ReLope: KL-Regularized LoRA Probes for Multimodal LLM Routing

Anonymous ACL submission

Abstract

Routing has emerged as a promising strategy for balancing performance and cost in large language model (LLM) systems that combine lightweight models with powerful but expensive large models. Recent studies show that *probe routing*, which predicts the correctness of a small model using its hidden states, provides an effective solution in text-only LLMs. However, we observe that these probes degrade substantially when applied to multimodal LLMs (MLLMs). Through empirical analysis, we find that the presence of visual inputs weakens the separability of correctness signals in hidden states, making them harder to extract using standard probe designs. To address this challenge, we introduce two complementary approaches for improving probe routing in MLLMs. First, we propose the *Attention Probe*, which aggregates hidden states from the preceding layer based on attention scores to recover distributed correctness signals. Second, we present the *KL-Regularized LoRA Probe (ReLope)*, which inserts a lightweight LoRA adapter and applies a KL regularizer to learn routing-aware representations. Comprehensive experiments demonstrate that our methods consistently outperform strong routing baselines, suggesting that improving the quality of hidden states is key to effective routing in MLLMs.

1 Introduction

Large Language Models (LLMs) have rapidly advanced and are now widely deployed in real-world applications (Zhao et al., 2023). However, their deployment faces a fundamental trade-off: large models provide strong performance but introduce substantial costs, while smaller models are cheaper but less capable. This tension has motivated the development of *routing strategies*, which dynamically decide whether a query can be handled by a lightweight model or should be escalated to a stronger one (Chen et al., 2023, 2025; Jitkrittum et al., 2025). Such strategies enable hybrid LLM systems that achieve favorable trade-offs between performance and efficiency.

Most existing routing methods have been studied in the context of text-only LLMs. Recently, however, multimodal LLMs (MLLMs) (Yin et al., 2024; Liang et al., 2024) have emerged as a new foundation for AI systems capable of jointly processing text and visual inputs. While these models enable more complex reasoning tasks, they also introduce new challenges for efficient deployment. In particular, visual inputs significantly increase inference cost, creating an even stronger need for effective routing mechanisms in multimodal systems.

Among existing strategies, the trained probe has emerged as one of the most effective approaches (Gupta et al., 2024; Chuang et al., 2025). It trains lightweight classifiers on hidden states of the small model to estimate whether the generated answer is correct. Compared with heuristic confidence measures such as token probabilities and average entropy (Huang et al., 2023; Mahaut et al., 2024), probes capture richer semantic signals from LLMs' hidden states, leading to a strong ability in correctness estimation. Because probes operate directly on internal hidden states, they are also architecture-agnostic and, in principle, applicable across different model modalities. However, it remains unclear whether hidden state probes remain reliable when visual inputs are introduced. This naturally raises a research question: *Can probe routings designed for text-only LLMs transfer effectively to MLLMs, and if not, how should they be adapted?*

Through extensive experiments, we identify a consistent failure pattern: probes that are effective for text-only routing become substantially less reliable once visual inputs are introduced. Across ScienceQA, A-OKVQA, and MMMU (Saikh et al., 2022; Schwenk et al., 2022; Yue et al., 2024), their routing AUC drops in the multimodal setting, even when the images provide information necessary for

answering the questions. This observation raises a deeper issue than a simple loss of probe accuracy: visual evidence can benefit answer generation while making correctness cues harder to read from the last-token hidden state. Motivated by (Guo et al., 2023), we conduct a matched representation analysis using HSIC and CKA, which measure dependence between hidden states and correctness without training a router. The analysis consistently indicates weaker correctness dependence for raw multimodal inputs than for caption-based counterparts. Together, these findings suggest that multimodal processing disperses or obscures routing-relevant information, motivating routers that actively recover and reshape such signals rather than directly probing fixed hidden states.

To address this challenge, we propose two complementary techniques for improving probe routing in MLLMs. First, we introduce the *Attention Probe*, which aggregates token-level hidden states from the preceding layer using an attention mechanism. It recovers distributed correctness signals that may be diluted in the final token hidden state. Second, we propose the *KL-Regularized LoRA Probe (ReLope)*, which adapts hidden representations through a lightweight Low-Rank Adaptation (LoRA) module (Hu et al., 2022) and regularizes them using a KL divergence inspired by the Variational Information Bottleneck (VIB) theory (Alemi et al., 2016). This regularization encourages the model to learn compressed representations that retain information relevant for routing decisions while suppressing task-irrelevant visual noise. Comprehensive experiments show that both Attention Probe and ReLope consistently improve routing performance over existing baselines, with ReLope achieving the strongest accuracy–escalation-rate curves and negligible measured routing latency overhead in our hybrid MLLM setting.

