and (2). Hence, we aim to collectively minimize our objective (1a) and the queue length[2].

Decision Problem for Each Request. We aim to minimize the operating cost of every inference request t while complying with our SLA requirement α. This trade-off is formulated as follows:

$$\min_{\{y_{m,t}:\forall m\}} \quad V \cdot \textstyle\sum_{m=1}^{M} y_{m,t} E_{m,t} + Q_t \left(\alpha - \sum_{m=1}^{M} y_{m,t} \hat{s}_{m,t}\right), \tag{3a}$$

$$\text{s.t.} \quad \text{Constraints (1c), (1d)}. \tag{3b}$$

As SLAs come in various configurations, we introduce the parameter $V > 0$ in (3) that controls the speed at which we reach α and the trade-off between operating cost and time at which we can guarantee constraint (SLA) satisfaction. The effect of V will be further discussed in both theory and experiments in later sections. In particular, constraint satisfaction is theoretically guaranteed for large enough T (see Section 3). When V is small, the constraint violation decreases faster in T, i.e., we achieve the SLA requirement α more quickly, but the cost is higher, and vice versa. In practice, we see that a fixed V obtained from coarse tuning works reasonably well across a wide range of benchmarks (see Section 4).

[2]We note that the average satisfaction rate over requests needs to be greater than or equal to α in the constraint, so the direction of inequalities in the constraint is opposite to [17], thus our queue update equation in (2) is slightly different from that in [17].

Note that we assume that the user satisfaction is captured immediately after the LLM response is generated. In practice, users would be shown a feedback prompt to assess the response quality or request human assistance if the response is unsatisfactory. Therefore, in (2), we use $s_{m,t}$ to update the virtual queue length to capture the actual constraint satisfaction, but we use $\hat{s}_{m,t}, \forall m$, to solve the per-request optimization problem (3). We consider the operating cost per request $E_{m,t}, \forall m, t$, to be known[3] once we receive the request and obtain some of its basic information, e.g., number of tokens in the prompt.

2.3 Request Satisfaction Prediction

We now describe how to obtain the predicted request satisfaction probability $\hat{s}_{m,t}, \forall m, t$, which is required to solve (3). To facilitate the description, let $\mathbf{z}_t$ denote the parameter vector of the *lightweight* request satisfaction predictor used for request t, and $\mathbf{a}_t \sim \mathcal{A}$ denote the t-th input request sampled from some distribution $\mathcal{A}$. The predictor provides an M-dimensional output $\hat{s}(\mathbf{z}_t, \mathbf{a}_t)$, which includes $\{\hat{s}_{m,t}, \forall m\}$. We write the m-th component of the output vector as $\hat{s}(\mathbf{z}_t, \mathbf{a}_t)_m = \hat{s}_{m,t}$. We omit the subscript t in $\mathbf{z}_t$ in the following when it is unnecessary to specify the request index t.

We define the regularized cross entropy objective of request satisfaction predictions for over all possible incoming requests and for all models:

$$F(\mathbf{z}) := -\mathbb{E}_{\mathbf{a}_t \sim \mathcal{A}} \left[\frac{1}{M} \sum_{m=1}^{M} \left(s_{m,t} \log \hat{s}(\mathbf{z}, \mathbf{a}_t)_m + (1 - s_{m,t}) \log(1 - \hat{s}(\mathbf{z}, \mathbf{a}_t))_m \right) \right] + \frac{\mu}{2} \|\mathbf{z}\|^2, \quad (4)$$

where the last term is a regularization term when $\mu > 0$.

We learn $\mathbf{z}$ through a probabilistic exploration and update procedure. To *explore* the model zoo, we query all models[4] with the same input request to obtain their actual request satisfaction $s_{m,t}$, allowing us to learn $\mathbf{z}$. More specifically, as shown in Algorithm 1, we sample from a distribution $X_t \sim \text{Bernoulli}(p_t)$, where the exploration probability $p_t = \min(1, \frac{c}{\sqrt[4]{t}})$ and $c > 0$ is a parameter that adjusts this probability. The probability p_t decays over time as the estimation $\hat{s}_{m,t}$ improves with each exploration and update step. The larger c, the more likely it is to perform an exploration step over time. When exploring, we always use the output from the largest model as the final model output as we have already incurred operating cost to query the largest model, i.e., we *do not* use the solution from (3) in this case and return the output of the largest model. Especially for the first few arriving requests, when we do not know how to choose the optimal LLM for request t, we must explore each m in the model zoo for the best-performing model for each request. In this way, we capture the actual user preference $s_{m,t}$ of each m, which we use to learn $\mathbf{z}$.

We note that the predictor we train here only predicts whether each model m can produce a satisfactory response to an incoming request. Different from related works such as [3, 18], we do not use this trained predictor as a router directly. Instead, the output of the predictor is fed into the per-request optimization problem (3), and the model selection decision is obtained by solving (3).

Key Insight. The key idea behind exploration and predictor update with a probability p_t that decreases in t is to strike a balance between obtaining an accurate predictor and limiting the additional cost incurred due to exploration. If p_t decreases too fast in t, it will take a long time to obtain an accurate predictor although the cost overhead due to exploration is low. In contrast, if p_t decreases too slowly in t, we obtain an accurate predictor quickly, but the cost overhead is also high since we do many more exploration steps than it is actually needed. By choosing $p_t = \min(1, \frac{c}{\sqrt[4]{t}})$, we have a good balance between the predictor accuracy and exploration cost overhead, as also confirmed by our theoretical analysis in the next section.

3 Performance Analysis

In this section, we provide a theoretical performance analysis of our proposed MESS+ algorithm. We focus on showing constraint satisfaction and optimality of our solution. Since SLAs are often volume-based, service quality guarantees are typically given for a specific number of requests or

[3] Practically, this can be done by profiling the cost of inference calls before adding a model to the zoo.

[4] In Line 6 of Algorithm 1, we set $y_{m,t} = 1$ for all m to reflect that all the models are queried during exploration. This is with a slight abuse of notation, since this choice of $\{y_{m,t}\}$ does not satisfy (1c). However, we use this configuration to indicate that cost has been incurred for querying all the LLMs.

https://doi.org/10.52202/085713-1804

transactions. Thus, we focus on obtaining results that hold for any $T \geq 1$ in our analysis, instead of only considering $T \to \infty$ as in [17]. As commonly done in the literature, our analysis relies on some reasonable assumptions to make the problem mathematically tractable. The full proofs for all the following theorems are provided in the appendix.

3.1 Constraint Satisfaction

Assumption 1. Let $\tilde{m} := \arg\max_m \hat{s}_{m,t}$ and $\psi \geq 0$ be some constant independent of the total number of requests T and $\psi \ll T$. We assume that, when $t \geq \psi$, there exist $\beta > 0$ and $q \in (0, 1]$, such that for any m (inclusive of $\tilde{m}$) with $\hat{s}_{\tilde{m},t} - \hat{s}_{m,t} \leq \beta$, we have $\Pr\{s_{m,t} = 1\} \geq q > \alpha$.

This assumption states that, after a finite number of requests ψ, our request satisfaction predictor becomes accurate enough, so that models within a certain range β of the maximum predicted satisfaction guarantees that the minimum probability of actual satisfaction is at least $q > \alpha$.

Theorem 1. *For any $t \geq 1$, we have the following upper bounds on the virtual queue length:*

$$\mathbb{E}\left[Q_t\right] \leq \max\left\{\alpha\psi; \tfrac{V\Delta_E}{\beta}\right\} + \sqrt{t} \;\; \textit{and} \;\; \tfrac{1}{t}\sum_{\tau=1}^{t}\mathbb{E}\left[Q_t\right] \leq \max\left\{\alpha\psi; \tfrac{V\Delta_E}{\beta}\right\} + \tfrac{1}{2(q-\alpha)}, \quad (5)$$

where $\Delta_E := E_{\max} - E_{\min}$, in which $E_{\max}$ and $E_{\min}$ are the maximum and minimum operating costs of any model, respectively.

The proof for Theorem 1 is based on a unique observation that, when $Q_t > \max\left\{\alpha\psi; {}^{V\Delta_E}/_{\beta}\right\}$, the solution to (3) is guaranteed to satisfy $\hat{s}_{\tilde{m},t} - \hat{s}_{m^*,t} \leq \beta$, where m^* denotes the solution to (3) such that $y_{m^*,t} = 1$. The final result is then obtained by bounding the Lyapunov drift of an auxiliary queue capturing how much Q_t exceeds $\max\left\{\alpha\psi; {}^{V\Delta_E}/_{\beta}\right\}$. Based on the queue update in (2), it is easy to obtain the following corollary.

Corollary 1. *We have the following upper bound of constraint violation (averaged over time):*

$$\alpha - \tfrac{1}{T}\sum_{t=1}^{T}\sum_{m=1}^{M}\mathbb{E}\left[y_{m,t}s_{m,t}\right] = \tfrac{\mathbb{E}[Q_T]}{T} \leq \max\left\{\tfrac{\alpha\psi}{T}; \tfrac{V\Delta_E}{\beta T}\right\} + \tfrac{1}{\sqrt{T}} = \mathcal{O}\left(\tfrac{V}{T} + \tfrac{1}{\sqrt{T}}\right). \quad (6)$$

We observe that the constraint violation is guaranteed to be arbitrarily small when T is sufficiently large. If full SLA compliance (denoted by α') needs to be guaranteed after a finite number T_0 of requests, we can choose α to be slightly larger than α', so that $\max\left\{\tfrac{\alpha\psi}{T_0}; \tfrac{V\Delta_E}{\beta T_0}\right\} + \tfrac{1}{\sqrt{T_0}} \leq \alpha - \alpha'$, which holds for $\alpha = \alpha' + \tfrac{V\Delta_E}{\beta T_0} + \tfrac{1}{\sqrt{T_0}}$ when $\tfrac{\alpha\psi}{T_0} \leq \tfrac{V\Delta_E}{\beta T_0}$. In practice, as we see in our experiments in Section 4, our MESS+ algorithm satisfies the constraint empirically after a relatively small number (e.g., slightly more than a thousand) of requests even if we use α as the SLA requirement directly.

3.2 Cost Optimality

Assumption 2. We assume that both the input request content $\mathbf{a}_t$ and cost $E_{m,t}, \forall m$ are i.i.d. across t, while for the same t, $E_{m,t}$ may be dependent on $\mathbf{a}_t$.

Assumption 3. The predictor training loss $F(\mathbf{z})$ is L-smooth. It also satisfies the Polyak–Łojasiewicz (PL) condition with parameter $\mu > 0$ (and $\mu \leq L$), i.e., $\tfrac{1}{2}\|\nabla F(\mathbf{z})\|^2 \geq \mu\big(F(\mathbf{z}) - F_{\min}\big), \forall \mathbf{z}$, where $F_{\min} := \min_{\mathbf{z}} F(\mathbf{z})$. Its stochastic gradient is unbiased and has a bounded variance of σ^2, i.e., $\mathbb{E}\big[\nabla F(\mathbf{z}, \mathbf{a})\big|\mathbf{z}\big] = \nabla F(\mathbf{z})$ and $\mathbb{E}\big[\,\big\|\nabla F(\mathbf{z}, \mathbf{a}) - \nabla F(\mathbf{z})\big\|^2\,\big|\mathbf{z}\big] \leq \sigma^2$, $\forall \mathbf{z}$, where $\nabla F(\mathbf{z}, \mathbf{a})$ denotes the stochastic gradient of $F(\mathbf{z})$ on sample $\mathbf{a} \sim \mathcal{A}$.

Assumption 2 is commonly used in the Lyapunov drift-plus-penalty framework [17] and Assumption 3 is common in stochastic gradient descent (SGD) convergence analysis. In our experiments, we empirically show that MESS+ also works well in non-i.i.d. settings (Appendix C.4.2).

Theorem 2. *For $\{y_{m,t} : \forall m, t\}$ obtained from the MESS+ algorithm (Algorithm 1), there exists a learning rate schedule $\{\eta_t : \forall t\}$, such that we have*

$$\mathbb{E}\left[\tfrac{1}{T}\sum_{t=1}^{T}\sum_{m=1}^{M} y_{m,t}E_{m,t}\right] \leq E^{\text{OPT}} + \mathcal{O}\left(\tfrac{M}{\sqrt[4]{T}} + \tfrac{1}{V} + MF_{\min}\right), \quad (7)$$

where E^{OPT} is the optimal solution to (1) *that is obtained from an idealized stationary policy which assumes full statistical knowledge of requests in $t \in \{1, \ldots, T\}$.*

The proof of Theorem 2 includes *key novel steps* to capture the joint effect of per-request optimization in (3), the error of the request satisfaction predictor, and the additional cost incurred due to exploration for training the predictor. The latter two aspects do not exist in the framework in [17]. In particular, we first bound the drift-plus-penalty expression and obtain a term $\mathbb{E}\left[Q_t\right] \cdot \mathbb{E}\big[\max_m\{\big|\hat{s}_{m,t} - s_{m,t}\big|\}\big]$ in the bound. Then, we further bound the prediction error $\mathbb{E}\big[\max_m\{\big|\hat{s}_{m,t} - s_{m,t}\big|\}\big]$ using SGD convergence analysis while considering properties of the cross-entropy loss (4), where the number of SGD steps for predictor training (Line 9 of Algorithm 1) is related to the exploration probability $p_t = \mathcal{O}(1/\sqrt[4]{t})$. We also incorporate the cost overhead for exploration that is $\mathcal{O}(M/\sqrt[4]{T})$. Combining these and using the average queue length bound in Theorem 1, we obtain the result.

We have several important observations from Theorem 2. First, combining with Theorem 1, we can confirm that V controls the trade-off between cost optimality and constraint satisfaction, where a larger V boosts cost optimality but slows down constraint satisfaction, and vice versa. Second, the cost optimality gap depends on $MF_{\min}$, where we recall that M is the number of LLMs in the zoo and $F_{\min}$ is the minimum loss that can be obtained for training the request satisfaction predictor. When the predictor is capable, $F_{\min}$ is small, which is what we also observe in the experiments in Section 4. Finally, when we choose $V = \sqrt{T}$, we can observe from Theorems 1 and 2 that, as $T \to \infty$, MESS+ guarantees full SLA satisfaction and approximate cost optimality up to $MF_{\min}$, i.e., the minimum loss of the predictor times the number of LLMs in the zoo.

4 Experiments

We demonstrate the effectiveness of MESS+ across a set of experiments with state-of-the-art LLM benchmarks. Our code is publicly available[5] and we provide full experimental details in the appendix.

4.1 Setup

Language Model Zoo & Benchmarks. Our model zoo is comprised of three models: Llama 3.2 1B (L1B), Llama 3.1 8B (L8B), and Llama 3.3 70B model (L70B). We use the LM-Eval Harness [10] and deploy three reasoning benchmarks (ARC Easy, ARC Challenge, and Winogrande) [5, 21] as well as five Q&A benchmarks (BoolQ, LogiQA, PiQA, SciQ, and SocialIQA) [2, 4, 13, 14, 23]. All benchmarks are evaluated zero-shot. On a per-sample basis, the benchmarks generate binary feedback signals that indicate whether a request has been satisfied. These binary signals are used as labels for training our request satisfaction predictor.

Request Satisfaction Predictor. In line with related work [18], we choose ModernBERT [25] as a transformer backbone and implement a multi-label classifier on top of it, where each label corresponds to whether a model can satisfy a user request. We freeze the transformer parameters and only train the classifier with SGD. The classifier is trained online while running each benchmark as described in our algorithm. We set $c = 0.1$, if not specified otherwise.

User & Service Provider Requirements. For each benchmark, we consider a pre-defined α stating the minimum request satisfaction rate over time. In practice, this is specified in the SLA between the user and service provider. The value of α in our experiments is set individually for every benchmark, based on the capabilities of the models in our zoo. To complete the SLA, a user and service provider need to agree on the cost of an inference service, i.e., they need to negotiate how many SLA violations are acceptable in the beginning. The service provider sets V accordingly, to minimize operating costs over time. Regardless of V, MESS+ will eventually converge towards α, as our theory has shown; V only defines how long it may take. We set $V = 0.0001$ by default. *Since SLA metrics are usually measured over a period of time instead of instantaneously, it is sufficient to satisfy SLA requirements after a pre-defined number of requests.*

Objective. We measure the effectiveness of MESS+ by its ability to meet α at minimal operating cost. In our experiments, we use the per-request energy consumption when querying an LLM (measured in megajoule, MJ) as the cost metric. We also present the model call ratio, i.e., the share of benchmark requests routed to each model.

Baselines. We first look at each individual LLM with regard to their operating cost and request satisfaction capabilities. Then, we compare MESS+ with three adaptive routing baselines, namely

[5]MESS+ code repository: https://github.com/laminair/mess-plus

https://doi.org/10.52202/085713-1804

Table 2: Main results. Performance across three reasoning and five Q&A benchmarks. Green highlights all methods that satisfy the service level requirement α and red all violations. The most cost efficient single model satisfying α is underlined and the most efficient adaptive method is highlighted in **bold**. We report operating costs in megajoule (MJ) energy consumption. For a full overview with more α variations please see the appendix.

Benchmark	ARC Challenge ($\alpha = 50\%$)			ARC Easy ($\alpha = 75\%$)			BoolQ ($\alpha = 80\%$)		
Method	Operating Cost (in MJ)	Request. Satisfaction (in %)	Model Call Ratio (L70B/L8B/L1B)	Operating Cost (in MJ)	Request. Satisfaction (in %)	Model Call Ratio (L70B/L8B/L1B)	Operating Cost (in MJ)	Request. Satisfaction (in %)	Model Call Ratio (L70B/L8B/L1B)
Llama 3.2 1B only	$0.09_{\pm 0.00}$	$37.88_{\pm 5.39}$	0% / 0% / 100%	$0.20_{\pm 0.00}$	$62.76_{\pm 5.37}$	0% / 0% / 100%	$0.14_{\pm 0.00}$	$69.17_{\pm 5.13}$	0% / 0% / 100%
Llama 3.1 8B only	$\underline{0.46}_{\pm 0.00}$	$54.44_{\pm 5.54}$	0% / 100% / 0%	$\underline{0.97}_{\pm 0.00}$	$79.72_{\pm 4.47}$	0% / 100% / 0%	$\underline{0.43}_{\pm 0.00}$	$84.16_{\pm 4.06}$	0% / 100% / 0%
Llama 3.3 70B only	$2.35_{\pm 0.01}$	$60.84_{\pm 5.43}$	100% / 0% / 0%	$4.05_{\pm 0.01}$	$83.12_{\pm 4.16}$	100% / 0% / 0%	$3.40_{\pm 0.00}$	$88.78_{\pm 3.51}$	100% / 0% / 0%
Educated Guessing	$1.00_{\pm 0.09}$	$51.65_{\pm 2.98}$	35% / 31% / 34%	$2.00_{\pm 0.08}$	$74.00_{\pm 4.39}$	31% / 32% / 36%	$1.31_{\pm 0.04}$	$80.47_{\pm 1.08}$	33% / 34% / 33%
RouteLLM [18]	$1.24_{\pm 0.10}$	$51.17_{\pm 2.93}$	50% / 0% / 50%	$4.05_{\pm 0.01}$	$82.54_{\pm 2.12}$	100% / 0% / 0%	$2.96_{\pm 0.04}$	$86.83_{\pm 1.27}$	87% / 0% / 13%
RouterDC [3]	$2.09_{\pm 0.06}$	$60.94_{\pm 2.92}$	88% / 12% / 0%	$3.61_{\pm 0.06}$	$82.30_{\pm 2.60}$	85% / 15% / 0%	$2.14_{\pm 0.05}$	$87.06_{\pm 2.70}$	58% / 42% / 0%
MESS+ (ours)	$\mathbf{0.83}_{\pm 0.07}$	$53.64_{\pm 3.13}$	41% / 41% / 18%	$\mathbf{1.74}_{\pm 0.06}$	$77.06_{\pm 1.76}$	22% / 61% / 18%	$\mathbf{0.90}_{\pm 0.04}$	$82.16_{\pm 1.68}$	31% / 45% / 24%

Benchmark	LogiQA ($\alpha = 40\%$)			PiQA ($\alpha = 78\%$)			SciQ ($\alpha = 96\%$)		
Method	Operating Cost (in MJ)	Request. Satisfaction (in %)	Model Call Ratio (L70B/L8B/L1B)	Operating Cost (in MJ)	Request. Satisfaction (in %)	Model Call Ratio (L70B/L8B/L1B)	Operating Cost (in MJ)	Request. Satisfaction (in %)	Model Call Ratio (L70B/L8B/L1B)
Llama 3.2 1B only	$0.17_{\pm 0.00}$	$27.19_{\pm 4.94}$	0% / 0% / 100%	$0.07_{\pm 0.00}$	$74.05_{\pm 4.87}$	0% / 0% / 100%	$0.10_{\pm 0.00}$	$93.80_{\pm 2.68}$	0% / 0% / 100%
Llama 3.1 8B only	$0.81_{\pm 0.00}$	$29.03_{\pm 5.04}$	0% / 100% / 0%	$\underline{0.36}_{\pm 0.00}$	$79.33_{\pm 4.50}$	0% / 100% / 0%	$\underline{0.44}_{\pm 0.00}$	$97.00_{\pm 1.90}$	0% / 100% / 0%
Llama 3.3 70B only	$\underline{4.11}_{\pm 0.02}$	$49.31_{\pm 5.56}$	100% / 0% / 0%	$1.84_{\pm 0.01}$	$82.70_{\pm 4.20}$	100% / 0% / 0%	$2.23_{\pm 0.02}$	$97.10_{\pm 1.87}$	100% / 0% / 0%
Educated Guessing	$2.51_{\pm 0.09}$	$39.88_{\pm 4.57}$	56% / 21% / 22%	$0.76_{\pm 0.04}$	$78.89_{\pm 1.52}$	34% / 32% / 34%	$0.92_{\pm 0.09}$	$96.51_{\pm 1.49}$	31% / 36% / 32%
RouteLLM [18]	$3.97_{\pm 0.04}$	$47.71_{\pm 3.38}$	98% / 0% / 2%	$1.25_{\pm 0.05}$	$78.35_{\pm 1.42}$	66% / 0% / 34%	$2.16_{\pm 0.04}$	$97.76_{\pm 0.73}$	95% / 0% / 5%
RouterDC [3]	$2.67_{\pm 0.08}$	$47.13_{\pm 3.08}$	70% / 29% / 2%	$1.85_{\pm 0.01}$	$82.34_{\pm 1.33}$	100% / 0% / 0%	$1.90_{\pm 0.07}$	$97.95_{\pm 0.81}$	82% / 18% / 0%
MESS+ (ours)	$\mathbf{2.50}_{\pm 0.09}$	$41.02_{\pm 3.59}$	59% / 17% / 23%	$\mathbf{0.67}_{\pm 0.04}$	$79.20_{\pm 2.58}$	35% / 45% / 19%	$\mathbf{0.83}_{\pm 0.04}$	$96.01_{\pm 2.05}$	27% / 39% / 34%

Benchmark	SocialIQA ($\alpha = 44\%$)			Winogrande ($\alpha = 70\%$)			*Avg. across all Benchmarks* ($\alpha = 66\%$)		
Method	Operating Cost (in MJ)	Request. Satisfaction (in %)	Model Call Ratio (L70B/L8B/L1B)	Operating Cost (in MJ)	Request. Satisfaction (in %)	Model Call Ratio (L70B/L8B/L1B)	Operating Cost (in MJ)	Request. Satisfaction (in %)	Model Call Ratio (L70B/L8B/L1B)
Llama 3.2 1B only	$0.13_{\pm 0.00}$	$41.71_{\pm 5.48}$	0% / 0% / 100%	$0.06_{\pm 0.00}$	$59.67_{\pm 5.45}$	0% / 0% / 100%	$0.12_{\pm 0.00}$	$58.28_{\pm 4.92}$	0% / 0% / 100%
Llama 3.1 8B only	$\underline{0.59}_{\pm 0.00}$	$48.31_{\pm 5.55}$	0% / 100% / 0%	$\underline{0.25}_{\pm 0.00}$	$73.64_{\pm 4.90}$	0% / 100% / 0%	$\underline{0.54}_{\pm 0.00}$	$68.20_{\pm 4.49}$	0% / 100% / 0%
Llama 3.3 70B only	$3.00_{\pm 0.00}$	$48.67_{\pm 5.56}$	100% / 0% / 0%	$1.29_{\pm 0.00}$	$79.08_{\pm 4.52}$	100% / 0% / 0%	$2.91_{\pm 0.01}$	$73.70_{\pm 4.35}$	100% / 0% / 0%
Educated Guessing	$1.22_{\pm 0.06}$	$47.71_{\pm 2.50}$	33% / 32% / 35%	$0.54_{\pm 0.04}$	$70.67_{\pm 3.35}$	35% / 30% / 35%	$1.28_{\pm 0.07}$	$67.47_{\pm 2.73}$	36% / 31% / 33%
RouteLLM [18]	$2.02_{\pm 0.07}$	$44.32_{\pm 2.40}$	65% / 0% / 35%	$1.27_{\pm 0.02}$	$80.82_{\pm 2.53}$	97% / 0% / 3%	$2.04_{\pm 0.05}$	$71.19_{\pm 2.10}$	82% / 0% / 18%
RouterDC [3]	$2.89_{\pm 0.03}$	$46.76_{\pm 2.62}$	95% / 5% / 0%	$1.30_{\pm 0.00}$	$80.86_{\pm 2.49}$	100% / 0% / 0%	$2.11_{\pm 0.04}$	$73.17_{\pm 2.32}$	85% / 15% / 0%
MESS+ (ours)	$\mathbf{0.67}_{\pm 0.04}$	$45.88_{\pm 3.14}$	22% / 38% / 41%	$\mathbf{0.52}_{\pm 0.04}$	$73.57_{\pm 3.14}$	43% / 40% / 17%	$\mathbf{1.08}_{\pm 0.05}$	$68.44_{\pm 2.63}$	34% / 40% / 26%

RouteLLM [18], *RouterDC* [3], and *"educated guessing"*. We configure RouteLLM with their BERT-based router model configuration. Note that RouteLLM only supports routing between two models, so we set the small and large model to L1B and L70B, respectively. RouterDC supports routing between multiple LLMs, i.e., can route our entire model zoo. We adopt the configuration from the RouterDC paper [3]. Additionally, we employ an "educated guessing" baseline that randomly chooses an LLM by assuming the availability of prior knowledge on the probability that each LLM satisfies the request, while conforming to our SLA requirements over time. For all the baselines, we tune available hyperparameters so that the baselines satisfy the SLA requirement while being the most cost efficient. We provide further details on all the baselines in the appendix.

4.2 Results

We divide our evaluations into four main segments and an addendum focused on practical aspects. First, we evaluate at the overall cost optimality objective. Second, we look at the request satisfaction rate. Third, we explore the control dynamics of V. Fourth, we explore the routing overhead and how to train the predictor. We then evaluate the characteristics of MESS+ by looking into sparse user feedback and increasing the number of models in a zoo.

Cost Optimality. The main results are shown in Table 2 where we set individual SLA constraints α per benchmark. Our key observation is that, among all the adaptive methods, *MESS+ is consistently the cost optimal solution for achieving a target request satisfaction rate*, which is due to its effective model choice. While the baselines prefer choosing larger and more expensive LLMs from our zoo, our approach tends to rely more on cost effective and smaller models, while providing satisfactory responses at the rate specified by α. By selecting larger models, the baselines overshoot our SLA requirement at the expense of a notable cost overhead. Overall, MESS+ is about $2\times$ more cost efficient than existing model routing techniques and 20% more efficient than our random baseline that knows average benchmark statistics when routing a request.

Request Satisfaction. For users, it is key that their requests are getting responses with a guaranteed minimum satisfaction rate, so that they can reliably offload tasks to an AI co-pilot. When looking

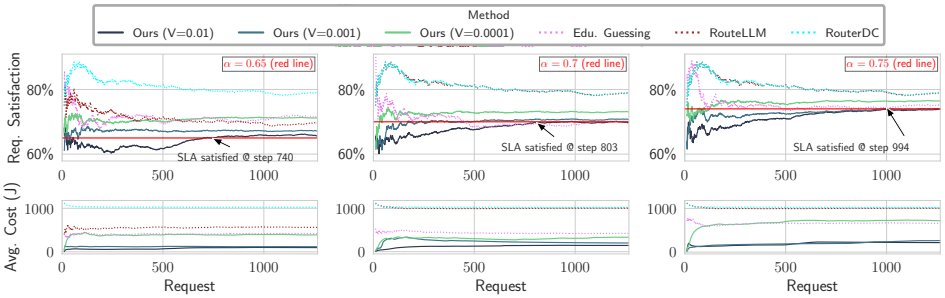


Figure 2: We run several experiments on the Winogrande benchmark with varying α and V configurations to show the request satisfaction and cost dynamics over time. With MESS+, the average request satisfaction rate always converges toward α. We further report the first step at which the highest V value satisfies our SLA requirement. Other benchmarks are in the appendix.

Table 3: Routing overhead compared to the average per inference call cost per benchmark. The routing overhead remains well below 10% of the average inference call cost and depend largely on the sequence length of incoming requests. The average inference call costs per benchmark are computed across all LLMs in a zoo.

Metric	ARC Challenge	ARC Easy	BoolQ	LogiQA	PiQA	SciQ	SocialIQA	Winogrande	Avg.
Avg. Predictor Cost (J)	$5.75_{\pm 0.96}$	$5.69_{\pm 0.56}$	$16.80_{\pm 0.62}$	$41.13_{\pm 0.88}$	$15.38_{\pm 0.70}$	$15.12_{\pm 0.85}$	$26.67_{\pm 0.60}$	$4.89_{\pm 0.72}$	$16.43_{\pm 0.74}$
Avg. LLM Call Cost (J)	$589.97_{\pm 100.18}$	$516.07_{\pm 95.29}$	$236.17_{\pm 39.93}$	$833.21_{\pm 144.34}$	$273.44_{\pm 44.30}$	$263.64_{\pm 48.86}$	$304.45_{\pm 53.03}$	$300.56_{\pm 55.07}$	$414.69_{\pm 72.63}$
Prediction Overhead (%)	$1.01_{\pm 0.28}$	$1.15_{\pm 0.28}$	$7.35_{\pm 1.56}$	$5.12_{\pm 1.19}$	$5.82_{\pm 1.67}$	$5.95_{\pm 1.26}$	$9.08_{\pm 1.96}$	$1.69_{\pm 0.47}$	$4.65_{\pm 1.08}$

at how precisely an adaptive routing technique approaches the SLA requirement α, *MESS+ usually shows the smallest margin, i.e., with MESS+ routing the LLM zoo provides responses that are closely matching the SLA requirement and therefore cost optimal*. All other dynamic routing techniques tend to overshoot α and prefer the strongest model in the zoo to satisfy responses. Our "educated guessing" baseline also undershoots α in some cases, which renders it impractical for minimum service level guarantees, in addition to its requirement of prior statistical knowledge that is usually impractical to obtain. In general, overshooting α naturally means higher per-request operating cost. Practically, MESS+ needs a pre-defined number of steps to converge towards α, which is acceptable from an SLA perspective as discussed earlier.

Effect of V. As discussed in previous sections, V controls the trade-off between the speed of convergence to constraint satisfaction and cost efficiency. Figure 2 shows that choosing a large V leads to longer convergence times for constraint satisfaction, regardless of the value of α. Conversely, a small value of V yields constraint satisfaction within a short amount of time at the expense of increased operating costs. This manifests in the varying operating costs per request over time. When comparing to our baselines, we observe that *MESS+ offers the lowest per-request operating cost among the dynamic routing approaches* by a large margin.

Routing Overhead. Adaptive routing incurs a cost overhead for routing an incoming request to an LLM within a zoo. We measure the cost to make this decision with MESS+ across benchmarks (Table 3). For an incoming request, obtaining satisfaction probabilities from *our predictor incurs a minimal cost overhead* of 4.65% on average, compared to the cost for making an inference call to an LLM. Overall, the cost for making a request satisfaction prediction depends on the length of the incoming request.

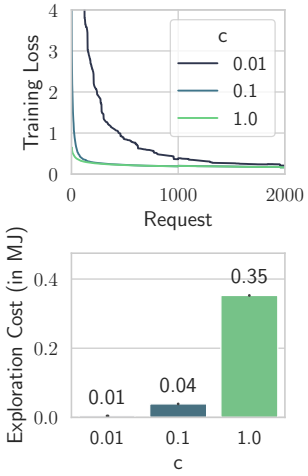


Figure 3: Predictor training performance, averaged across all 8 benchmarks. We control the exploration probability of MESS+ with c. Our predictor learns effectively with a small c already.

Predictor Training. As MESS+ requires online learning of a predictor that predicts the request satisfaction for every LLM in the zoo, we study how the rate of exploration that is parameterized with c affects both the predictor training convergence and cost overhead incurred by exploration and training. Our study shows that *a relatively small value of c already leads to a strong predictor within a short amount of time* (Figure 3), as shown by the training loss. The lower we choose c the faster we move from exploring the model zoo in the beginning towards using the MESS+ objective (3) for model selection decision making.

https://doi.org/10.52202/085713-1804

Table 4: MESS+ provides strong performance even when users provide sparse feedback (i.e., only when a user sends a feedback signal).

Benchmark	ARC C. ($\alpha = 50\%$)		ARC E. ($\alpha = 75\%$)		BoolQ ($\alpha = 20\%$)		LogiQA ($\alpha = 40\%$)	
Feedback Density	20%	100%	20%	100%	20%	100%	20%	100%
Request Satisfaction (in %)	$50.20_{\pm 5.39}$	$50.64_{\pm 2.28}$	$75.10_{\pm 1.12}$	$75.66_{\pm 0.34}$	$20.66_{\pm 3.17}$	$20.34_{\pm 1.78}$	$40.22_{\pm 0.86}$	$40.46_{\pm 0.09}$
Cost (in MJ)	$0.79_{\pm 0.01}$	$0.83_{\pm 0.01}$	$1.69_{\pm 0.01}$	$1.74_{\pm 0.01}$	$0.87_{\pm 0.01}$	$0.90_{\pm 0.01}$	$2.47_{\pm 0.02}$	$2.50_{\pm 0.01}$

Benchmark	PiQA ($\alpha = 78\%$)		SciQ ($\alpha = 96\%$)		SocialIQA ($\alpha = 44\%$)		Winogrande ($\alpha = 70\%$)	
Feedback Density	20%	100%	20%	100%	20%	100%	20%	100%
Request Satisfaction (in %)	$78.48_{\pm 1.18}$	$78.91_{\pm 0.83}$	$96.08_{\pm 1.13}$	$96.11_{\pm 0.62}$	$44.24_{\pm 1.42}$	$44.53_{\pm 0.91}$	$70.97_{\pm 1.71}$	$70.36_{\pm 1.03}$
Cost (in MJ)	$0.64_{\pm 0.01}$	$0.67_{\pm 0.01}$	$0.81_{\pm 0.01}$	$0.83_{\pm 0.01}$	$0.66_{\pm 0.01}$	$0.67_{\pm 0.01}$	$0.50_{\pm 0.01}$	$0.52_{\pm 0.01}$

Sparse User Feedback. MESS+ relies on online user feedback. We evaluate how the performance of MESS+ varies when only a fraction of users actually provides feedback for requests that would normally be used to train our predictor. We compare perfect conditions (full feedback) to sparse feedback with 20% of requests receiving user feedback. Our approach maintains strong performance when introducing sparsity (Table 4). Over time, both the dense feedback and sparse scenario provide strict SLA compliance and incur similar operating cost. The operating cost for the sparse feedback are slightly lower than with dense feedback. This is because of Q receiving fewer updates and a prolonged time to stabilize, which leads to a higher favorability of smaller and cheaper models in the zoo in the beginning (avg. 1.05 MJ vs. 1.08 MJ across 8 benchmarks).

Table 5: Main results with a larger model zoo containing models from the Qwen 2/2.5 family. MESS+ scales well as the number of models grows and shows effective routing capabilities. Detailed results can be found in the appendix.

Category	Mean ($\alpha = 67\%$)		
Subcategory	Operating Cost	Request. Satisfaction	Model Call Ratio (Q32B/Q7B/Q1.5B/Q0.5B)
Qwen2 0.5B	$0.12_{\pm 0.00}$	$54.12_{\pm 45.15}$	0% / 0% / 0% / 100%
Qwen2 1.5B	$0.16_{\pm 0.00}$	$61.13_{\pm 4.79}$	0% / 0% / 100% / 0%
Qwen2 7B	$\underline{0.40}_{\pm 0.00}$	$67.07_{\pm 4.61}$	0% / 100% / 0% / 0%
Qwen2.5 32B	$1.60_{\pm 0.00}$	$70.91_{\pm 4.48}$	100% / 0% / 0% / 0%
Educated Guessing	$0.99_{\pm 0.00}$	$67.02_{\pm 2.32}$	53% / 26% / 11% / 10%
RouteLLM	$1.37_{\pm 0.00}$	$69.01_{\pm 2.31}$	83% / 0% / 0% / 17%
RouterDC	$1.13_{\pm 0.00}$	$69.17_{\pm 2.47}$	63% / 17% / 9% / 11%
MESS+ (ours)	$\mathbf{0.84}_{\pm 0.00}$	$67.55_{\pm 3.23}$	48% / 26% / 10% / 16%

Larger Model Zoo. To demonstrate the scalability of MESS+, we deploy a larger model zoo containing four models: Qwen 2.5 32B (Q32B), Qwen 2 7B (Q7B), Qwen 2 1.5B (Q1.5B), and Qwen 2 0.5B (Q0.5B). The zoo not only is larger but also contains models that have more similar cost characteristics than our LLama3 model zoo. This makes routing in our case more difficult. Our approach exhibits strong performance and strictly maintains SLA compliance under these more challenging conditions (Table 5). Specifically, our routing approach outperforms existing baselines in terms of cost efficiency by a factor of up to $1.6\times$, demonstrating the suitability of MESS+ for providing the most appropriate models for any incoming user request even in larger model zoos.

5 Conclusion

We have presented MESS+, which is a novel theoretically grounded method for automatic model selection in language model zoos. On average, our approach reduces operating costs by $2\times$ compared to well-established adaptive routing baselines in the literature, while still satisfying SLA requirements. Overall, MESS+ shows strong generalization across various state-of-the-art benchmarks and is the first approach to optimizing operating costs while providing rigorous service level guarantees, when serving user requests with a model zoo. Further, our online-learning based approach to model selection removes the need for curating routing preference datasets. Our technique can be particularly useful for cost-aware systems that serve quality-sensitive workloads and require a lower-bound on user satisfaction rate.

While our work shows strong performance across a wide range of tasks, we observe some practical limitations that can be addressed in future work. Our approach expects readily available user satisfaction labels for any request. In reality, user feedback may be only sparingly available. This would require a modified strategy for virtual queue update and zoo exploration based on feedback availability. In addition, our predictor training approach expects complete request satisfaction labels for each model during exploration. Practically, this can be challenging as showing multiple outputs to users and requesting feedback may be confusing. Nevertheless, it is worth pointing out that our work serves as an important foundation for these extensions in the future.

https://doi.org/10.52202/085713-1804

Acknowledgements

This work is supported by the Bavarian Ministry of Economic Affairs, Regional Development and Energy (Grant: DIK0446/01).

References

[1] P. Aggarwal, A. Madaan, A. Anand, S. P. Potharaju, S. Mishra, P. Zhou, A. Gupta, D. Rajagopal, K. Kappaganthu, Y. Yang, S. Upadhyay, M. Faruqui, and M. . Automix: Automatically mixing language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=e6WrwIvgzX.

[2] Y. Bisk, R. Zellers, R. Le bras, J. Gao, and Y. Choi. Piqa: Reasoning about physical commonsense in natural language. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7432–7439, Apr. 2020. ISSN 2159-5399. doi: 10.1609/aaai.v34i05.6239. URL http://dx.doi.org/10.1609/aaai.v34i05.6239.

[3] S. Chen, W. Jiang, B. Lin, J. Kwok, and Y. Zhang. RouterDC: Query-based router by dual contrastive learning for assembling large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=7RQvjayHrM.

[4] C. Clark, K. Lee, M.-W. Chang, T. Kwiatkowski, M. Collins, and K. Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *NAACL*, 2019.

[5] P. Clark, I. Cowhey, O. Etzioni, T. Khot, A. Sabharwal, C. Schoenick, and O. Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*, 2018.

[6] Council of the European Union. Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence and amending Regulations (EC) No 300/2008, (EU) No 167/2013, (EU) No 168/2013, (EU) 2018/858, (EU) 2018/1139 and (EU) 2019/2144 and Directives 2014/90/EU, (EU) 2016/797 and (EU) 2020/1828 (Artificial Intelligence Act)Text with EEA relevance., jul 2024. URL https://eur-lex.europa.eu/eli/reg/2024/1689/oj. Document 32024R1689.

[7] D. Ding, A. Mallick, C. Wang, R. Sim, S. Mukherjee, V. Rühle, L. V. S. Lakshmanan, and A. H. Awadallah. Hybrid LLM: Cost-efficient and quality-aware query routing. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=02f3mUtqnM.

[8] A. Dubey, A. Jauhri, et al. The Llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

[9] C. Fourrier, N. Habib, A. Lozovskaya, K. Szafer, and T. Wolf. Open llm leaderboard v2. https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard, 2024.

[10] L. Gao, J. Tow, et al. A framework for few-shot language model evaluation, Sept. 2021. URL https://doi.org/10.5281/zenodo.5371628.

[11] I. Granite Team. Granite 3.0 language models, Oct. 2024. URL https://github.com/ibm-granite/granite-3.0-language-models/blob/main/paper.pdf.