Our contributions are summarized as follows:

- We identify and empirically uncover a previously underexplored limitation of probe routing: its performance degrades in MLLMs due to weaker correctness signals in hidden states.
- We propose two effective routing techniques in MLLMs based on the trained probe: the *Attention Probe*, which aggregates token-level hidden states to recover distributed routing signals, and *ReLope*, which learns routing-aware representations via LoRA adaptation and KL regularization.
- Extensive experiments on five multimodal benchmarks demonstrate that our methods consistently outperform existing routing approaches, significantly improving routing quality while maintaining efficient inference in hybrid MLLM systems.

2 Related Work

2.1 Routing Before the Small Model

One family of methods determines routing decisions before querying the small model, relying only on the input queries (Shah and Shridhar, 2025; Ong et al., 2024; Panda et al., 2025; Zhang et al., 2025). Early works train a pretrained router to classify which model will succeed for a given query. Examples include classifier-based routers such as (Shnitzer et al., 2023) and SelectLLM (Ong et al., 2024), reward-based routers like (Lu et al., 2023), and cost-aware strategies like Routoo (Mohammadshahi et al., 2024). Others use non-pretrained strategies, such as Eagle (Zhao et al., 2024), which leverage heuristic selection without extra training.

These methods avoid invoking the small model when unnecessary, reducing cost. However, they are highly dependent on the quality of collected data for training or comparison. More importantly, these methods focus on the text-only prompts. When extending to MLLMs, their router architectures should be redesigned to incorporate additional modalities such as images or videos, which substantially increases implementation complexity.

2.2 Routing After the Small Model

A second class of methods leverages information from the small model. This includes both its final outputs (cascade) and hidden states probe. Cascade approaches run the small model and then apply a rule to decide whether escalation to the large model is needed according to the output of the small model. FrugalGPT (Chen et al., 2023) highlights that cascades can substantially reduce cost while maintaining quality. Most subsequent strategies use different rules to evaluate the small model's confidence as a routing score (Zhao et al., 2024; Chuang et al., 2025; Yue et al., 2023). For example, MoT (Yue et al., 2023) applies multiple inferences and refers to the answer consistency to route. The work (Chuang et al., 2025) evaluates the performance of existing uncertainty estimation methods in the routing problem. Another work (Gupta et al., 2024) shows that sequence-level un-

certainty is biased (favoring short or long generations), and proposes refined rules using token-level uncertainty quantiles.

Another line of work directly trains a probe on hidden states of the small model to predict whether its answer will be correct. Post-hoc embedding probe methods (Chuang et al., 2025; Gupta et al., 2024; Mahaut et al., 2024) show that probes trained on intermediate layers can effectively capture correctness signals. These methods have achieved strong performance in text-only routing, balancing efficiency and reliability. However, we empirically observe that the introduction of images leads to degraded routing accuracy, motivating us to design new routing strategies. VILA (Lin et al., 2024) studies how visual-language pre-training choices affect downstream VLM capability, including preservation of text-only ability. In contrast, our question is whether a fixed small MLLM's hidden states reveal its per-example correctness sufficiently well for routing; our metric, supervision, and intervention are therefore routing-specific.

3 Preliminaries

3.1 Problem Definition

We consider a hybrid MLLM system composed of a small model, such as an on-device model, and a large model that is more capable but also more expensive, typically deployed in the cloud. In this setting, a router is used to determine whether the small model is sufficient for a given query $\mathbf{x}$ or whether the query should be escalated to the large model. Routing can therefore be naturally formulated as correctness prediction for the small model: if the small model is expected to answer correctly, the system uses its output; otherwise, the query is deferred to the large model.

Formally, let $h_s(\mathbf{x})$ denote the prediction of the small model and $h_l(\mathbf{x})$ denote the prediction of the large one. The router $r(\mathbf{x})$ makes a binary decision:

$$r(\mathbf{x}) = \begin{cases} 0, & \text{defer to } h_l(\mathbf{x}), \\ 1, & \text{use } h_s(\mathbf{x}). \end{cases}$$

Let $Y \in \{0, 1\}$ denote whether the output of the small model is correct with respect to the ground truth, where $Y = 1$ indicates correctness. The goal is to design $r(\mathbf{x})$ to accurately predict Y, thereby using the small model when it is likely to be correct and deferring to the large model otherwise.

3.2 Probes Trained on Hidden States

A representative model-aware routing approach uses hidden states of the small model to predict whether its output $h_s(\mathbf{x})$ will be correct. Previous works have demonstrated that hidden states provide a strong signal of the confidence, thereby achieving strong performance in LLM routing (Chuang et al., 2025; Gupta et al., 2024; Mahaut et al., 2024).