[12] D. Jiang, X. Ren, and B. Y. Lin. LLM-blender: Ensembling large language models with pairwise ranking and generative fusion. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14165–14178, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.792. URL https://aclanthology.org/2023.acl-long.792/.

[13] M. G. Johannes Welbl, Nelson F. Liu. Crowdsourcing multiple choice science questions. 2017.

 https://doi.org/10.52202/085713-1804

[14] J. Liu, L. Cui, H. Liu, D. Huang, Y. Wang, and Y. Zhang. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, IJCAI-PRICAI-2020, page 3622–3628. International Joint Conferences on Artificial Intelligence Organization, July 2020. doi: 10.24963/ijcai.2020/501. URL http://dx.doi.org/10.24963/ijcai.2020/501.

[15] K. Lu, H. Yuan, R. Lin, J. Lin, Z. Yuan, C. Zhou, and J. Zhou. Routing to the expert: Efficient reward-guided ensemble of large language models. In K. Duh, H. Gomez, and S. Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1964–1974, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.109. URL https://aclanthology.org/2024.naacl-long.109/.

[16] H. Naveed, A. U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman, N. Akhtar, N. Barnes, and A. Mian. A comprehensive overview of large language models, 2023. URL https://arxiv.org/abs/2307.06435.

[17] M. Neely. *Stochastic network optimization with application to communication and queueing systems*. Springer Nature, 2022.

[18] I. Ong, A. Almahairi, V. Wu, W.-L. Chiang, T. Wu, J. E. Gonzalez, M. W. Kadous, and I. Stoica. Routellm: Learning to route llms with preference data. *arXiv preprint arXiv:2406.18665*, 2024.

[19] Qwen, A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei, H. Lin, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Lin, K. Dang, K. Lu, K. Bao, K. Yang, L. Yu, M. Li, M. Xue, P. Zhang, Q. Zhu, R. Men, R. Lin, T. Li, T. Tang, T. Xia, X. Ren, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Wan, Y. Liu, Z. Cui, Z. Zhang, and Z. Qiu. Qwen2.5 technical report, 2024. URL https://arxiv.org/abs/2412.15115.

[20] Reuters. Microsoft deal propels three mile island restart, with key permits still needed. https://www.reuters.com/markets/deals/constellation-inks-power-supply-deal-with-microsoft-2024-09-20/, 2024. [Accessed 24-09-2024].

[21] K. Sakaguchi, R. Le Bras, C. Bhagavatula, and Y. Choi. Winogrande: An adversarial winograd schema challenge at scale. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8732–8740, Apr. 2020. ISSN 2159-5399. doi: 10.1609/aaai.v34i05.6399. URL http://dx.doi.org/10.1609/aaai.v34i05.6399.

[22] S. Samsi, D. Zhao, J. McDonald, B. Li, A. Michaleas, M. Jones, W. Bergeron, J. Kepner, D. Tiwari, and V. Gadepally. From words to watts: Benchmarking the energy costs of large language model inference. In *2023 IEEE High Performance Extreme Computing Conference (HPEC)*, pages 1–9, 2023. doi: 10.1109/HPEC58863.2023.10363447.

[23] M. Sap, H. Rashkin, D. Chen, R. LeBras, and Y. Choi. Socialiqa: Commonsense reasoning about social interactions, 2019. URL https://arxiv.org/abs/1904.09728.

[24] D. Stripelis, Z. Xu, Z. Hu, A. D. Shah, H. Jin, Y. Yao, J. Zhang, T. Zhang, S. Avestimehr, and C. He. TensorOpera router: A multi-model router for efficient LLM inference. In F. Dernoncourt, D. Preoţiuc-Pietro, and A. Shimorina, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 452–462, Miami, Florida, US, Nov. 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-industry.34. URL https://aclanthology.org/2024.emnlp-industry.34/.

[25] B. Warner, A. Chaffin, B. Clavié, O. Weller, O. Hallström, S. Taghadouini, A. Gallagher, R. Biswas, F. Ladhak, T. Aarsen, N. Cooper, G. Adams, J. Howard, and I. Poli. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference, 2024. URL https://arxiv.org/abs/2412.13663.

[26] A. Wettig, A. Gupta, S. Malik, and D. Chen. QuRating: Selecting high-quality data for training language models. In R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett, and F. Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*,

volume 235 of *Proceedings of Machine Learning Research*, pages 52915–52971. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/wettig24a.html.

 https://doi.org/10.52202/085713-1804

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We propose a new algorithm that introduces guarantees for minimum user satisfaction rates in language model zoos while optimizing for operating cost, which can be practical for inference endpoint services.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations are discussed at the end of the conclusion section (Section 5).

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We provide formal proofs for all our theorems in the supplementary material.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Our codebase will be made public (for now: anonymized for review) and is based on the widely used LM-Eval harness for better usability by others. We describe our algorithm in detail in the main paper and provide extensive details on the technical implementation in the appendix.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: All code repositories and benchmarks we use are open-source. Our code (incl. that to reproduce our data) will be made public, if accepted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide a high-level overview of the experimental setup in the main paper and discuss all remaining details in the appendix.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Our experiments have been repeated multiple times and all reported results are based on the mean observations.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We provide full details of our hardware setup in the appendix.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our work adheres to the NeurIPS ethical guidelines.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Our work focuses on cost efficiency. In case of language models, this can have substantial positive effects on the energy consumption, which reduces the environmental burden of AI. In all other regards, our work bears the same risks and safeguards as the underlying models used in the language model zoo.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our work provides the same safeguards as those implemented in the models in our language model zoo.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have credited the authors of all code libraries, datasets, and models used in our paper.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Our code repository will be made openly available once the paper is accepted.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: –

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

https://doi.org/10.52202/085713-1804

Answer: [NA]

Justification: –

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: We have only used LLMs to improve writing on a sentence level.

Appendix

 https://doi.org/10.52202/085713-1804

A Proofs

A.1 Proof of Theorem 1

Let $\gamma := \max\left\{\alpha\psi; \frac{V\Delta_E}{\beta}\right\}$. For any given t, let m^* denote the selected model obtained from the solution to (3), i.e., $y_{m^*,t} = 1$ and $y_{m',t} = 0$ for $m' \neq m^*$.

We first prove that, when $Q_t > \gamma$, we have $\hat{s}_{\tilde{m},t} - \hat{s}_{m^*,t} \leq \beta$, where $\tilde{m} := \arg\max_m \hat{s}_{m,t}$ as defined in Assumption 1. The proof is by contradiction. Suppose $\hat{s}_{\tilde{m},t} - \hat{s}_{m^*,t} > \beta$, then choosing $\tilde{m}$ instead of m^*, i.e., letting $y_{\tilde{m},t} = 1$ and $y_{m^*,t} = 0$, will reduce the second term of the objective (3a) by $Q_t\beta$. The increase in the first term of (3a) due to this change is at most $V\Delta_E$. Since $Q_t > \gamma \geq \frac{V\Delta_E}{\beta}$, we know that this alternative model choice decreases the objective (3a), which contradicts that m^* is the optimal model choice from (3).

From Assumption 1, we then have $\Pr\{s_{m^*,t} = 1\} \geq q > \alpha$ when $Q_t > \gamma$. We also note that the queue arrival at step t is $\alpha - s_{m^*,t} \in (0, 1)$.

Let $Z_t := \max\{0, Q_t - \gamma\}$. We have

$$\begin{aligned}
&\frac{1}{2}\mathbb{E}\left[Z_{t+1}^2 - Z_t^2 \middle| Z_t\right] \\
&= \frac{1}{2}\mathbb{E}\left[\left(\max\{0, Z_t + \alpha - s_{m^*,t}\}\right)^2 - Z_t^2 \middle| Z_t\right] \\
&\leq \frac{1}{2}\mathbb{E}\left[\left(Z_t + \alpha - s_{m^*,t}\right)^2 - Z_t^2 \middle| Z_t\right] \\
&\leq \mathbb{E}\left[Z_t\left(\alpha - s_{m^*,t}\right) + \frac{1}{2}\left(\alpha - s_{m^*,t}\right)^2 \middle| Z_t\right] \\
&\leq \mathbb{E}\left[Z_t\left(\alpha - s_{m^*,t}\right) \middle| Z_t\right] + \frac{1}{2} \\
&\leq Z_t\left(\alpha - q\right) + \frac{1}{2},
\end{aligned}$$

where the last inequality is because, as shown above, $\Pr\{s_{m^*,t} = 1\} \geq q$ when $Q_t > \gamma$ which is equivalent to $Z_t > 0$.

Taking total expectation gives

$$\mathbb{E}\left[Z_{t+1}^2 - Z_t^2\right] \leq -2\mathbb{E}\left[Z_t\right](q - \alpha) + 1 \leq 1.$$

Since $Z_1 = 0$, telescoping gives

$$\mathbb{E}\left[Z_t^2\right] \leq -2\sum_{\tau=1}^{t-1}\mathbb{E}\left[Z_\tau\right](q - \alpha) + (t - 1), \tag{A.1}$$

for $t > 1$.

Then, noting that $q > \alpha$ and $Z_t \geq 0$, by Jensen's inequality, we have

$$\mathbb{E}\left[Z_t\right] \leq \sqrt{\mathbb{E}\left[Z_t^2\right]} \leq \sqrt{t-1} \leq \sqrt{t}, \tag{A.2}$$

for $t \geq 1$.

In addition, from (A.1), we also have

$$2\sum_{\tau=1}^{t-1}\mathbb{E}\left[Z_\tau\right](q - \alpha) \leq (t - 1) - \mathbb{E}\left[Z_t^2\right] \leq t - 1.$$

Therefore, after replacing $t - 1$ with t,

$$\frac{1}{t}\sum_{\tau=1}^{t}\mathbb{E}\left[Z_\tau\right] \leq \frac{1}{2(q - a)}, \tag{A.3}$$

for $t \geq 1$.

The final result then follows by combining $Q_t \leq \gamma + Z_t$ with (A.2) and (A.3), giving the two bounds. □

A.2 Proof of Theorem 2

We prove Theorem 2 by first introducing a few lemmas.

Lemma A.1. *Choosing* $\eta_k = \min\left\{\frac{2}{\mu(k+1)}, \frac{1}{L}\right\}$*, after* $k \geq 1$ *steps of predictor training using SGD with the loss function defined in* (4)*, we have*

$$\mathbb{E}\left[F(\mathbf{z}_k)\right] - F_{\min} \leq \mathcal{O}\left(\frac{1}{k}\right). \tag{A.4}$$

Proof. Let k denote the index of exploration steps. The SGD update of predictor training is

$$\mathbf{z}_{k+1} \leftarrow \mathbf{z}_k - \eta_k \nabla F(\mathbf{z}_k, \mathbf{a}_k) \tag{A.5}$$

Let $\mathbf{g}_k := \nabla F(\mathbf{z}_k, \mathbf{a}_k)$ for convenience.

By L-smoothness, we have

$$\begin{aligned}
\mathbb{E}\left[F(\mathbf{z}_{k+1})\middle|\mathbf{z}_k\right] &\leq F(\mathbf{z}_k) - \eta_k \left\langle \nabla F(\mathbf{z}_k), \mathbb{E}\left[\mathbf{g}_k\middle|\mathbf{z}_k\right]\right\rangle + \frac{L\eta_k^2}{2}\mathbb{E}\left[\left\|\mathbf{g}_k\right\|^2\middle|\mathbf{z}_k\right] \\
&\leq F(\mathbf{z}_k) - \eta_k \left\|\nabla F(\mathbf{z}_k)\right\|^2 + \frac{L\eta_k^2}{2}\left(\left\|\nabla F(\mathbf{z}_k)\right\|^2 + \sigma^2\right) \\
&= F(\mathbf{z}_k) - \left(\eta_k - \frac{L\eta_k^2}{2}\right)\left\|\nabla F(\mathbf{z}_k)\right\|^2 + \frac{L\eta_k^2\sigma^2}{2}.
\end{aligned} \tag{A.6}$$

From PL condition, we have $\left\|\nabla F(\mathbf{z})\right\|^2 \geq 2\mu\left(F(\mathbf{z}) - F_{\min}\right), \forall \mathbf{z}$. When $\eta_k < \frac{2}{L}$, subtracting $F_{\min}$ on both sides of (A.6) and plugging in the PL inequality gives

$$\begin{aligned}
\mathbb{E}\left[F(\mathbf{z}_{k+1}) - F_{\min}\middle|\mathbf{z}_k\right] &\leq \left(F(\mathbf{z}_k) - F_{\min}\right) - 2\mu\left(\eta_k - \frac{L\eta_k^2}{2}\right)\left(F(\mathbf{z}_k) - F_{\min}\right) + \frac{L\eta_k^2\sigma^2}{2} \\
&= \left(1 - 2\mu\eta_k\left(1 - \frac{L\eta_k}{2}\right)\right)\left(F(\mathbf{z}_k) - F_{\min}\right) + \frac{L\eta_k^2\sigma^2}{2}.
\end{aligned} \tag{A.7}$$

Let $\eta_k \leq \frac{1}{L}$. We have

$$\mathbb{E}\left[F(\mathbf{z}_{k+1}) - F_{\min}\middle|\mathbf{z}_k\right] \leq (1 - \mu\eta_k)\left(F(\mathbf{z}_k) - F_{\min}\right) + \frac{L\eta_k^2\sigma^2}{2}. \tag{A.8}$$

Taking total expectation gives

$$\mathbb{E}\left[F(\mathbf{z}_{k+1})\right] - F_{\min} \leq (1 - \mu\eta_k)\left(\mathbb{E}\left[F(\mathbf{z}_k)\right] - F_{\min}\right) + \frac{L\eta_k^2\sigma^2}{2}. \tag{A.9}$$

Let $\mathcal{F}_k := \mathbb{E}\left[F(\mathbf{z}_k)\right] - F_{\min}$. We have

$$\begin{aligned}
&\mathcal{F}_2 \leq (1 - \mu\eta_1)\mathcal{F}_1 + \frac{L\eta_1^2\sigma^2}{2} \\
&\mathcal{F}_3 \leq (1 - \mu\eta_1)(1 - \mu\eta_2)\mathcal{F}_1 + \frac{L\sigma^2((1 - \mu\eta_2)\eta_1^2 + \eta_2^2)}{2} \\
&\mathcal{F}_4 \leq (1 - \mu\eta_1)(1 - \mu\eta_2)(1 - \mu\eta_3)\mathcal{F}_1 + \frac{L\sigma^2((1 - \mu\eta_3)(1 - \mu\eta_2)\eta_1^2 + (1 - \mu\eta_3)\eta_2^2 + \eta_3^2)}{2} \\
&\ldots
\end{aligned}$$

https://doi.org/10.52202/085713-1804

Therefore,

$$\mathcal{F}_k \leq \mathcal{F}_1 \prod_{\kappa=1}^{k-1}(1-\mu\eta_\kappa) + \frac{L\sigma^2 \sum_{\kappa=1}^{k-1} \eta_\kappa^2 \prod_{\kappa'=\kappa+1}^{k-1}(1-\mu\eta_{\kappa'})}{2}. \tag{A.10}$$

Recall that $\eta_k = \min\left\{\frac{2}{\mu(k+1)}, \frac{1}{L}\right\}$. We note that, due to L-smoothness, we have $\mu \leq L$, because otherwise the L-smoothness contradicts with the PL condition. Therefore, $\frac{2L}{\mu} \geq 2$. We have $\eta_k = \frac{1}{L}$ when $k \leq \tilde{k} := \left\lfloor \frac{2L}{\mu} - 1 \right\rfloor$. For $k > \tilde{k}$, we have $\eta_k = \frac{2}{\mu(k+1)}$, and in this case, $1 - \mu\eta_k = \frac{k-1}{k+1}$.

We first consider the second term of (A.10). For $k \leq \tilde{k} + 1$,

$$G(k) := \sum_{\kappa=1}^{k-1} \eta_\kappa^2 \prod_{\kappa'=\kappa+1}^{k-1}(1-\mu\eta_{\kappa'}) = \sum_{\kappa=1}^{k-1} \frac{1}{L^2}\left(1-\frac{\mu}{L}\right)^{\kappa-1} = \frac{1}{L^2} \cdot \frac{1-\left(1-\frac{\mu}{L}\right)^{k-2}}{\frac{\mu}{L}} \leq \frac{1}{L\mu}.$$

Let $k_0 := \tilde{k} + 1$. From the above, we have $G(k_0) \leq \frac{1}{L\mu}$. For $k > k_0$, we note that

$$\begin{aligned} G(k+1) &= (1-\mu\eta_k)G(k) + \eta_k^2 \\ &= \left(1 - \frac{2}{k+1}\right) G(k) + \frac{4}{\mu^2(k+1)^2} \\ &= \frac{k-1}{k+1} \cdot G(k) + \frac{4}{\mu^2(k+1)^2}. \end{aligned}$$

Thus,

$$\begin{aligned} G(k_0+1) &\leq \frac{\tilde{k}}{L\mu(k_0+1)} + \frac{4}{\mu^2(k_0+1)^2} \\ G(k_0+2) &\leq \frac{\tilde{k}k_0}{L\mu(k_0+1)(k_0+2)} + \frac{4k_0}{\mu^2(k_0+1)^2(k_0+2)} + \frac{4}{\mu^2(k_0+2)^2} \\ G(k_0+3) &\leq \frac{\tilde{k}k_0}{L\mu(k_0+2)(k_0+3)} + \frac{4k_0}{\mu^2(k_0+1)(k_0+2)(k_0+3)} + \frac{4(k_0+1)}{\mu^2(k_0+2)^2(k_0+3)} \\ &\quad + \frac{4}{\mu^2(k_0+3)^2} \\ G(k_0+4) &\leq \frac{\tilde{k}k_0}{L\mu(k_0+3)(k_0+4)} + \frac{4k_0}{\mu^2(k_0+1)(k_0+3)(k_0+4)} + \frac{4(k_0+1)}{\mu^2(k_0+2)(k_0+3)(k_0+4)} \\ &\quad + \frac{4(k_0+2)}{\mu^2(k_0+3)^2(k_0+4)} + \frac{4}{\mu^2(k_0+4)^2} \\ &\cdots \end{aligned}$$

For general $k > k_0$, after upper bounding some terms, we have

$$\begin{aligned} G(k) &\leq \frac{\tilde{k}k_0}{L\mu(k-1)k} + \frac{1}{\mu^2}\sum_{\kappa=k_0}^{k-2} \frac{4\kappa}{(\kappa+1)(k-1)k} + \frac{4}{\mu^2k^2} \\ &\leq \frac{\tilde{k}k_0}{L\mu(k-1)k} + \frac{1}{\mu^2}\sum_{\kappa=k_0}^{k-2} \frac{4}{(k-1)k} + \frac{4}{\mu^2k^2} \\ &\leq \frac{\tilde{k}k_0}{L\mu(k-1)k} + \frac{4}{\mu^2k} \\ &= \mathcal{O}\left(\frac{1}{k}\right), \end{aligned} \tag{A.11}$$

where the last equality is because $\tilde{k}$ and k_0 are constants as they only depend on L and μ.

Next, consider the first term of (A.10). When $k \leq k_0$,

$$H(k) := \prod_{\kappa=1}^{k-1}(1-\mu\eta_\kappa) = \left(1-\frac{\mu}{L}\right)^{k-1}.$$

For $k > k_0$, we have

$$\begin{aligned}
H(k_0+1) &= H(k_0)\cdot\frac{\tilde{k}}{k_0+1}\\
H(k_0+2) &= H(k_0)\cdot\frac{\tilde{k}k_0}{(k_0+1)(k_0+2)}\\
H(k_0+3) &= H(k_0)\cdot\frac{\tilde{k}k_0}{(k_0+2)(k_0+3)}\\
H(k_0+4) &= H(k_0)\cdot\frac{\tilde{k}k_0}{(k_0+3)(k_0+4)}\\
&\cdots
\end{aligned}$$

For general $k \geq k_0$,

$$H(k) = \left(1-\frac{\mu}{L}\right)^{\tilde{k}}\cdot\frac{\tilde{k}k_0}{(k-1)k} = \mathcal{O}\left(\frac{1}{k^2}\right), \tag{A.12}$$

because $\tilde{k}$ and k_0 are constants as they only depend on L and μ.

Combining (A.11) and (A.12) with (A.10), we obtain

$$\mathcal{F}_k = \mathbb{E}\left[F(\mathbf{z}_k)\right] - F_{\min} \leq \mathcal{O}\left(\frac{1}{k}\right). \tag{A.13}$$

□

Lemma A.2. *Choosing $\eta_k = \min\left\{\frac{2}{\mu(k+1)}, \frac{1}{L}\right\}$, for any $t \geq 1$, assume that after t requests, k steps of predictor training has occurred. We have*

$$\mathbb{E}\left[\max_m |\hat{s}_{m,t} - s_{m,t}|\right] \leq MF_{\min} + \mathcal{O}\left(\frac{M}{k}\right). \tag{A.14}$$

Proof. For the ease of discussion, let ϵ denote the upper bound of $\mathbb{E}\left[F(\mathbf{z}_k)\right] - F_{\min}$ so that $\epsilon = \mathcal{O}\left(\frac{1}{k}\right)$ according to Lemma A.1.

When $\mathbb{E}\left[F(\mathbf{z}_k)\right] - F_{\min} \leq \epsilon$, from (4) and noting that the cross entropy is non-negative, we have

$$\begin{aligned}
&\mathbb{E}\left[\max_m \left\{-s_{m,t}\log\hat{s}_{m,t} - (1-s_{m,t})\log(1-\hat{s}_{m,t})\right\}\right]\\
&\leq \mathbb{E}\left[\sum_{m=1}^{M}\left(-s_{m,t}\log\hat{s}_{m,t} - (1-s_{m,t})\log(1-\hat{s}_{m,t})\right)\right]\\
&\leq M\mathbb{E}\left[F(\mathbf{z}_k)\right]\\
&\leq MF_{\min} + \epsilon M.
\end{aligned} \tag{A.15}$$

For arbitrary sample $\mathbf{a}_t$ and model m, let

$$\Gamma := -s_{m,t}\log\hat{s}_{m,t} - (1-s_{m,t})\log(1-\hat{s}_{m,t}).$$

We consider two cases as follows.

When $s_{m,t} = 0$, we have

$$\Gamma = -\log(1-\hat{s}_{m,t})$$

https://doi.org/10.52202/085713-1804

$$\Rightarrow \hat{s}_{m,t} = 1 - e^{-\Gamma}$$
$$\Rightarrow |\hat{s}_{m,t} - s_{m,t}| = 1 - e^{-\Gamma}.$$

When $s_{m,t} = 1$, we have

$$\Gamma = -\log \hat{s}_{m,t}$$
$$\Rightarrow 1 - \hat{s}_{m,t} = 1 - e^{-\Gamma}$$
$$\Rightarrow |\hat{s}_{m,t} - s_{m,t}| = 1 - e^{-\Gamma}.$$

Noting the elementary inequality $e^{-\Gamma} \geq 1 - \Gamma$, we obtain

$$|\hat{s}_{m,t} - s_{m,t}| \leq \Gamma.$$

Because this relation holds for any sample and the corresponding Γ defined on the sample, the expectation of the cross-entropy loss cannot be smaller than the expectation of the absolute difference. Combining with (A.15), we have

$$\mathbb{E}\left[\max_m |\hat{s}_{m,t} - s_{m,t}|\right] \leq MF_{\min} + \epsilon M = MF_{\min} + \mathcal{O}\left(\frac{M}{k}\right). \tag{A.16}$$

□

Lemma A.3. *Let k denote the random variable of the number of predictor training steps after processing t requests. We have*

$$\mathbb{E}[k] \leq \mathcal{O}\left(t^{\frac{3}{4}}\right) \text{ and } \mathbb{E}\left[\frac{1}{k}\right] \leq \mathcal{O}\left(\frac{1}{t^{\frac{3}{4}}}\right). \tag{A.17}$$

Proof. Recall that $X_t \sim \text{Bernoulli}(p_t)$ is an indicator denoting whether an SGD step for predictor training occurs when processing request t. In the following, we assume that k is the total number of SGD steps after processing t requests. We have

$$\lambda := \mathbb{E}[k] = \mathbb{E}\left[\sum_{\tau=1}^{t} X_\tau\right] = \sum_{\tau=1}^{t} p_\tau = \sum_{\tau=1}^{t} \min\left(1, \frac{c}{\sqrt[4]{\tau}}\right) = \Theta\left(t^{\frac{3}{4}}\right). \tag{A.18}$$

This proves the first result.

Considering $\frac{1}{k}$, we note that

$$\mathbb{E}\left[\frac{1}{k}\right] = \mathbb{E}\left[\frac{1}{k} \cdot \mathbb{1}_{k \geq \lambda/2}\right] + \mathbb{E}\left[\frac{1}{k} \cdot \mathbb{1}_{k < \lambda/2}\right] \leq \frac{2}{\lambda} + \mathbb{E}\left[\frac{1}{k} \cdot \mathbb{1}_{k < \lambda/2}\right], \tag{A.19}$$

where $\mathbb{1}_{\mathcal{C}}$ is an indicator function of whether the condition $\mathcal{C}$ holds.

We now consider the last term in (A.19). The multiplicative Chernoff bound shows that

$$\Pr\{k \leq (1-\delta)\lambda\} \leq e^{-\frac{\delta^2 \lambda}{2}},$$

for $0 < \delta < 1$. Choosing $\delta = \frac{1}{2}$ gives

$$\Pr\left\{k \leq \frac{\lambda}{2}\right\} \leq e^{-\frac{\lambda}{8}}.$$

Because $k \geq 1$,

$$\mathbb{E}\left[\frac{1}{k} \cdot \mathbb{1}_{k < \lambda/2}\right] \leq \mathbb{E}\left[\mathbb{1}_{k < \lambda/2}\right] \leq e^{-\frac{\lambda}{8}} = \frac{1}{e^{\frac{\lambda}{8}}} \leq \frac{1}{1 + \frac{\lambda}{8}}, \tag{A.20}$$

where the last inequality is due to the elementary relation that $e^x \geq 1 + x$ for any x.

Combining (A.18), (A.19), and (A.20), we obtain

$$\mathbb{E}\left[\frac{1}{k}\right] \leq \mathcal{O}\left(\frac{1}{t^{\frac{3}{4}}}\right). \tag{A.21}$$

□

Based on these lemmas, we are now ready to prove Theorem 2.

Proof of Theorem 2. We first consider any t such that $X_t = 0$, i.e., no exploration or predictor training. For the ease of presentation, let m^* denote the optimal solution to (3) for some given t, i.e., $y_{m^*,t} = 1$ and $y_{m',t} = 0$ for $m' \neq m^*$, where t is inferred from the context. We consider the following Lyapunov drift of the queue length:

$$\begin{aligned}
\frac{1}{2}\mathbb{E}\left[Q_{t+1}^2 - Q_t^2 \middle| Q_t\right] &= \frac{1}{2}\mathbb{E}\left[\left(\max\{0, Q_t + \alpha - s_{m^*,t}\}\right)^2 - Q_t^2 \middle| Q_t\right] \\
&\leq \frac{1}{2}\mathbb{E}\left[\left(Q_t + \alpha - s_{m^*,t}\right)^2 - Q_t^2 \middle| Q_t\right] \\
&\leq \frac{1}{2}\mathbb{E}\left[2Q_t(\alpha - s_{m^*,t}) + 1 \middle| Q_t\right] \\
&= \mathbb{E}\left[Q_t(\alpha - s_{m^*,t} + \hat{s}_{m^*,t} - \hat{s}_{m^*,t}) \middle| Q_t\right] + \frac{1}{2} \\
&= \mathbb{E}\left[Q_t(\alpha - \hat{s}_{m^*,t}) \middle| Q_t\right] + \mathbb{E}\left[Q_t(\hat{s}_{m^*,t} - s_{m^*,t}) \middle| Q_t\right] + \frac{1}{2} \\
&\leq \mathbb{E}\left[Q_t(\alpha - \hat{s}_{m^*,t}) \middle| Q_t\right] + Q_t\mathbb{E}\left[\max_m |\hat{s}_{m,t} - s_{m,t}|\right] + \frac{1}{2}, \qquad \text{(A.22)}
\end{aligned}$$

where the second inequality follows from expanding the square and $(\alpha - s_{m^*,t})^2 \leq 1$.

Let $\{y_{m,t}^{\text{OPT}}, \forall m\}$ denote the result of an optimal stationary policy to (1). For our online decision-making problem (3), we get

$$\begin{aligned}
&\mathbb{E}\left[VE_{m^*,t} + Q_t\left(\alpha - \hat{s}_{m^*,t}\right) \middle| Q_t\right] \\
&\leq \mathbb{E}\left[V \cdot \sum_{m=1}^{M} y_{m,t}^{\text{OPT}} E_{m,t} + Q_t \sum_{m=1}^{M} y_{m,t}^{\text{OPT}}\left(\alpha - \hat{s}_{m,t}\right) \middle| Q_t\right] \\
&= \mathbb{E}\left[V \cdot \sum_{m=1}^{M} y_{m,t}^{\text{OPT}} E_{m,t} + Q_t\left(\sum_{m=1}^{M} y_{m,t}^{\text{OPT}}(\alpha - \hat{s}_{m,t} + s_{m,t} - s_{m,t})\right) \middle| Q_t\right] \\
&= V\mathbb{E}\left[\sum_{m=1}^{M} y_{m,t}^{\text{OPT}} E_{m,t}\right] + Q_t\mathbb{E}\left[\sum_{m=1}^{M} y_{m,t}^{\text{OPT}}(s_{m,t} - \hat{s}_{m,t})\right] + Q_t\mathbb{E}\left[\sum_{m=1}^{M} y_{m,t}^{\text{OPT}}(\alpha - s_{m,t})\right] \\
&\leq VE^{\text{OPT}} + Q_t\mathbb{E}\left[\max_m\{\left|\hat{s}_{m,t} - s_{m,t}\right|\}\right], \qquad \text{(A.23)}
\end{aligned}$$

where the last inequality is because the optimal stationary policy satisfies the constraint (1b) and the cost is minimized when the constraint holds with equality, thus $\mathbb{E}\left[\sum_{m=1}^{M} y_{m,t}^{\text{OPT}}(\alpha - s_{m,t})\right] = 0$.

Combining (A.22) and (A.23), we obtain

$$\begin{aligned}
&\frac{1}{2}\mathbb{E}\left[Q_{t+1}^2 - Q_t^2 \middle| Q_t\right] + V\mathbb{E}\left[\sum_{m=1}^{M} y_{m,t}E_{m,t} \middle| Q_t\right] \\
&\leq \mathbb{E}\left[Q_t(\alpha - \hat{s}_{m^*,t}) \middle| Q_t\right] + Q_t\mathbb{E}\left[\max_m |\hat{s}_{m,t} - s_{m,t}|\right] + \frac{1}{2} + V\mathbb{E}\left[\sum_{m=1}^{M} y_{m,t}E_{m,t} \middle| Q_t\right] \\
&= \mathbb{E}\left[VE_{m^*,t} + Q_t\left(\alpha - \hat{s}_{m^*,t}\right) \middle| Q_t\right] + Q_t\mathbb{E}\left[\max_m |\hat{s}_{m,t} - s_{m,t}|\right] + \frac{1}{2} \\
&\leq VE^{\text{OPT}} + Q_t\mathbb{E}\left[\max_m\{\left|\hat{s}_{m,t} - s_{m,t}\right|\}\right] + Q_t\mathbb{E}\left[\max_m |\hat{s}_{m,t} - s_{m,t}|\right] + \frac{1}{2}
\end{aligned}$$

https://doi.org/10.52202/085713-1804

$$= VE^{\text{OPT}} + 2Q_t\mathbb{E}\left[\max_m\{\left|\hat{s}_{m,t} - s_{m,t}\right|\}\right] + \frac{1}{2}.$$

Taking total expectation, we obtain

$$\begin{aligned}&\frac{1}{2}\mathbb{E}\left[Q_{t+1}^2 - Q_t^2\right] + V\mathbb{E}\left[\sum_{m=1}^{M} y_{m,t}E_{m,t}\right]\\ &\leq VE^{\text{OPT}} + 2\mathbb{E}\left[Q_t\right]\cdot\mathbb{E}\left[\max_m\{\left|\hat{s}_{m,t} - s_{m,t}\right|\}\right] + \frac{1}{2}.\end{aligned}$$

Therefore,

$$\begin{aligned}&\mathbb{E}\left[\sum_{m=1}^{M} y_{m,t}E_{m,t}\right]\\ &\leq E^{\text{OPT}} + \frac{2}{V}\mathbb{E}\left[Q_t\right]\cdot\mathbb{E}\left[\max_m\{\left|\hat{s}_{m,t} - s_{m,t}\right|\}\right] + \frac{1}{2V} + \frac{1}{2V}\mathbb{E}\left[Q_t^2 - Q_{t+1}^2\right]\\ &\leq E^{\text{OPT}} + \frac{2}{V}\mathbb{E}\left[Q_t\right]\cdot\left(\mathbb{E}\left[\mathcal{O}\left(\frac{M}{k}\right)\right] + MF_{\min}\right) + \frac{1}{2V} + \frac{1}{2V}\mathbb{E}\left[Q_t^2 - Q_{t+1}^2\right]\\ &\leq E^{\text{OPT}} + \frac{2}{V}\left((\gamma+\sqrt{t})\mathcal{O}\left(\frac{M}{t^{\frac{3}{4}}}\right) + MF_{\min}\mathbb{E}\left[Q_t\right]\right) + \frac{1}{2V} + \frac{1}{2V}\mathbb{E}\left[Q_t^2 - Q_{t+1}^2\right],\end{aligned}$$

where the second inequality uses Lemma A.2 and the last inequality uses Theorem 1 and $\gamma := \max\left\{\alpha\psi; \frac{V\Delta_E}{\beta}\right\}$.

For any t with $X_t = 1$, we note that the maximum cost due to exploration when processing such requests is $ME_{\max}$.

We now include all t where either $X_t = 0$ or $X_t = 1$. Then,

$$\begin{aligned}&\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\sum_{m=1}^{M} y_{m,t}E_{m,t}\right]\\ &\leq E^{\text{OPT}} + \frac{2}{V}\left(\frac{1}{T}\sum_{t=1}^{T}(\gamma+\sqrt{t})\mathcal{O}\left(\frac{M}{t^{\frac{3}{4}}}\right) + \frac{MF_{\min}}{T}\sum_{t=1}^{T}\mathbb{E}\left[Q_t\right]\right) + \frac{1}{2V} + \frac{1}{2VT}\mathbb{E}\left[Q_1^2 - Q_{t+1}^2\right]\\ &\quad + \frac{\mathbb{E}\left[K\right]ME_{\max}}{T}\\ &\leq E^{\text{OPT}} + \mathcal{O}\left(\frac{1}{T}\sum_{t=1}^{T}\frac{M}{t^{\frac{3}{4}-\frac{1}{2}}} + MF_{\min} + \frac{1}{V} + \frac{M}{T^{\frac{1}{4}}}\right)\\ &= E^{\text{OPT}} + \mathcal{O}\left(\frac{M}{\sqrt[4]{T}} + MF_{\min} + \frac{1}{V}\right),\end{aligned}$$

where the second inequality uses Theorem 1 and Lemma A.3. □

B Experimental Details

In this section we provide full details for our experimental design.

Datasets & Benchmarks. We use the LM-Eval Harness as a basis for all of our evaluations. We use a zero-shot setting for each benchmark. Our evaluations are based on the default configurations provided by LM Eval. We use vLLM v0.8.4 as the inference backend for LM Eval.

LLM Zoo. Our zoo coniststs of three Llama models, namely Llama 3.2 1B, Llama 3.1 8B, and Llama 3.3 70B. We use publicly available model checkpoints available on the HuggingFace Hub, specifically, `meta-llama/Llama-3.2-1B-Instruct`, `unsloth/Meta-Llama-3.1-8B-Instruct-bnb-4bit`, and `unsloth/Llama-3.3-70B-Instruct-bnb-4bit`.

Request Satisfaction Predictor. We design a predictor model based on the ModernBert transformer and implement a classification head on top of it. Before we pass the final-BERT-layer outputs into the classification layer, we pool the outputs and use a dropout (= 0.1) for better training effectiveness. We first have a linear layer, followed by a layer norm operation, followed by a ReLU activation, another dropout (= 0.1), and then a final linear layer. We use a sigmoid function to compute the classifier logits. For each benchmark in our main paper, we ran a hyperparameter sweep to identify the most effective hyperparameter combinations. They are listed in Table 6.

Table 6: Predictor model hyperparameter configurations across benchmarks.

Benchmark	Learning Rate	Weight Decay	Momentum	Max. Seq. Len.	Dropout
ARC Challenge	0.0606	0.01	0.90	256	0.1
ARC Easy	0.0826	0.01	0.90	256	0.1
BoolQ	0.0767	0.01	0.95	128	0.1
LogiQA	0.0272	0.01	0.90	256	0.1
PiQA	0.0367	0.01	0.90	64	0.1
SciQ	0.0596	0.01	0.95	64	0.1
SocialIQA	0.0542	0.01	0.95	64	0.1
Winogrande	0.0660	0.01	0.90	64	0.1

Random Baseline with Constraint Satisfaction. We implement a random baseline that follows constraint satisfaction, or in other words, SLA compliance. We use the following implementation to facilitate the baseline.

```
def calculate_probabilities(model_accuracies: list, alpha: float):
    """
    Function to compute a priori probabilities for a baseline
    model selection process that provides SLA compliance over time.
    """
    accuracies = np.array(model_accuracies)
    n = len(accuracies)

    # Check if possible
    if alpha > max(accuracies):
        raise ValueError("Alpha too high")

    p = np.ones(n) / n

    for _ in range(5000):
        current_acc = np.dot(p, accuracies)

        if current_acc >= alpha - 1e-6:
            return p

        # Simple update
        for i in range(n):
            if accuracies[i] > current_acc:
                p[i] *= 1.01  # Increase good models
```


https://doi.org/10.52202/085713-1804

```
            else:
                p[i] *= 0.99  # Decrease bad models

        # Normalize
        p = p / np.sum(p)


    idx_sorted = np.argsort(accuracies)[::-1]

    # Calculate minimum probability for best model
    best_acc = accuracies[idx_sorted[0]]
    worst_acc = accuracies[idx_sorted[-1]]

    # Start with minimum probabilities for all
    min_prob = 1e-10
    p = np.full(n, min_prob)
    remaining = 1.0 - n * min_prob

    # Distribute remaining probability
    for i in range(n):
        idx = idx_sorted[i]

        if i == n - 1:
            p[idx] += remaining
        else:
            # Give more to better models
            weight = (accuracies[idx] - worst_acc) / (best_acc - worst_acc)
            allocation = remaining * weight * 0.8
            p[idx] += allocation
            remaining -= allocation

    p = p / np.sum(p)

    return p
```

RouteLLM. To reproduce the RouteLLM results, we integrated the RouteLLM controller into our existing evaluation pipeline. RouteLLM supports only two models: a weak and a strong model. We configured these as the Llama 3.2 1B (weak) and 70B (strong) models, respectively. We chose to use the BERT-based router of RouteLLM. We swept the routing threshold from 0.1 to 0.9 in increments of 0.1. As shown in Table 7, RouteLLM requires careful tuning of the routing threshold to achieve desired performance, which lacks a direct mapping to user-specified service level requirements.

Table 7: Mapping from MESS+ α to RouteLLM decision threshold.

Dataset	α_1	Thresh_1	α_2	Thresh_2	α_3	Thresh_3
ARC Challenge	0.7	0.5	0.6	0.6	0.7	0.4
ARC Easy	0.6	0.8	0.7	0.65	0.6	0.75
BoolQ	0.5	0.8	0.5	0.85	0.6	0.7
LogiQA	0.7	0.3	0.5	0.45	0.5	0.4
PiQA	0.6	0.75	0.5	0.81	0.6	0.78
SciQ	0.5	0.97	0.6	0.95	0.5	0.96
SocialIQA	0.7	0.44	0.7	0.46	0.7	0.42
Winogrande	0.5	0.75	0.6	0.65	0.5	0.7

RouterDC. To reproduce RouterDC, we trained its routing module on our benchmark tasks. The router encodes each query using the pretrained encoder `microsoft/deberta-v3-base` and compares its representation to a set of trainable expert embeddings, one for each model in the ensemble. Cosine similarity is used to produce a logit vector over the experts, and the router is optimized to prefer more accurate models via contrastive losses. In our setup, the candidate experts were Llama 3 Instruct models with 1B, 8B, and 70B parameters. Training was performed for 1,000 steps using the AdamW optimizer, with a batch size of 64 and a learning rate of 5×10^{-5}. The training set consisted

of 100 queries from each of ten benchmarks, totaling 1,000 samples. For evaluation, we load the trained checkpoint and integrate the router into our existing evaluation pipeline.

Code. Our code base is made fully public on GitHub. It can be found here: *https://github.com/laminair/mess-plus*

Evaluation Setup. While our experiments can be run on a single GPU (with 80GB VRAM), we conducted our experiments using 2 H100 GPUs. We distribute the LLMs and the predictor as follows: The small and medium sized LLMs along with the predictor are located on 1 GPU and the large model is placed on the other GPU. We repeat each experiment with three different random seeds ($[42, 43, 44]$). Since we query the LLMs sequentially, we can capture their individual energy consumption. When doing parallel calls, it is necessary to place each LLM on a separate GPU and configure the Zeus monitor to properly return the energy statistics for each model.

C Additional Evaluations

C.1 Additional Results Related to Our Main Findings

Our experimental results demonstrate that the parameter V, which controls the priority given to cost efficiency in the MESS+ routing algorithm, exhibits significant influence on both energy consumption and performance metrics across multiple benchmarks (Tables 8 and 9). In the main results with the standard V value, MESS+ achieves remarkable energy efficiency with an average operating cost of 1.08 MJ while maintaining satisfactory performance (68.44% request satisfaction) across all benchmarks. When reducing V to 0.0001, thereby decreasing the emphasis on energy efficiency, we observe a 65.7% increase in operating costs to 1.79 MJ with only a marginal improvement in performance to 69.16%. Further reducing V to 0.00001 yields an additional cost increase to 1.88 MJ (74.1% higher than the standard configuration) while performance improves only slightly to 69.41%. These diminishing returns highlight the effectiveness of our approach in balancing the performance-efficiency trade-off. Notably, the distribution of model calls shifts substantially as V decreases—the utilization of the 70B model increases from 34% with standard V to 54% with V=0.00001, while mid-sized 8B model usage decreases from 40% to 23%, indicating a clear preference for higher-capacity models when efficiency constraints are relaxed. Individual benchmarks exhibit varying sensitivities to the V parameter; LogiQA shows the most substantial performance gain (41.02% to 43.89%) with decreased V values, while SciQ maintains relatively stable performance ($\approx 96\%$) despite significant variations in model call distribution. The ARC Easy benchmark demonstrates one of the most dramatic cost increases, from 1.74 MJ to 5.39 MJ at the lowest V value, emphasizing how routing decisions can substantially impact energy consumption for specific task types. Even with reduced emphasis on efficiency, MESS+ maintains competitive or superior performance compared to alternative methods like RouteLLM and RouterDC while consuming less energy on average. These findings underscore the flexibility of our approach in accommodating different deployment scenarios where either performance or energy efficiency might be prioritized, while consistently outperforming baseline single-model approaches for the same levels of request satisfaction.

C.2 The relationship between α and V

Our experimental evaluation across multiple reasoning benchmarks (ARC Challenge, ARC Easy, BoolQ, LogiQA, PiQA, SciQ, and SocialIQA) exhibits the exact relationship between α and V that we show in our theoretical analysis. The results demonstrate that lower V values ($V = 0.0001$) consistently achieve higher request satisfaction rates while incurring greater computational costs, exhibiting a slower convergence to stability but ultimately reaching higher performance plateaus that can satisfy more demanding Service Level Agreement (SLA) thresholds. Conversely, higher V values ($V = 0.01$) prioritize cost efficiency, resulting in significantly lower average costs, faster initial convergence, but ultimately lower satisfaction plateaus that are very close to α. Medium V values ($V = 0.001$) strike a compelling balance, offering reasonable satisfaction rates with moderate computational investment. Notably, our Winogrande analysis illuminates the explicit relationship between SLA satisfaction timing and both α thresholds and V values, with higher α requirements (0.65, 0.7, and 0.75) correspondingly satisfied at later request points (steps 740, 803, and 994, respectively). When compared against baseline methods, our approach approaches provides similar satisfaction levels like RouterDC while maintaining substantially lower computational costs, demonstrating superior efficiency in the satisfaction-cost frontier. These findings underpin that V provides an

 https://doi.org/10.52202/085713-1804

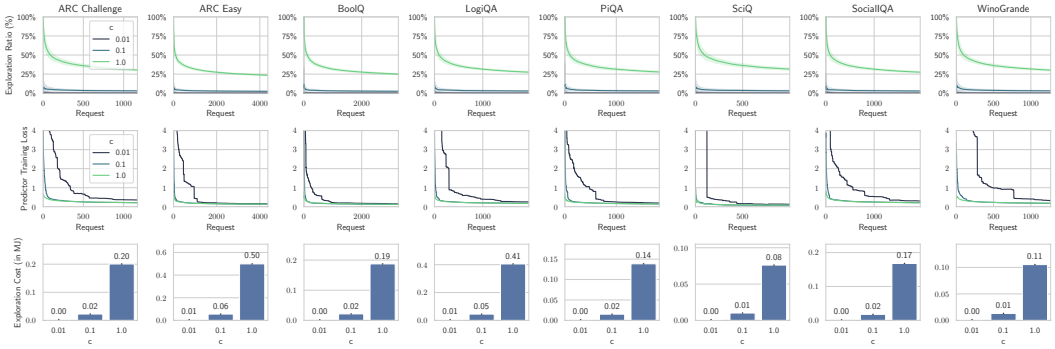


Figure C.1: Full overview of predictor training cost across all benchmarks used in our paper.

intuitive and flexible mechanism for system operators to deliberately navigate performance-cost trade-offs according to application-specific requirements, enabling precise calibration between resource efficiency and quality of service in large-scale LLM deployment environments.

C.3 Predictor Training Evaluation

Our analysis of exploration-exploitation dynamics across eight reasoning benchmarks reveals critical insights for efficient predictive modeling (Figure C.1). We observe that the exploration parameter (c) exhibits predictable effects across benchmarks, with higher values (c = 1.0) maintaining robust exploration but at approximately ten-fold increased energy costs compared to conservative settings (c = 0.01). Notably, task complexity correlates with resource requirements, as evidenced by the significantly higher exploration costs. The predictor training loss patterns indicate that higher exploration parameters facilitate faster convergence and lower overall loss values, suggesting more robust optimization, though with diminishing returns relative to energy expenditure. These findings highlight the importance of context-aware parameter selection in balancing performance gains against computational costs, particularly relevant as AI systems scale and energy efficiency becomes increasingly critical. Generally, we find that choosing $c = 0.1$ provides a strong basis for MESS+ across benchmarks.

Table 8: Additional results for our main results with a smaller value for $V = 0.0001$, which reduces the priority for cost efficiency.

Benchmark	ARC Challenge ($\alpha = 50\%$)			ARC Easy ($\alpha = 75\%$)			BoolQ ($\alpha = 80\%$)		
Method	Operating Cost (in MJ)	Request Satisfaction (in %)	Model Call Ratio (in %) (L70B/L8B/L1B)	Operating Cost (in MJ)	Request Satisfaction (in %)	Model Call Ratio (in %) (L70B/L8B/L1B)	Operating Cost (in MJ)	Request Satisfaction (in %)	Model Call Ratio (in %) (L70B/L8B/L1B)
Llama 1B	$0.09_{\pm 0.00}$	$37.88_{\pm 5.39}$	0% / 0% / 100%	$0.20_{\pm 0.00}$	$62.76_{\pm 5.37}$	0% / 0% / 100%	$0.14_{\pm 0.00}$	$69.17_{\pm 5.13}$	0% / 0% / 100%
Llama 8B	$0.46_{\pm 0.00}$	$54.44_{\pm 5.54}$	0% / 100% / 0%	$0.97_{\pm 0.00}$	$79.72_{\pm 4.47}$	0% / 100% / 0%	$0.43_{\pm 0.00}$	$84.16_{\pm 4.06}$	0% / 100% / 0%
Llama 70B	$2.35_{\pm 0.01}$	$60.84_{\pm 5.43}$	100% / 0% / 0%	$5.05_{\pm 0.01}$	$83.12_{\pm 4.16}$	100% / 0% / 0%	$3.40_{\pm 0.00}$	$88.78_{\pm 3.51}$	100% / 0% / 0%
Educated Guessing	$\mathbf{1.00}_{\pm 0.09}$	$51.65_{\pm 2.98}$	35% / 31% / 34%	$\mathbf{2.00}_{\pm 0.08}$	$74.00_{\pm 4.39}$	31% / 32% / 36%	$\mathbf{1.31}_{\pm 0.04}$	$80.47_{\pm 1.08}$	33% / 34% / 33%
RouteLLM	$1.24_{\pm 0.10}$	$51.17_{\pm 2.93}$	50% / 0% / 50%	$4.05_{\pm 0.01}$	$82.54_{\pm 2.12}$	100% / 0% / 0%	$2.96_{\pm 0.04}$	$86.83_{\pm 1.27}$	87% / 0% / 13%
RouterDC	$2.09_{\pm 0.06}$	$60.94_{\pm 2.92}$	88% / 12% / 0%	$3.61_{\pm 0.06}$	$82.30_{\pm 2.60}$	85% / 15% / 0%	$2.14_{\pm 0.05}$	$87.06_{\pm 2.70}$	58% / 42% / 0%
MESS+ (ours)	$1.51_{\pm 0.09}$	$54.58_{\pm 3.15}$	70% / 9% / 21%	$4.87_{\pm 0.09}$	$78.20_{\pm 1.70}$	54% / 28% / 19%	$1.38_{\pm 0.05}$	$81.12_{\pm 4.99}$	38% / 30% / 32%
Benchmark	LogiQA ($\alpha = 40\%$)			PiQA ($\alpha = 78\%$)			SciQ ($\alpha = 96\%$)		
Method	Operating Cost (in MJ)	Request Satisfaction (in %)	Model Call Ratio (in %) (L70B/L8B/L1B)	Operating Cost (in MJ)	Request Satisfaction (in %)	Model Call Ratio (in %) (L70B/L8B/L1B)	Operating Cost (in MJ)	Request Satisfaction (in %)	Model Call Ratio (in %) (L70B/L8B/L1B)
Llama 1B	$0.17_{\pm 0.00}$	$27.19_{\pm 4.94}$	0% / 0% / 100%	$0.07_{\pm 0.00}$	$74.05_{\pm 4.87}$	0% / 0% / 100%	$0.10_{\pm 0.00}$	$93.80_{\pm 2.68}$	0% / 0% / 100%
Llama 8B	$0.81_{\pm 0.00}$	$29.03_{\pm 5.04}$	0% / 100% / 0%	$0.36_{\pm 0.00}$	$79.33_{\pm 4.50}$	0% / 100% / 0%	$0.44_{\pm 0.00}$	$97.00_{\pm 1.90}$	0% / 100% / 0%
Llama 70B	$4.11_{\pm 0.02}$	$49.31_{\pm 5.56}$	100% / 0% / 0%	$1.84_{\pm 0.01}$	$82.70_{\pm 4.20}$	100% / 0% / 0%	$2.23_{\pm 0.02}$	$97.10_{\pm 1.87}$	100% / 0% / 0%
Educated Guessing	$2.51_{\pm 0.09}$	$39.88_{\pm 4.57}$	56% / 21% / 22%	$\mathbf{0.76}_{\pm 0.04}$	$78.89_{\pm 1.52}$	34% / 32% / 34%	$0.92_{\pm 0.09}$	$96.51_{\pm 1.49}$	31% / 36% / 32%
RouteLLM	$1.33_{\pm 0.04}$	$47.71_{\pm 3.38}$	98% / 0% / 2%	$1.25_{\pm 0.05}$	$78.35_{\pm 1.42}$	66% / 0% / 34%	$2.16_{\pm 0.04}$	$97.76_{\pm 0.73}$	95% / 0% / 5%
RouterDC	$\mathbf{1.09}_{\pm 0.08}$	$47.13_{\pm 3.08}$	70% / 29% / 2%	$1.85_{\pm 0.01}$	$82.34_{\pm 1.33}$	100% / 0% / 0%	$1.90_{\pm 0.07}$	$97.95_{\pm 0.81}$	82% / 18% / 0%
MESS+ (ours)	$2.97_{\pm 0.09}$	$43.89_{\pm 4.52}$	73% / 3% / 24%	$0.84_{\pm 0.04}$	$79.23_{\pm 2.84}$	45% / 35% / 21%	$\mathbf{0.41}_{\pm 0.05}$	$96.12_{\pm 2.33}$	22% / 33% / 45%
Category	SocialIQA ($\alpha = 44\%$)			Winogrande ($\alpha = 70\%$)			*Avg. across all Benchmarks* ($\alpha = 66\%$)		
Subcategory	Operating Cost (in MJ)	Request Satisfaction (in %)	Model Call Ratio (in %) (L70B/L8B/L1B)	Operating Cost (in MJ)	Request Satisfaction (in %)	Model Call Ratio (in %) (L70B/L8B/L1B)	Operating Cost (in MJ)	Request Satisfaction (in %)	Model Call Ratio (in %) (L70B/L8B/L1B)
Llama 1B	$0.13_{\pm 0.00}$	$41.71_{\pm 5.48}$	0% / 0% / 100%	$0.06_{\pm 0.00}$	$59.67_{\pm 5.45}$	0% / 0% / 100%	$0.12_{\pm 0.00}$	$58.28_{\pm 4.92}$	0% / 0% / 100%
Llama 8B	$0.59_{\pm 0.00}$	$48.31_{\pm 5.55}$	0% / 100% / 0%	$0.25_{\pm 0.00}$	$73.64_{\pm 4.90}$	0% / 100% / 0%	$0.54_{\pm 0.00}$	$68.20_{\pm 4.49}$	0% / 100% / 0%
Llama 70B	$3.00_{\pm 0.00}$	$48.67_{\pm 5.56}$	100% / 0% / 0%	$1.29_{\pm 0.00}$	$79.08_{\pm 4.52}$	100% / 0% / 0%	$2.91_{\pm 0.01}$	$73.70_{\pm 4.35}$	100% / 0% / 0%
Educated Guessing	$\mathbf{1.22}_{\pm 0.06}$	$47.71_{\pm 2.50}$	33% / 32% / 35%	$\mathbf{0.54}_{\pm 0.04}$	$70.67_{\pm 3.35}$	35% / 30% / 35%	$\mathbf{1.28}_{\pm 0.07}$	$67.47_{\pm 2.73}$	36% / 31% / 33%
RouteLLM	$2.02_{\pm 0.07}$	$44.32_{\pm 2.40}$	65% / 0% / 35%	$1.27_{\pm 0.02}$	$80.82_{\pm 2.53}$	97% / 0% / 3%	$2.04_{\pm 0.05}$	$71.19_{\pm 2.10}$	82% / 0% / 18%
RouterDC	$2.89_{\pm 0.03}$	$46.76_{\pm 2.62}$	95% / 5% / 0%	$1.30_{\pm 0.00}$	$80.86_{\pm 2.49}$	100% / 0% / 0%	$2.11_{\pm 0.04}$	$73.17_{\pm 2.32}$	85% / 15% / 0%
MESS+ (ours)	$1.46_{\pm 0.07}$	$46.21_{\pm 2.88}$	43% / 24% / 33%	$0.90_{\pm 0.04}$	$74.93_{\pm 2.52}$	73% / 8% / 19%	$1.79_{\pm 0.06}$	$69.16_{\pm 3.12}$	52% / 21% / 27%

C.4 Additional Experiments on Larger Model Zoos

To demonstrate the performance of MESS+ even in larger zoos, we conduct additional experiments on a zoo containing four models: Qwen 2.5 32B (Q32B), Qwen 2 7B (Q7B), Qwen 2 1.5B (Q1.5B), and Qwen 2 0.5B (Q0.5B) with the check-

Table 9: Additional results for our main results with a smaller value for $V = 0.00001$, which reduces the priority for cost efficiency even further.

Category	ARC Challenge ($\alpha = 50\%$)			ARC Easy ($\alpha = 75\%$)			BoolQ ($\alpha = 80\%$)		
Subcategory	Operating Cost (in MJ)	Request Satisfaction (in %)	Model Call Ratio (in %) (L70B/L8B/L1B)	Operating Cost (in MJ)	Request Satisfaction (in %)	Model Call Ratio (in %) (L70B/L8B/L1B)	Operating Cost (in MJ)	Request Satisfaction (in %)	Model Call Ratio (in %) (L70B/L8B/L1B)
Llama 1B	$0.09_{\pm 0.00}$	$37.88_{\pm 5.39}$	0% / 0% / 100%	$0.20_{\pm 0.00}$	$62.76_{\pm 5.37}$	0% / 0% / 100%	$0.14_{\pm 0.00}$	$69.17_{\pm 5.13}$	0% / 0% / 100%
Llama 8B	$0.46_{\pm 0.00}$	$54.44_{\pm 5.54}$	0% / 100% / 0%	$0.97_{\pm 0.00}$	$79.72_{\pm 4.47}$	0% / 100% / 0%	$0.43_{\pm 0.00}$	$84.16_{\pm 4.06}$	0% / 100% / 0%
Llama 70B	$2.35_{\pm 0.01}$	$60.84_{\pm 5.43}$	100% / 0% / 0%	$5.05_{\pm 0.01}$	$83.12_{\pm 4.16}$	100% / 0% / 0%	$3.40_{\pm 0.00}$	$88.78_{\pm 3.51}$	100% / 0% / 0%
Educated Guessing	$\mathbf{1.00}_{\pm 0.09}$	$51.65_{\pm 2.98}$	35% / 31% / 34%	$\mathbf{2.00}_{\pm 0.08}$	$74.00_{\pm 4.39}$	31% / 32% / 36%	$\mathbf{1.31}_{\pm 0.04}$	$80.47_{\pm 1.08}$	33% / 34% / 33%
RouteLLM	$1.24_{\pm 0.10}$	$51.17_{\pm 2.93}$	50% / 0% / 50%	$4.05_{\pm 0.01}$	$82.54_{\pm 2.12}$	100% / 0% / 0%	$2.96_{\pm 0.04}$	$86.83_{\pm 1.27}$	87% / 0% / 13%
RouterDC	$2.09_{\pm 0.06}$	$60.94_{\pm 2.92}$	88% / 12% / 0%	$3.61_{\pm 0.06}$	$82.30_{\pm 2.60}$	85% / 15% / 0%	$2.14_{\pm 0.05}$	$87.06_{\pm 2.70}$	58% / 42% / 0%
MESS+ (ours)	$1.61_{\pm 0.09}$	$54.31_{\pm 2.87}$	68% / 11% / 21%	$5.39_{\pm 0.09}$	$78.28_{\pm 1.63}$	65% / 16% / 19%	$1.38_{\pm 0.05}$	$82.25_{\pm 3.11}$	40% / 33% / 27%

Category	LogiQA ($\alpha = 40\%$)			PiQA ($\alpha = 78\%$)			SciQ ($\alpha = 96\%$)		
Subcategory	Operating Cost (in MJ)	Request Satisfaction (in %)	Model Call Ratio (in %) (L70B/L8B/L1B)	Operating Cost (in MJ)	Request Satisfaction (in %)	Model Call Ratio (in %) (L70B/L8B/L1B)	Operating Cost (in MJ)	Request (in %) Satisfaction	Model Call Ratio (in %) (L70B/L8B/L1B)
Llama 1B	$0.17_{\pm 0.00}$	$27.19_{\pm 4.94}$	0% / 0% / 100%	$0.07_{\pm 0.00}$	$74.05_{\pm 4.87}$	0% / 0% / 100%	$0.10_{\pm 0.00}$	$93.80_{\pm 2.68}$	0% / 0% / 100%
Llama 8B	$0.81_{\pm 0.00}$	$29.03_{\pm 5.04}$	0% / 100% / 0%	$0.36_{\pm 0.00}$	$79.33_{\pm 4.50}$	0% / 100% / 0%	$0.44_{\pm 0.00}$	$97.00_{\pm 1.90}$	0% / 100% / 0%
Llama 70B	$4.11_{\pm 0.02}$	$49.31_{\pm 5.56}$	100% / 0% / 0%	$1.84_{\pm 0.01}$	$82.70_{\pm 4.20}$	100% / 0% / 0%	$2.23_{\pm 0.02}$	$97.10_{\pm 1.87}$	100% / 0% / 0%
Educated Guessing	$2.51_{\pm 0.09}$	$39.88_{\pm 4.57}$	56% / 21% / 22%	$\mathbf{0.76}_{\pm 0.04}$	$78.89_{\pm 1.52}$	34% / 32% / 34%	$0.92_{\pm 0.09}$	$96.51_{\pm 1.49}$	31% / 36% / 32%
RouteLLM	$1.33_{\pm 0.04}$	$47.71_{\pm 3.38}$	98% / 0% / 2%	$1.25_{\pm 0.05}$	$78.35_{\pm 1.42}$	66% / 0% / 34%	$2.16_{\pm 0.04}$	$97.76_{\pm 0.73}$	95% / 0% / 5%
RouterDC	$\mathbf{1.09}_{\pm 0.08}$	$47.13_{\pm 3.08}$	70% / 29% / 2%	$1.85_{\pm 0.01}$	$82.34_{\pm 1.33}$	100% / 0% / 0%	$1.90_{\pm 0.07}$	$97.95_{\pm 0.81}$	82% / 18% / 0%
MESS+ (ours)	$2.86_{\pm 0.09}$	$43.89_{\pm 4.55}$	72% / 2% / 26%	$1.01_{\pm 0.04}$	$79.94_{\pm 2.62}$	51% / 38% / 12%	$\mathbf{0.56}_{\pm 0.07}$	$96.17_{\pm 2.12}$	20% / 37% / 43%

Category	SocialIQA ($\alpha = 44\%$)			Winogrande ($\alpha = 70\%$)			*Avg. across all Benchmarks* ($\alpha = 66\%$)		
Subcategory	Operating Cost (in MJ)	Request Satisfaction (in %)	Model Call Ratio (in %) (L70B/L8B/L1B)	Operating Cost (in MJ)	Request Satisfaction (in %)	Model Call Ratio (in %) (L70B/L8B/L1B)	Operating Cost (in MJ)	Request Satisfaction (in %)	Model Call Ratio (in %) (L70B/L8B/L1B)
Llama 1B	$0.13_{\pm 0.00}$	$41.71_{\pm 5.48}$	0% / 0% / 100%	$0.06_{\pm 0.00}$	$59.67_{\pm 5.45}$	0% / 0% / 100%	$0.12_{\pm 0.00}$	$58.28_{\pm 4.92}$	0% / 0% / 100%
Llama 8B	$0.59_{\pm 0.00}$	$48.31_{\pm 5.55}$	0% / 100% / 0%	$0.25_{\pm 0.00}$	$73.64_{\pm 4.90}$	0% / 100% / 0%	$0.54_{\pm 0.00}$	$68.20_{\pm 4.49}$	0% / 100% / 0%
Llama 70B	$3.00_{\pm 0.00}$	$48.67_{\pm 5.56}$	100% / 0% / 0%	$1.29_{\pm 0.00}$	$79.08_{\pm 4.52}$	100% / 0% / 0%	$2.91_{\pm 0.01}$	$73.70_{\pm 4.35}$	100% / 0% / 0%
Educated Guessing	$\mathbf{1.22}_{\pm 0.06}$	$47.71_{\pm 2.50}$	33% / 32% / 35%	$\mathbf{0.54}_{\pm 0.04}$	$70.67_{\pm 3.35}$	35% / 30% / 35%	$\mathbf{1.28}_{\pm 0.07}$	$67.47_{\pm 2.73}$	36% / 31% / 33%
RouteLLM	$2.02_{\pm 0.07}$	$44.32_{\pm 2.40}$	65% / 0% / 35%	$1.27_{\pm 0.02}$	$80.82_{\pm 2.53}$	97% / 0% / 3%	$2.04_{\pm 0.05}$	$71.19_{\pm 2.10}$	82% / 0% / 18%
RouterDC	$2.89_{\pm 0.03}$	$46.76_{\pm 2.62}$	95% / 5% / 0%	$1.30_{\pm 0.00}$	$80.86_{\pm 2.49}$	100% / 0% / 0%	$2.11_{\pm 0.04}$	$73.17_{\pm 2.32}$	85% / 15% / 0%
MESS+ (ours)	$1.34_{\pm 0.06}$	$46.75_{\pm 2.64}$	45% / 35% / 20%	$0.87_{\pm 0.04}$	$74.69_{\pm 2.55}$	71% / 10% / 18%	$1.88_{\pm 0.07}$	$69.41_{\pm 2.76}$	54% / 23% / 23%

Table 10: Main results with the Qwen 2 model family. Specifically, we use Qwen 2.5 32B, Qwen 2 7B, Qwen 2 1.5B, and Qwen 2 0.5B. MESS+ also outperforms our Educated Guessing baseline in larger model zoos. $V = 0.0001$.

Category	ARC Challenge ($\alpha = 55\%$)			ARC Easy ($\alpha = 77\%$)			BoolQ ($\alpha = 80\%$)		
Subcategory	Operating Cost	Request. Satisfaction	Model Call Ratio (Q32B/Q7B/Q1.5B/Q0.5B)	Operating Cost	Request. Satisfaction	Model Call Ratio (Q32B/Q7B/Q1.5B/Q0.5B)	Operating Cost	Request. Satisfaction	Model Call Ratio (Q32B/Q7B/Q1.5B/Q0.5B)
Qwen2 0.5B	$0.10_{\pm 0.00}$	$30.03_{\pm 45.86}$	0% / 0% / 0% / 100%	$0.21_{\pm 0.00}$	$54.88_{\pm 49.77}$	0% / 0% / 0% / 100%	$0.11_{\pm 0.00}$	$63.09_{\pm 48.26}$	0% / 0% / 0% / 100%
Qwen2 1.5B	$0.14_{\pm 0.00}$	$40.10_{\pm 5.45}$	0% / 0% / 100% / 0%	$0.28_{\pm 0.00}$	$66.62_{\pm 5.24}$	0% / 0% / 100% / 0%	$0.12_{\pm 0.00}$	$76.27_{\pm 4.73}$	0% / 0% / 100% / 0%
Qwen2 7B	$0.31_{\pm 0.00}$	$50.94_{\pm 5.56}$	0% / 100% / 0% / 0%	$0.70_{\pm 0.00}$	$75.42_{\pm 4.78}$	0% / 100% / 0% / 0%	$0.25_{\pm 0.00}$	$84.13_{\pm 4.06}$	0% / 100% / 0% / 0%
Qwen2.5 32B	$1.33_{\pm 0.00}$	$58.28_{\pm 5.48}$	100% / 0% / 0% / 0%	$2.73_{\pm 0.00}$	$78.20_{\pm 4.59}$	100% / 0% / 0% / 0%	$1.63_{\pm 0.00}$	$89.60_{\pm 3.39}$	100% / 0% / 0% / 0%
Educated Guessing	$1.26_{\pm 0.00}$	$56.76_{\pm 3.66}$	82% / 15% / 2% / 1%	$2.22_{\pm 0.00}$	$77.46_{\pm 1.27}$	76% / 20% / 2% / 2%	$0.81_{\pm 0.00}$	$82.40_{\pm 1.15}$	41% / 34% / 14% / 11%
RouteLLM	$1.33_{\pm 0.01}$	$58.16_{\pm 2.56}$	100% / 0% / 0% / 0%	$2.73_{\pm 0.01}$	$77.60_{\pm 2.79}$	99% / 0% / 0% / 1%	$1.43_{\pm 0.00}$	$87.31_{\pm 1.96}$	87% / 0% / 0% / 13%
RouterDC	$1.26_{\pm 0.00}$	$58.65_{\pm 2.63}$	89% / 2% / 9% / 0%	$2.03_{\pm 0.00}$	$77.44_{\pm 4.15}$	70% / 19% / 3% / 8%	$1.45_{\pm 0.00}$	$89.41_{\pm 1.52}$	87% / 8% / 4% / 1%
MESS+ (ours)	$\mathbf{1.18}_{\pm 0.00}$	$56.21_{\pm 4.26}$	83% / 7% / 2% / 8%	$\mathbf{1.69}_{\pm 0.00}$	$77.06_{\pm 2.17}$	53% / 42% / 2% / 3%	$\mathbf{0.78}_{\pm 0.00}$	$82.48_{\pm 3.16}$	40% / 32% / 11% / 17%

Category	LogiQA ($\alpha = 33\%$)			PiQA ($\alpha = 79\%$)			SciQ ($\alpha = 93\%$)		
Subcategory	Operating Cost	Request. Satisfaction	Model Call Ratio (Q32B/Q7B/Q1.5B/Q0.5B)	Operating Cost	Request. Satisfaction	Model Call Ratio (Q32B/Q7B/Q1.5B/Q0.5B)	Operating Cost	Request. Satisfaction	Model Call Ratio (Q32B/Q7B/Q1.5B/Q0.5B)
Qwen2 0.5B	$0.06_{\pm 0.00}$	$25.35_{\pm 43.53}$	0% / 0% / 0% / 100%	$0.09_{\pm 0.00}$	$69.15_{\pm 46.20}$	0% / 0% / 0% / 100%	$0.09_{\pm 0.00}$	$91.20_{\pm 28.34}$	0% / 0% / 0% / 100%
Qwen2 1.5B	$0.08_{\pm 0.00}$	$24.27_{\pm 4.77}$	0% / 0% / 100% / 0%	$0.11_{\pm 0.00}$	$76.06_{\pm 4.74}$	0% / 0% / 100% / 0%	$0.12_{\pm 0.00}$	$94.40_{\pm 2.56}$	0% / 0% / 100% / 0%
Qwen2 7B	$0.06_{\pm 0.00}$	$31.18_{\pm 5.15}$	0% / 100% / 0% / 0%	$0.25_{\pm 0.00}$	$79.49_{\pm 4.49}$	0% / 100% / 0% / 0%	$0.34_{\pm 0.00}$	$95.50_{\pm 2.30}$	0% / 100% / 0% / 0%
Qwen2.5 32B	$0.80_{\pm 0.00}$	$40.86_{\pm 5.47}$	100% / 0% / 0% / 0%	$1.06_{\pm 0.00}$	$80.41_{\pm 4.41}$	100% / 0% / 0% / 0%	$1.19_{\pm 0.00}$	$96.70_{\pm 1.99}$	100% / 0% / 0% / 0%
Educated Guessing	$0.39_{\pm 0.00}$	$33.08_{\pm 3.70}$	50% / 22% / 18% / 10%	$0.71_{\pm 0.00}$	$79.02_{\pm 2.16}$	55% / 31% / 6% / 8%	$0.44_{\pm 0.00}$	$94.35_{\pm 1.15}$	25% / 24% / 25% / 27%
RouteLLM	$0.53_{\pm 0.00}$	$33.62_{\pm 3.75}$	64% / 0% / 0% / 36%	$1.01_{\pm 0.00}$	$79.28_{\pm 1.58}$	95% / 0% / 0% / 5%	$0.68_{\pm 0.00}$	$94.72_{\pm 1.34}$	53% / 0% / 0% / 47%
RouterDC	$0.38_{\pm 0.00}$	$33.16_{\pm 3.20}$	50% / 31% / 12% / 7%	$0.74_{\pm 0.00}$	$79.86_{\pm 1.67}$	63% / 17% / 17% / 3%	$0.46_{\pm 0.00}$	$94.97_{\pm 0.79}$	30% / 16% / 12% / 42%
MESS+ (ours)	$\mathbf{0.34}_{\pm 0.00}$	$33.16_{\pm 5.62}$	49% / 33% / 1% / 17%	$\mathbf{0.66}_{\pm 0.00}$	$79.30_{\pm 2.54}$	56% / 25% / 13% / 6%	$\mathbf{0.21}_{\pm 0.00}$	$93.91_{\pm 1.75}$	18% / 3% / 22% / 57%

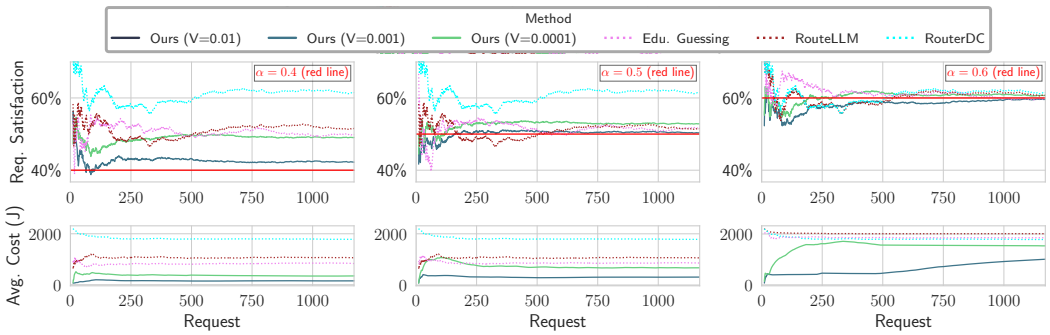
Category	SocialIQA ($\alpha = 47\%$)			Winogrande ($\alpha = 71\%$)			Mean ($\alpha = 0.67$)		
Subcategory	Operating Cost	Request. Satisfaction	Model Call Ratio (Q32B/Q7B/Q1.5B/Q0.5B)	Operating Cost	Request. Satisfaction	Model Call Ratio (Q32B/Q7B/Q1.5B/Q0.5B)	Operating Cost	Request. Satisfaction	Model Call Ratio (Q32B/Q7B/Q1.5B/Q0.5B)
Qwen2 0.5B	$0.26_{\pm 0.00}$	$43.35_{\pm 49.56}$	0% / 0% / 0% / 100%	$0.06_{\pm 0.00}$	$55.88_{\pm 49.67}$	0% / 0% / 0% / 100%	$0.12_{\pm 0.00}$	$54.12_{\pm 45.15}$	0% / 0% / 0% / 100%
Qwen2 1.5B	$0.36_{\pm 0.00}$	$46.37_{\pm 5.54}$	0% / 0% / 100% / 0%	$0.08_{\pm 0.00}$	$64.96_{\pm 5.30}$	0% / 0% / 100% / 0%	$0.16_{\pm 0.00}$	$61.13_{\pm 4.79}$	0% / 0% / 100% / 0%
Qwen2 7B	$1.05_{\pm 0.00}$	$48.21_{\pm 5.55}$	0% / 100% / 0% / 0%	$0.23_{\pm 0.00}$	$71.67_{\pm 5.01}$	0% / 100% / 0% / 0%	$0.40_{\pm 0.00}$	$67.07_{\pm 4.61}$	0% / 100% / 0% / 0%
Qwen2.5 32B	$3.34_{\pm 0.00}$	$50.92_{\pm 5.56}$	100% / 0% / 0% / 0%	$0.73_{\pm 0.00}$	$72.30_{\pm 4.97}$	100% / 0% / 0% / 0%	$1.60_{\pm 0.00}$	$70.91_{\pm 4.48}$	100% / 0% / 0% / 0%
Educated Guessing	$1.47_{\pm 0.00}$	$47.31_{\pm 1.83}$	32% / 29% / 23% / 16%	$0.64_{\pm 0.00}$	$71.59_{\pm 3.67}$	64% / 28% / 3% / 5%	$0.99_{\pm 0.00}$	$67.02_{\pm 2.32}$	53% / 26% / 11% / 10%
RouteLLM	$2.58_{\pm 0.00}$	$47.47_{\pm 1.74}$	65% / 0% / 0% / 35%	$0.71_{\pm 0.00}$	$73.85_{\pm 2.83}$	97% / 0% / 0% / 3%	$1.37_{\pm 0.00}$	$69.01_{\pm 2.31}$	83% / 0% / 0% / 17%
RouterDC	$1.94_{\pm 0.00}$	$48.09_{\pm 2.71}$	47% / 25% / 10% / 18%	$0.57_{\pm 0.00}$	$71.76_{\pm 3.10}$	68% / 20% / 3% / 9%	$1.13_{\pm 0.00}$	$69.17_{\pm 2.47}$	63% / 17% / 9% / 11%
MESS+ (ours)	$\mathbf{1.35}_{\pm 0.00}$	$47.82_{\pm 2.91}$	28% / 30% / 25% / 17%	$\mathbf{0.51}_{\pm 0.00}$	$71.04_{\pm 3.40}$	59% / 33% / 1% / 7%	$\mathbf{0.84}_{\pm 0.00}$	$67.55_{\pm 3.23}$	48% / 26% / 10% / 16%

points, `unsloth/Qwen2.5-32B-Instruct-bnb-4bit`, `unsloth/Qwen2-7B-bnb-4bit`, `Qwen/Qwen2-1.5B-Instruct`, and `Qwen/Qwen2-0.5B-Instruct`, respectively. All checkpoints are readily available on the HuggingFace Hub. We leave all other hyperparameters unchanged and follow the setup we use in our Llama3 model-only zoo. Aside from the Llama 3-only and Qwen 2-only model zoos, we also provide results for a mixed model zoo and an evaluation of varied operating cost characteristics.

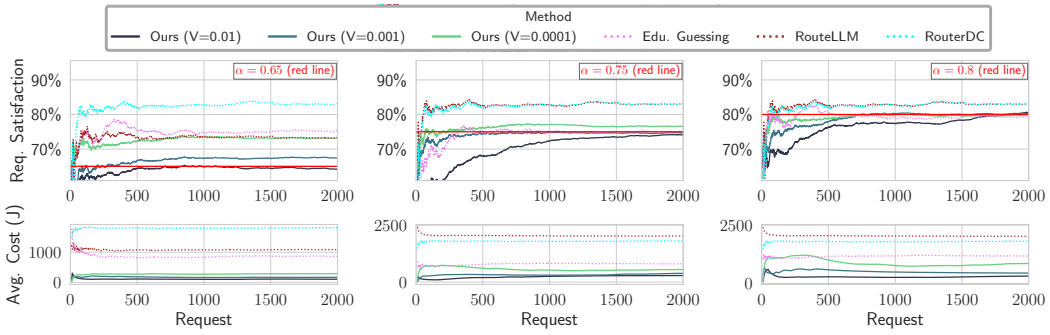
C.4.1 Supplementary Results in Support of the Main Findings with Four Qwen 2 Models

The results on the Qwen 2 only model zoo (Table 5) suggest that zoo expansion can improve cost-performance trade-offs. Increasing the number of models in a zoo has a beneficial effect on overall cost effectiveness, even the cost characteristics of models are more homogeneous than in the Llama3-only model zoo. Notably, the lightweight 0.5B model in the Qwen configuration is

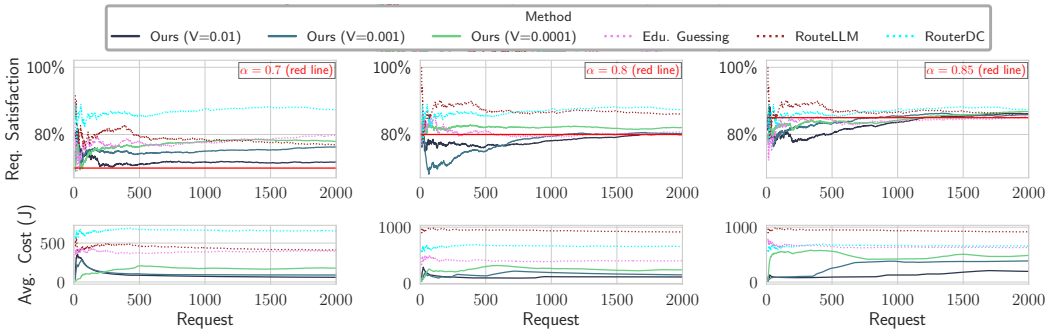
https://doi.org/10.52202/085713-1804



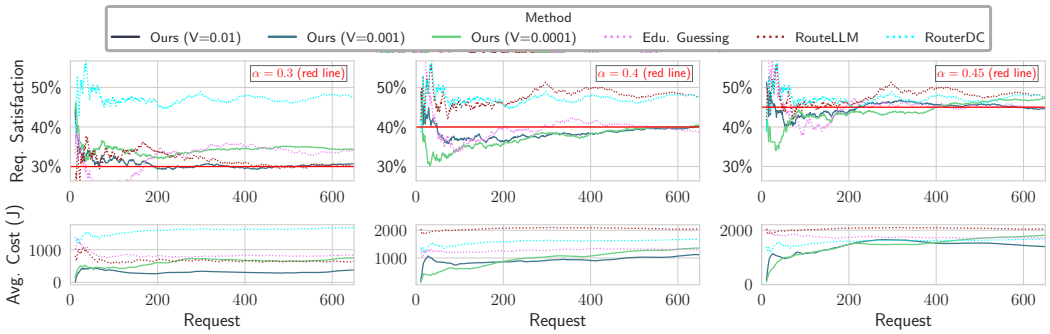
(a) ARC Challenge



(b) ARC Easy



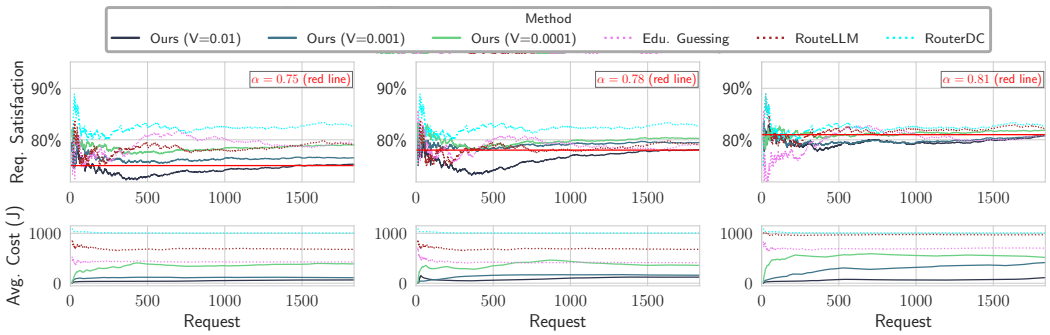
(c) BoolQ



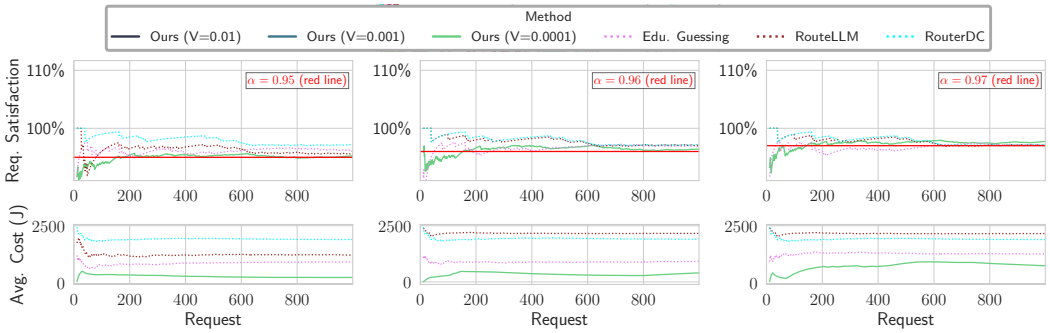
(d) LogiQA

Figure C.2: The dynamics between α and V manifest further across all benchmarks in addition to Winogrande in the main paper. Part 1.

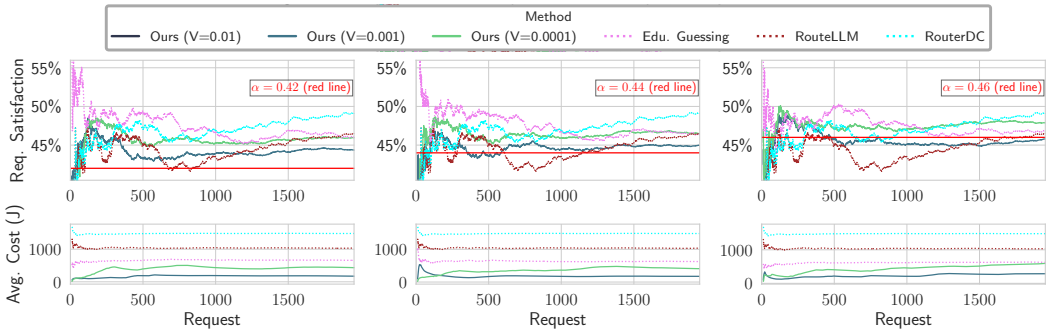
utilized 16% of the time on average, with particularly high usage on simpler tasks like SciQ (57% of calls), demonstrating that there is substantial demand for extremely efficient inference even when larger models are available. This shows that the optimal model zoo size depends on the specific task distribution and performance requirements of the application. Furthermore, the fact that all



(a) PiQA



(b) SciQ



(c) SocialIQA

Figure C.3: The dynamics between α and V manifest further across all benchmarks in addition to Winogrande in the main paper. Part 2.

methods maintain similar relative performance rankings across both configurations indicates that routing algorithm effectiveness may be more important than model zoo size for achieving consistent performance improvements. The scalability benefits appear most pronounced for tasks with high computational variance, where the additional routing granularity can better match query complexity to model capability.

C.4.2 Results on Non-Stationary Concatenated Benchmarks

To evaluate the robustness of adaptive routing methods under more realistic conditions, we also analyze performance on a non-stationary benchmark created by concatenating three distinct datasets: ARC Challenge, PiQA, and Winogrande. This configuration simulates real-world scenarios where query distributions shift dynamically, as the combined benchmark exhibits non-IID characteristics with varying difficulty levels and task types throughout the evaluation sequence (Table 11).

The non-stationary benchmark results demonstrate that MESS+ maintains its cost efficiency advantage even under distributional shifts and under strict SLA compliance, achieving 1.40 MJ operating cost compared to RouterDC's 2.29 MJ and RouteLLM's 2.92 MJ - representing 39% and 52% cost reductions respectively. The non-stationary results are particularly highlight the importance of an

https://doi.org/10.52202/085713-1804

Table 11: Results when concatenating ARC Challenge, PiQA, and Winogrande. Even though the three benchmarks exhibit distinct characteristics, MESS+ shows strong performance compared to our Educated Guessing baseline

Category	Non-stationary Benchmark ($\alpha = 67\%$)		
Subcategory	Operating Cost	Request. Satisfaction	Model Call Ratio (Q32B/Q7B/Q1.5B/Q0.5B)
Qwen2 0.5B	$0.26_{\pm 0.01}$	$54.50_{\pm 49.80}$	0% / 0% / 0% / 100%
Qwen2 1.5B	$0.35_{\pm 0.01}$	$62.92_{\pm 5.37}$	0% / 0% / 100% / 0%
Qwen2 7B	$0.93_{\pm 0.01}$	$69.35_{\pm 5.12}$	0% / 100% / 0% / 0%
Qwen2.5 32B	$3.12_{\pm 0.01}$	$71.94_{\pm 4.99}$	100% / 0% / 0% / 0%
Educated Guessing	$1.64_{\pm 0.01}$	$68.29_{\pm 3.72}$	45% / 41% / 6% / 8%
RouteLLM	$2.92_{\pm 0.01}$	$72.38_{\pm 3.29}$	97% / 0% / 0% / 3%
RouterDC	$2.29_{\pm 0.01}$	$72.36_{\pm 2.46}$	68% / 15% / 4% / 13%
MESS+ (ours)	$\mathbf{1.40}_{\pm 0.01}$	$68.57_{\pm 2.28}$	43% / 41% / 9% / 7%

Table 12: Qwen 2 model zoo with a varied cost spread around the mean cost per request among models in the zoo.

Category	ARC Challenge ($\alpha = 55\%$)		
Subcategory	Operating Cost	Request. Satisfaction	Model Call Ratio (Q32B/Q7B/Q1.5B/Q0.5B)
Qwen2 0.5B	$0.20_{\pm 0.01}$	$30.03_{\pm 45.85}$	0% / 0% / 0% / 100%
Qwen2 1.5B	$0.27_{\pm 0.01}$	$40.10_{\pm 5.45}$	0% / 0% / 100% / 0%
Qwen2 7B	$0.61_{\pm 0.01}$	$50.94_{\pm 5.56}$	0% / 100% / 0% / 0%
Qwen2.5 32B	$2.67_{\pm 0.01}$	$58.28_{\pm 5.48}$	100% / 0% / 0% / 0%
Educated Guessing	$0.58_{\pm 0.01}$	$55.91_{\pm 2.53}$	75% / 20% / 2% / 3%
RouteLLM	$2.67_{\pm 0.01}$	$58.16_{\pm 2.56}$	100% / 0% / 0% / 0%
RouterDC	$0.79_{\pm 0.01}$	$56.92_{\pm 3.07}$	55% / 33% / 12% / 0%
MESS+ (ours)	$\mathbf{0.49}_{\pm 0.01}$	$55.69_{\pm 4.31}$	65% / 21% / 7% / 7%

Category	ARC Easy ($\alpha = 77\%$)		
Subcategory	Operating Cost	Request. Satisfaction	Model Call Ratio (Q32B/Q7B/Q1.5B/Q0.5B)
Qwen2 0.5B	$0.21_{\pm 0.01}$	$54.88_{\pm 49.77}$	0% / 0% / 0% / 100%
Qwen2 1.5B	$0.28_{\pm 0.01}$	$66.62_{\pm 5.24}$	0% / 0% / 100% / 0%
Qwen2 7B	$0.70_{\pm 0.01}$	$75.42_{\pm 4.78}$	0% / 100% / 0% / 0%
Qwen2.5 32B	$2.73_{\pm 0.01}$	$78.20_{\pm 4.59}$	100% / 0% / 0% / 0%
Educated Guessing	$1.22_{\pm 0.01}$	$77.38_{\pm 2.21}$	76% / 20% / 2% / 2%
RouteLLM	$2.73_{\pm 0.01}$	$78.05_{\pm 2.79}$	100% / 0% / 0% / 0%
RouterDC	$1.87_{\pm 0.01}$	$81.43_{\pm 1.93}$	63% / 26% / 11% / 0%
MESS+ (ours)	$\mathbf{0.31}_{\pm 0.01}$	$77.95_{\pm 4.67}$	77% / 10% / 8% / 5%

Category	BoolQ ($\alpha = 80\%$)		
Subcategory	Operating Cost	Request. Satisfaction	Model Call Ratio (Q32B/Q7B/Q1.5B/Q0.5B)
Qwen2 0.5B	$0.11_{\pm 0.01}$	$63.09_{\pm 48.26}$	0% / 0% / 0% / 100%
Qwen2 1.5B	$0.12_{\pm 0.01}$	$76.27_{\pm 4.73}$	0% / 0% / 100% / 0%
Qwen2 7B	$0.25_{\pm 0.01}$	$84.13_{\pm 4.06}$	0% / 100% / 0% / 0%
Qwen2.5 32B	$1.63_{\pm 0.01}$	$89.60_{\pm 3.39}$	100% / 0% / 0% / 0%
Educated Guessing	$0.49_{\pm 0.01}$	$80.62_{\pm 1.36}$	31% / 30% / 20% / 20%
RouteLLM	$1.45_{\pm 0.01}$	$87.31_{\pm 1.96}$	87% / 0% / 0% / 13%
RouterDC	$1.49_{\pm 0.01}$	$88.97_{\pm 1.88}$	89% / 6% / 5% / 0%
MESS+ (ours)	$\mathbf{0.27}_{\pm 0.01}$	$80.02_{\pm 3.03}$	25% / 34% / 19% / 22%

Category	LogiQA ($\alpha = 33\%$)		
Subcategory	Operating Cost	Request. Satisfaction	Model Call Ratio (Q32B/Q7B/Q1.5B/Q0.5B)
Qwen2 0.5B	$0.06_{\pm 0.01}$	$25.35_{\pm 43.53}$	0% / 0% / 0% / 100%
Qwen2 1.5B	$0.08_{\pm 0.01}$	$24.27_{\pm 4.77}$	0% / 0% / 100% / 0%
Qwen2 7B	$0.06_{\pm 0.01}$	$31.18_{\pm 5.15}$	0% / 100% / 0% / 0%
Qwen2.5 32B	$0.80_{\pm 0.01}$	$40.86_{\pm 5.47}$	100% / 0% / 0% / 0%
Educated Guessing	$0.25_{\pm 0.01}$	$33.26_{\pm 3.59}$	42% / 21% / 18% / 19%
RouteLLM	$0.53_{\pm 0.01}$	$33.62_{\pm 3.75}$	64% / 0% / 0% / 36%
RouterDC	$0.35_{\pm 0.01}$	$33.01_{\pm 3.60}$	43% / 23% / 30% / 4%
MESS+ (ours)	$\mathbf{0.29}_{\pm 0.01}$	$33.02_{\pm 5.19}$	43% / 25% / 5% / 28%

Category	PiQA ($\alpha = 79\%$)		
Subcategory	Operating Cost	Request. Satisfaction	Model Call Ratio (Q32B/Q7B/Q1.5B/Q0.5B)
Qwen2 0.5B	$0.17_{\pm 0.01}$	$69.15_{\pm 46.19}$	0% / 0% / 0% / 100%
Qwen2 1.5B	$0.22_{\pm 0.01}$	$76.06_{\pm 4.74}$	0% / 0% / 100% / 0%
Qwen2 7B	$0.50_{\pm 0.01}$	$79.49_{\pm 4.49}$	0% / 100% / 0% / 0%
Qwen2.5 32B	$2.11_{\pm 0.01}$	$80.41_{\pm 4.41}$	100% / 0% / 0% / 0%
Educated Guessing	$0.40_{\pm 0.01}$	$79.05_{\pm 3.08}$	44% / 44% / 6% / 6%
RouteLLM	$2.01_{\pm 0.01}$	$79.28_{\pm 1.58}$	95% / 0% / 0% / 5%
RouterDC	$1.06_{\pm 0.01}$	$79.98_{\pm 1.56}$	49% / 22% / 12% / 17%
MESS+ (ours)	$\mathbf{0.31}_{\pm 0.01}$	$79.08_{\pm 2.61}$	39% / 28% / 20% / 13%

Category	SciQ ($\alpha = 93\%$)		
Subcategory	Operating Cost	Request. Satisfaction	Model Call Ratio (Q32B/Q7B/Q1.5B/Q0.5B)
Qwen2 0.5B	$0.09_{\pm 0.01}$	$91.20_{\pm 28.34}$	0% / 0% / 0% / 100%
Qwen2 1.5B	$0.12_{\pm 0.01}$	$94.40_{\pm 2.56}$	0% / 0% / 100% / 0%
Qwen2 7B	$0.34_{\pm 0.01}$	$95.50_{\pm 2.30}$	0% / 100% / 0% / 0%
Qwen2.5 32B	$1.19_{\pm 0.01}$	$96.70_{\pm 1.99}$	100% / 0% / 0% / 0%
Educated Guessing	$0.38_{\pm 0.01}$	$93.84_{\pm 1.30}$	23% / 23% / 27% / 27%
RouteLLM	$0.64_{\pm 0.01}$	$94.72_{\pm 1.34}$	53% / 0% / 0% / 47%
RouterDC	$0.91_{\pm 0.01}$	$94.75_{\pm 0.93}$	80% / 17% / 1% / 2%
MESS+ (ours)	$\mathbf{0.31}_{\pm 0.01}$	$93.17_{\pm 3.27}$	16% / 8% / 14% / 62%

Category	SocialIQA ($\alpha = 47\%$)		
Subcategory	Operating Cost	Request. Satisfaction	Model Call Ratio (Q32B/Q7B/Q1.5B/Q0.5B)
Qwen2 0.5B	$0.26_{\pm 0.01}$	$43.35_{\pm 49.56}$	0% / 0% / 0% / 100%
Qwen2 1.5B	$0.36_{\pm 0.01}$	$46.37_{\pm 5.54}$	0% / 0% / 100% / 0%
Qwen2 7B	$1.05_{\pm 0.01}$	$48.21_{\pm 5.55}$	0% / 100% / 0% / 0%
Qwen2.5 32B	$3.34_{\pm 0.01}$	$50.92_{\pm 5.56}$	100% / 0% / 0% / 0%
Educated Guessing	$1.16_{\pm 0.01}$	$48.21_{\pm 2.00}$	25% / 26% / 25% / 24%
RouteLLM	$2.21_{\pm 0.01}$	$47.47_{\pm 1.74}$	65% / 0% / 0% / 35%
RouterDC	$1.26_{\pm 0.01}$	$48.70_{\pm 2.73}$	37% / 32% / 19% / 12%
MESS+ (ours)	$\mathbf{1.13}_{\pm 0.01}$	$47.68_{\pm 2.20}$	34% / 19% / 38% / 10%

Category	Winogrande ($\alpha = 71\%$)		
Subcategory	Operating Cost	Request. Satisfaction	Model Call Ratio (Q32B/Q7B/Q1.5B/Q0.5B)
Qwen2 0.5B	$0.12_{\pm 0.01}$	$55.88_{\pm 49.66}$	0% / 0% / 0% / 100%
Qwen2 1.5B	$0.16_{\pm 0.01}$	$64.96_{\pm 5.30}$	0% / 0% / 100% / 0%
Qwen2 7B	$0.47_{\pm 0.01}$	$71.67_{\pm 5.01}$	0% / 100% / 0% / 0%
Qwen2.5 32B	$1.46_{\pm 0.01}$	$72.30_{\pm 4.97}$	100% / 0% / 0% / 0%
Educated Guessing	$0.30_{\pm 0.01}$	$71.02_{\pm 3.61}$	47% / 44% / 4% / 5%
RouteLLM	$1.42_{\pm 0.01}$	$73.85_{\pm 2.83}$	97% / 0% / 0% / 3%
RouterDC	$0.73_{\pm 0.01}$	$74.08_{\pm 2.71}$	61% / 32% / 2% / 5%
MESS+ (ours)	$\mathbf{0.29}_{\pm 0.01}$	$71.05_{\pm 4.18}$	45% / 42% / 10% / 3%

Category	Mean		
Subcategory	Operating Cost	Request. Satisfaction	Model Call Ratio (Q32B/Q7B/Q1.5B/Q0.5B)
Qwen2 0.5B	$0.15_{\pm 0.01}$	$54.12_{\pm 45.15}$	0% / 0% / 0% / 100%
Qwen2 1.5B	$0.20_{\pm 0.01}$	$61.13_{\pm 4.79}$	0% / 0% / 100% / 0%
Qwen2 7B	$0.50_{\pm 0.01}$	$67.07_{\pm 4.61}$	0% / 100% / 0% / 0%
Qwen2.5 32B	$1.99_{\pm 0.01}$	$70.91_{\pm 4.48}$	100% / 0% / 0% / 0%
Educated Guessing	$0.59_{\pm 0.01}$	$67.36_{\pm 2.46}$	45% / 29% / 13% / 13%
RouteLLM	$1.71_{\pm 0.01}$	$69.06_{\pm 2.32}$	83% / 0% / 0% / 17%
RouterDC	$1.05_{\pm 0.01}$	$70.73_{\pm 2.30}$	71% / 17% / 11% / 0%
MESS+ (ours)	$\mathbf{0.33}_{\pm 0.01}$	$66.36_{\pm 3.68}$	33% / 29% / 15% / 24%

adaptive routing approach that learns from online feedback since that enables adaptation to query characteristics without requiring explicit knowledge of task boundaries or distribution shifts. This robustness under non-stationary conditions validates the practical applicability of adaptive routing methods in production environments where query distributions naturally shift over time due to changing user behaviors, seasonal patterns, or evolving application requirements.

C.4.3 Experiments on Narrow Cost Spreads

The narrowed cost ratio configuration presents a more challenging routing scenario by reducing the cost differences between models, which tests the robustness of MESS+ when cost-performance trade-offs become less pronounced (Table 12). In this configuration, the cost spread between the largest and smallest models is compressed from the original wide range to a much narrower band, making routing decisions more nuanced.

Under these constrained conditions, MESS+ demonstrates strong results, achieving an average operating cost of 0.33 MJ - a notable 68% improvement over RouterDC (1.05 MJ) and 81% improvement over RouteLLM (1.71 MJ). The narrowed cost spread reveals interesting behavioral adaptations in routing patterns. MESS+ maintains a balanced distribution (33%/29%/15%/24%) that heavily utilizes the smallest models, with the 0.5B model receiving 24% of calls compared to only 10-16% in previous configurations. This increased reliance on ultra-lightweight models indicates that MESS+ successfully identifies queries that can be handled efficiently even when cost differences are minimal.

Yet, the overall results validate the theoretical expectation that routing becomes more challenging when cost differentials decrease. Interestingly, the Educated Guessing baseline performs comparably

Table 13: Results on the Qwen 2 Model Zoo with sparse Q updates. We randomly sample whether we do a Q update from a uniform distribution with a threshold of 0.2.

Category	ARC Challenge ($\alpha = 55\%$)			ARC Easy ($\alpha = 77\%$)			BoolQ ($\alpha = 80\%$)		
Subcategory	Operating Cost	Request. Satisfaction	Model Call Ratio (Q32B/Q7B/Q1.5B/Q0.5B)	Operating Cost	Request. Satisfaction	Model Call Ratio (Q32B/Q7B/Q1.5B/Q0.5B)	Operating Cost	Request. Satisfaction	Model Call Ratio (Q32B/Q7B/Q1.5B/Q0.5B)
Qwen2 0.5B	$0.10_{\pm 0.01}$	$30.03_{\pm 45.86}$	0% / 0% / 0% / 100%	$0.21_{\pm 0.01}$	$54.88_{\pm 49.77}$	0% / 0% / 0% / 100%	$0.11_{\pm 0.01}$	$63.09_{\pm 48.26}$	0% / 0% / 0% / 100%
Qwen2 1.5B	$0.14_{\pm 0.01}$	$40.10_{\pm 5.45}$	0% / 0% / 100% / 0%	$0.28_{\pm 0.01}$	$66.62_{\pm 5.24}$	0% / 0% / 100% / 0%	$0.12_{\pm 0.01}$	$76.27_{\pm 4.73}$	0% / 0% / 100% / 0%
Qwen2 7B	$0.31_{\pm 0.01}$	$50.94_{\pm 5.56}$	0% / 100% / 0% / 0%	$0.70_{\pm 0.01}$	$75.42_{\pm 4.78}$	0% / 100% / 0% / 0%	$0.25_{\pm 0.01}$	$84.13_{\pm 4.06}$	0% / 100% / 0% / 0%
Qwen2.5 32B	$1.33_{\pm 0.01}$	$58.28_{\pm 5.48}$	100% / 0% / 0% / 0%	$2.73_{\pm 0.01}$	$78.20_{\pm 4.59}$	100% / 0% / 0% / 0%	$1.63_{\pm 0.01}$	$89.60_{\pm 3.39}$	100% / 0% / 0% / 0%
Educated Guessing	$1.26_{\pm 0.01}$	$56.76_{\pm 3.66}$	82% / 15% / 2% / 1%	$2.22_{\pm 0.01}$	$77.46_{\pm 1.27}$	76% / 20% / 2% / 2%	$1.47_{\pm 0.01}$	$82.40_{\pm 1.15}$	87% / 9% / 2% / 2%
RouteLLM	$1.33_{\pm 0.01}$	$58.16_{\pm 2.56}$	100% / 0% / 0% / 0%	$2.73_{\pm 0.01}$	$77.60_{\pm 2.79}$	99% / 0% / 0% / 1%	$1.43_{\pm 0.01}$	$87.31_{\pm 1.96}$	87% / 0% / 0% / 13%
RouterDC	$1.26_{\pm 0.01}$	$58.65_{\pm 2.63}$	89% / 2% / 9% / 0%	$2.03_{\pm 0.01}$	$77.44_{\pm 4.15}$	70% / 19% / 3% / 8%	$1.45_{\pm 0.01}$	$89.41_{\pm 1.52}$	87% / 8% / 4% / 1%
MESS+ (ours)	$\mathbf{1.41}_{\pm 0.01}$	$55.36_{\pm 4.99}$	44% / 28% / 16% / 12%	$\mathbf{1.94}_{\pm 0.01}$	$77.45_{\pm 2.77}$	67% / 10% / 16% / 7%	$\mathbf{1.27}_{\pm 0.01}$	$81.28_{\pm 3.40}$	83% / 12% / 1% / 4%

Category	LogiQA ($\alpha = 33\%$)			PiQA ($\alpha = 79\%$)			SciQ ($\alpha = 93\%$)		
Subcategory	Operating Cost	Request. Satisfaction	Model Call Ratio (Q32B/Q7B/Q1.5B/Q0.5B)	Operating Cost	Request. Satisfaction	Model Call Ratio (Q32B/Q7B/Q1.5B/Q0.5B)	Operating Cost	Request. Satisfaction	Model Call Ratio (Q32B/Q7B/Q1.5B/Q0.5B)
Qwen2 0.5B	$0.06_{\pm 0.01}$	$25.35_{\pm 43.53}$	0% / 0% / 0% / 100%	$0.09_{\pm 0.01}$	$69.15_{\pm 46.20}$	0% / 0% / 0% / 100%	$0.09_{\pm 0.01}$	$91.20_{\pm 28.34}$	0% / 0% / 0% / 100%
Qwen2 1.5B	$0.08_{\pm 0.01}$	$24.27_{\pm 4.77}$	0% / 0% / 100% / 0%	$0.11_{\pm 0.01}$	$76.06_{\pm 4.74}$	0% / 0% / 100% / 0%	$0.12_{\pm 0.01}$	$94.40_{\pm 2.56}$	0% / 0% / 100% / 0%
Qwen2 7B	$0.06_{\pm 0.01}$	$31.18_{\pm 5.15}$	0% / 100% / 0% / 0%	$0.25_{\pm 0.01}$	$79.49_{\pm 4.49}$	0% / 100% / 0% / 0%	$0.34_{\pm 0.01}$	$95.50_{\pm 2.30}$	0% / 100% / 0% / 0%
Qwen2.5 32B	$0.80_{\pm 0.01}$	$40.86_{\pm 5.47}$	100% / 0% / 0% / 0%	$1.06_{\pm 0.01}$	$80.41_{\pm 4.41}$	100% / 0% / 0% / 0%	$1.19_{\pm 0.01}$	$96.70_{\pm 1.99}$	100% / 0% / 0% / 0%
Educated Guessing	$0.39_{\pm 0.01}$	$33.08_{\pm 3.70}$	50% / 22% / 18% / 10%	$0.71_{\pm 0.01}$	$79.02_{\pm 2.16}$	55% / 31% / 6% / 8%	$0.44_{\pm 0.01}$	$94.35_{\pm 1.15}$	25% / 24% / 25% / 27%
RouteLLM	$0.53_{\pm 0.01}$	$33.62_{\pm 3.75}$	64% / 0% / 0% / 36%	$1.01_{\pm 0.01}$	$79.28_{\pm 1.58}$	95% / 0% / 0% / 5%	$0.68_{\pm 0.01}$	$94.72_{\pm 1.34}$	53% / 0% / 0% / 47%
RouterDC	$0.38_{\pm 0.01}$	$33.16_{\pm 3.20}$	50% / 31% / 12% / 7%	$0.74_{\pm 0.01}$	$79.86_{\pm 1.67}$	63% / 17% / 17% / 3%	$0.46_{\pm 0.01}$	$94.97_{\pm 0.79}$	30% / 16% / 12% / 42%
MESS+ (ours)	$\mathbf{0.11}_{\pm 0.01}$	$34.07_{\pm 4.21}$	34% / 31% / 10% / 24%	$\mathbf{0.70}_{\pm 0.01}$	$79.43_{\pm 2.26}$	59% / 19% / 13% / 9%	$\mathbf{0.06}_{\pm 0.01}$	$93.15_{\pm 1.62}$	2% / 4% / 23% / 71%

Category	SocialIQA ($\alpha = 47\%$)			Winogrande ($\alpha = 71\%$)			Mean ($\alpha = 66\%$)		
Subcategory	Operating Cost	Request. Satisfaction	Model Call Ratio (Q32B/Q7B/Q1.5B/Q0.5B)	Operating Cost	Request. Satisfaction	Model Call Ratio (Q32B/Q7B/Q1.5B/Q0.5B)	Operating Cost	Request. Satisfaction	Model Call Ratio (Q32B/Q7B/Q1.5B/Q0.5B)
Qwen2 0.5B	$0.26_{\pm 0.01}$	$43.35_{\pm 49.56}$	0% / 0% / 0% / 100%	$0.06_{\pm 0.01}$	$55.88_{\pm 49.67}$	0% / 0% / 0% / 100%	$0.12_{\pm 0.01}$	$54.12_{\pm 45.15}$	0% / 0% / 0% / 100%
Qwen2 1.5B	$0.36_{\pm 0.01}$	$46.37_{\pm 5.54}$	0% / 0% / 100% / 0%	$0.08_{\pm 0.01}$	$64.96_{\pm 5.30}$	0% / 0% / 100% / 0%	$0.16_{\pm 0.01}$	$61.13_{\pm 4.79}$	0% / 0% / 100% / 0%
Qwen2 7B	$1.05_{\pm 0.01}$	$48.21_{\pm 5.55}$	0% / 100% / 0% / 0%	$0.23_{\pm 0.01}$	$71.67_{\pm 5.01}$	0% / 100% / 0% / 0%	$0.40_{\pm 0.01}$	$67.07_{\pm 4.61}$	0% / 100% / 0% / 0%
Qwen2.5 32B	$3.34_{\pm 0.01}$	$50.92_{\pm 5.56}$	100% / 0% / 0% / 0%	$0.73_{\pm 0.01}$	$72.30_{\pm 4.97}$	100% / 0% / 0% / 0%	$1.60_{\pm 0.01}$	$70.91_{\pm 4.48}$	100% / 0% / 0% / 0%
Educated Guessing	$1.47_{\pm 0.01}$	$47.31_{\pm 1.83}$	32% / 29% / 23% / 16%	$0.64_{\pm 0.01}$	$71.59_{\pm 3.67}$	64% / 28% / 3% / 5%	$0.99_{\pm 0.01}$	$67.02_{\pm 2.32}$	53% / 26% / 11% / 10%
RouteLLM	$2.58_{\pm 0.01}$	$47.47_{\pm 1.74}$	65% / 0% / 0% / 35%	$0.71_{\pm 0.01}$	$73.85_{\pm 2.83}$	97% / 0% / 0% / 3%	$1.37_{\pm 0.01}$	$69.01_{\pm 2.31}$	83% / 0% / 0% / 17%
RouterDC	$1.94_{\pm 0.01}$	$48.09_{\pm 2.71}$	47% / 25% / 10% / 18%	$0.57_{\pm 0.01}$	$71.76_{\pm 3.10}$	68% / 20% / 3% / 9%	$1.13_{\pm 0.01}$	$69.17_{\pm 2.47}$	63% / 17% / 9% / 11%
MESS+ (ours)	$\mathbf{0.54}_{\pm 0.01}$	$47.66_{\pm 2.84}$	45% / 15% / 22% / 18%	$\mathbf{0.53}_{\pm 0.01}$	$71.41_{\pm 4.73}$	67% / 23% / 2% / 8%	$\mathbf{0.82}_{\pm 0.01}$	$67.47_{\pm 3.35}$	49% / 18% / 14% / 19%

to MESS+ in satisfaction (67.36% vs 66.36%) while requiring significantly higher costs (0.59 MJ vs 0.33 MJ), indicating that MESS+ maintains its core advantage of intelligent cost-performance optimization even under adverse conditions.

C.4.4 Experiments on Sparse Q updates

The sparse Q-update configuration, where feedback is provided only 20% of the time (an 80% reduction compared to perfect conditions), tests the robustness of MESS+ under severely limited feedback signals (Table 13).

This scenario simulates realistic deployment conditions where user feedback is scarce or expensive to obtain. Under these constrained learning conditions, MESS+ demonstrates remarkable resilience, maintaining an average operating cost of 0.82 MJ while achieving 67.47% request satisfaction. Compared to the full-feedback Qwen configuration (0.84 MJ, 67.55% satisfaction), the performance degradation is minimal - only 2% cost increase and 0.08 percentage point satisfaction decrease on average. This suggests that MESS+ can operate effectively even with severely limited feedback signals. The sparse feedback results reveal that MESS+ maintains its cost leadership over competing methods, achieving 27% cost savings over RouterDC (1.13 MJ) and 40% over RouteLLM (1.37 MJ). The routing pattern shifts toward increased reliance on smaller models (19% usage of 0.5B model vs 16% in full feedback), indicating that the algorithm becomes more conservative when learning signals are limited, defaulting to cost-efficient choices when confidence is low.

The results demonstrate that adaptive routing methods can maintain practical effectiveness under more realistic feedback constraints, with MESS+ showing particular robustness to sparse learning signals. This finding has important implications for production deployments where continuous user feedback may be limited or costly to collect.

C.4.5 Results with Models from the Llama 3 and Qwen 2 Model Families

The mixed model zoo configuration, combining models from both Llama and Qwen families (Table 14), provides additional insights into the robustness of adaptive routing approaches across heterogeneous model architectures. This configuration demonstrates that MESS+ maintains strong performance even when operating across different LLM families, achieving an average operating cost of 0.98 MJ with 67.47% request satisfaction, which strictly meets our SLA requirement ($\alpha = 0.67$). Comparing the mixed configuration to the homogeneous Qwen setup reveals interesting trade-offs. While the pure Qwen configuration achieves slightly better cost efficiency (0.84 MJ vs 0.98 MJ), the

https://doi.org/10.52202/085713-1804

Table 14: The performance of MESS+ remains strong even when mixing models from different LLM families. Our approach works independently from any model internals since we only require user requests as input and a feedback signal.

Category	ARC Challenge ($\alpha = 55\%$)			ARC Easy ($\alpha = 77\%$)			BoolQ ($\alpha = 80\%$)		
Subcategory	Operating Cost	Request. Satisfaction	Model Call Ratio (Q32B/Q7B/Q1.5B/Q0.5B)	Operating Cost	Request. Satisfaction	Model Call Ratio (Q32B/Q7B/Q1.5B/Q0.5B)	Operating Cost	Request. Satisfaction	Model Call Ratio (Q32B/Q7B/Q1.5B/Q0.5B)
Qwen2 0.5B	$0.20_{\pm 0.01}$	$30.03_{\pm 45.85}$	0% / 0% / 0% / 100%	$0.21_{\pm 0.01}$	$54.88_{\pm 49.77}$	0% / 0% / 0% / 100%	$0.11_{\pm 0.01}$	$63.09_{\pm 48.26}$	0% / 0% / 0% / 100%
Llama 3.2 1B	$0.27_{\pm 0.01}$	$40.10_{\pm 5.45}$	0% / 0% / 100% / 0%	$0.28_{\pm 0.01}$	$66.62_{\pm 5.24}$	0% / 0% / 100% / 0%	$0.12_{\pm 0.01}$	$76.27_{\pm 4.73}$	0% / 0% / 100% / 0%
Llama 3.1 8B	$0.61_{\pm 0.01}$	$50.94_{\pm 5.56}$	0% / 100% / 0% / 0%	$0.70_{\pm 0.01}$	$75.42_{\pm 4.78}$	0% / 100% / 0% / 0%	$0.25_{\pm 0.01}$	$84.13_{\pm 4.06}$	0% / 100% / 0% / 0%
Qwen2.5 32B	$2.67_{\pm 0.01}$	$58.28_{\pm 5.48}$	100% / 0% / 0% / 0%	$2.73_{\pm 0.01}$	$78.20_{\pm 4.59}$	100% / 0% / 0% / 0%	$1.63_{\pm 0.01}$	$89.60_{\pm 3.39}$	100% / 0% / 0% / 0%
Educated Guessing	$1.19_{\pm 0.01}$	$55.87_{\pm 2.62}$	64% / 4% / 28% / 4%	$1.98_{\pm 0.01}$	$77.30_{\pm 2.13}$	46% / 5% / 45% / 4%	$1.33_{\pm 0.01}$	$80.67_{\pm 2.42}$	80% / 16% / 3% / 1%
RouteLLM	$2.67_{\pm 0.01}$	$58.16_{\pm 2.56}$	100% / 0% / 0% / 0%	$2.73_{\pm 0.01}$	$77.06_{\pm 2.79}$	100% / 0% / 0% / 0%	$1.44_{\pm 0.01}$	$87.31_{\pm 1.96}$	87% / 0% / 0% / 13%
RouterDC	$1.64_{\pm 0.01}$	$56.01_{\pm 2.59}$	60% / 19% / 17% / 4%	$1.99_{\pm 0.01}$	$77.87_{\pm 1.30}$	49% / 23% / 19% / 9%	$1.38_{\pm 0.01}$	$89.53_{\pm 1.48}$	85% / 10% / 3% / 2%
MESS+ (ours)	$\mathbf{1.16}_{\pm 0.01}$	$55.44_{\pm 4.52}$	58% / 32% / 3% / 7%	$\mathbf{1.94}_{\pm 0.01}$	$77.24_{\pm 4.02}$	45% / 22% / 26% / 7%	$\mathbf{1.20}_{\pm 0.01}$	$81.75_{\pm 2.11}$	71% / 18% / 9% / 2%

Category	LogiQA ($\alpha = 33\%$)			PiQA ($\alpha = 79\%$)			SciQ ($\alpha = 93\%$)		
Subcategory	Operating Cost	Request. Satisfaction	Model Call Ratio (Q32B/Q7B/Q1.5B/Q0.5B)	Operating Cost	Request. Satisfaction	Model Call Ratio (Q32B/Q7B/Q1.5B/Q0.5B)	Operating Cost	Request. Satisfaction	Model Call Ratio (Q32B/Q7B/Q1.5B/Q0.5B)
Qwen2 0.5B	$0.06_{\pm 0.01}$	$25.35_{\pm 43.53}$	0% / 0% / 0% / 100%	$0.17_{\pm 0.01}$	$69.15_{\pm 46.19}$	0% / 0% / 0% / 100%	$0.09_{\pm 0.01}$	$91.20_{\pm 28.34}$	0% / 0% / 0% / 100%
Llama 3.2 1B	$0.08_{\pm 0.01}$	$24.27_{\pm 4.77}$	0% / 0% / 100% / 0%	$0.22_{\pm 0.01}$	$76.06_{\pm 4.74}$	0% / 0% / 100% / 0%	$0.12_{\pm 0.01}$	$94.40_{\pm 2.56}$	0% / 0% / 100% / 0%
Llama 3.1 8B	$0.16_{\pm 0.01}$	$31.18_{\pm 5.15}$	0% / 100% / 0% / 0%	$0.50_{\pm 0.01}$	$79.49_{\pm 4.49}$	0% / 100% / 0% / 0%	$0.34_{\pm 0.01}$	$95.50_{\pm 2.30}$	0% / 100% / 0% / 0%
Qwen2.5 32B	$0.80_{\pm 0.01}$	$40.86_{\pm 5.47}$	100% / 0% / 0% / 0%	$2.11_{\pm 0.01}$	$80.41_{\pm 4.41}$	100% / 0% / 0% / 0%	$1.19_{\pm 0.01}$	$96.70_{\pm 1.99}$	100% / 0% / 0% / 0%
Educated Guessing	$0.35_{\pm 0.01}$	$35.36_{\pm 2.84}$	42% / 20% / 17% / 21%	$0.85_{\pm 0.01}$	$79.42_{\pm 2.98}$	45% / 4% / 46% / 5%	$0.47_{\pm 0.01}$	$95.76_{\pm 1.08}$	26% / 25% / 27% / 21%
RouteLLM	$0.54_{\pm 0.01}$	$33.62_{\pm 3.75}$	64% / 0% / 0% / 36%	$2.03_{\pm 0.01}$	$79.28_{\pm 1.58}$	95% / 0% / 0% / 5%	$0.66_{\pm 0.01}$	$94.72_{\pm 1.34}$	53% / 0% / 0% / 47%
RouterDC	$\mathbf{0.34}_{\pm 0.01}$	$34.43_{\pm 4.68}$	40% / 18% / 20% / 22%	$0.80_{\pm 0.01}$	$79.05_{\pm 1.39}$	100% / 0% / 0% / 0%	$0.95_{\pm 0.01}$	$95.66_{\pm 0.89}$	88% / 5% / 3% / 4%
MESS+ (ours)	$\mathbf{0.34}_{\pm 0.01}$	$34.26_{\pm 4.06}$	39% / 17% / 21% / 23%	$\mathbf{0.74}_{\pm 0.01}$	$79.49_{\pm 2.81}$	41% / 29% / 22% / 8%	$\mathbf{0.33}_{\pm 0.01}$	$93.45_{\pm 2.06}$	17% / 12% / 36% / 35%

Category	SocialIQA ($\alpha = 47\%$)			Winogrande ($\alpha = 71\%$)			Mean		
Subcategory	Operating Cost	Request. Satisfaction	Model Call Ratio (Q32B/Q7B/Q1.5B/Q0.5B)	Operating Cost	Request. Satisfaction	Model Call Ratio (Q32B/Q7B/Q1.5B/Q0.5B)	Operating Cost	Request. Satisfaction	Model Call Ratio (Q32B/Q7B/Q1.5B/Q0.5B)
Qwen2 0.5B	$0.26_{\pm 0.01}$	$43.35_{\pm 49.56}$	0% / 0% / 0% / 100%	$0.12_{\pm 0.01}$	$55.88_{\pm 49.66}$	0% / 0% / 0% / 100%	$0.15_{\pm 0.01}$	$54.12_{\pm 45.15}$	0% / 0% / 0% / 100%
Llama 3.2 1B	$0.36_{\pm 0.01}$	$46.37_{\pm 5.54}$	0% / 0% / 100% / 0%	$0.16_{\pm 0.01}$	$64.96_{\pm 5.30}$	0% / 0% / 100% / 0%	$0.20_{\pm 0.01}$	$61.13_{\pm 4.79}$	0% / 0% / 100% / 0%
Llama 3.1 8B	$1.05_{\pm 0.01}$	$48.21_{\pm 5.55}$	0% / 100% / 0% / 0%	$0.47_{\pm 0.01}$	$71.67_{\pm 5.01}$	0% / 100% / 0% / 0%	$0.50_{\pm 0.01}$	$67.07_{\pm 4.61}$	0% / 100% / 0% / 0%
Qwen2.5 32B	$3.34_{\pm 0.01}$	$50.92_{\pm 5.56}$	100% / 0% / 0% / 0%	$1.46_{\pm 0.01}$	$72.30_{\pm 4.97}$	100% / 0% / 0% / 0%	$1.99_{\pm 0.01}$	$70.91_{\pm 4.48}$	100% / 0% / 0% / 0%
Educated Guessing	$1.77_{\pm 0.01}$	$47.07_{\pm 2.54}$	51% / 14% / 21% / 14%	$0.54_{\pm 0.01}$	$70.06_{\pm 1.76}$	40% / 19% / 35% / 6%	$1.06_{\pm 0.01}$	$67.69_{\pm 2.30}$	46% / 14% / 29% / 11%
RouteLLM	$2.81_{\pm 0.01}$	$47.47_{\pm 1.74}$	65% / 0% / 0% / 35%	$1.43_{\pm 0.01}$	$73.85_{\pm 2.83}$	97% / 0% / 0% / 3%	$1.79_{\pm 0.01}$	$68.93_{\pm 2.32}$	83% / 0% / 0% / 17%
RouterDC	$1.89_{\pm 0.01}$	$48.11_{\pm 2.78}$	54% / 10% / 16% / 20%	$1.46_{\pm 0.01}$	$72.69_{\pm 2.89}$	100% / 0% / 0% / 0%	$1.31_{\pm 0.01}$	$69.17_{\pm 2.25}$	72% / 11% / 10% / 7%
MESS+ (ours)	$\mathbf{1.64}_{\pm 0.01}$	$47.81_{\pm 2.39}$	47% / 24% / 12% / 16%	$\mathbf{0.52}_{\pm 0.01}$	$70.31_{\pm 4.31}$	39% / 47% / 8% / 6%	$\mathbf{0.98}_{\pm 0.01}$	$67.47_{\pm 3.28}$	42% / 27% / 17% / 14%

mixed setup shows comparable performance satisfaction levels. The mixed configuration exhibits a more balanced model call distribution (42%/27%/17%/14%) compared to the pure Qwen setup (48%/26%/10%/16%), suggesting that the Llama 3.1 8B model provides a valuable intermediate capability tier that complements the Qwen models effectively. Notably, the mixed configuration maintains MESS+'s cost benefits over other adaptive methods, with a 25% cost advantage over RouterDC (0.98 MJ vs 1.31 MJ) and a 45% advantage over RouteLLM (0.98 MJ vs 1.79 MJ). This demonstrates that our routing algorithm's effectiveness is not dependent on model family homogeneity, as the method successfully leverages diverse model capabilities based solely on user request inputs and response feedback signals rather than model internals. The cross-family results also highlight the importance of model selection within heterogeneous zoos. The mixed configuration shows that Llama 3.1 8B receives 27% of routing decisions on average, significantly higher than its Qwen 2 7B counterpart in the pure configuration (26%), suggesting that architectural differences between families can create complementary strengths that adaptive routing can exploit.