Formally, let θ denote the parameters of the LLM. Given an input query $\mathbf{x} = (x_1, \ldots, x_n)$ with sequence length n, the model produces a sequence of hidden states at layer l:

$$\mathbf{Z}^{(l)} = \{\mathbf{z}_1^{(l)}, \mathbf{z}_2^{(l)}, \ldots, \mathbf{z}_n^{(l)}\} \in \mathbb{R}^{n \times d},$$

where d denotes the hidden dimension. Following previous work (Chuang et al., 2025; Mahaut et al., 2024), the last-token hidden state $\mathbf{z}_n^{(l)}$ is used as the input of the probe.

A probe logit function $g_\phi : \mathbb{R}^d \to \mathbb{R}$, parameterized by ϕ, is trained to predict the correctness label Y. The predicted probability is defined as

$$\hat{y} = \text{sigmoid}\left(g_\phi(\mathbf{z}_n^{(l)})\right),$$

where $\hat{y}$ denotes the predicted probability that the small-model answer is correct. The probe learns a mapping from intermediate hidden states to the correctness label Y and therefore serves as a routing estimator. Compared with sequence-level signals such as input difficulty or output consistency, probes can capture richer semantic information encoded in hidden states (Chuang et al., 2025), which leads to effective routing performance.

AUC	MMMU	AOKVQA	ScienceQA
Text-only	**81.51**	**87.50**	**95.36**
Multimodal	76.51	82.03	83.95

Table 1: Routing AUC of probes on multimodal datasets. Probes learned on text-only samples consistently outperform probes learned on multimodal inputs.

3.3 Mechanism Analysis: Visual Inputs Dilute Correctness Cues

Although hidden-state probes are effective in text-only LLM routing, their performance degrades substantially in MLLMs. We first establish this phenomenon across datasets and then conduct a matched intervention to test whether it reflects weaker correctness information in the representation consumed by the router.

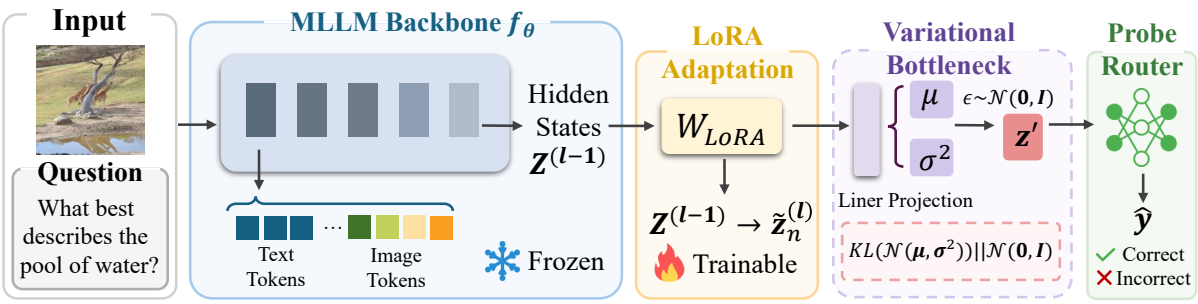


Figure 1: **ReLope pipeline.** Given an input sample $(\mathbf{x}, y)$, the MLLM f_θ produces hidden states $\mathbf{Z}^{(l-1)}$ at layer $l-1$. A lightweight LoRA adapter θ_{LoRA} produces adapted hidden states $\tilde{\mathbf{Z}}^{(l)}$. Its adapted last-token feature $\mathbf{z}_{\text{L}} = \tilde{\mathbf{z}}_n^{(l)}$ is mapped by two heads to $(\boldsymbol{\mu}, \log \boldsymbol{\sigma}^2)$, defining a bottleneck distribution. In both training and inference, the probe g_ϕ predicts correctness from the bottleneck feature $\mathbf{z}_{\text{B}}$: it is sampled through reparameterization in training and set to $\boldsymbol{\mu}$ in inference. The objective combines binary cross-entropy with a KL regularizer.

First, we use ScienceQA (Saikh et al., 2022), which naturally contains two subsets: a text-only subset and a subset with images. We train and evaluate probes separately on these two subsets. The results in Table 1 show that probe performance, measured by AUC, is consistently lower when images are present. Second, to reduce potential bias caused by differences in the underlying samples, we conduct an additional comparison on A-OKVQA (Schwenk et al., 2022) and MMMU (Yue et al., 2024), both of which originally contain only image-text paired samples. For each dataset, we construct a text-only version by prompting GPT to generate descriptions of the images. Probes are then trained and evaluated on both the original multimodal version and the caption-based text-only version. The results again show that probes perform better on text-only versions, indicating that the presence of images weakens correctness signals in hidden states, leading to lower routing accuracy.

A matched representation intervention. The above AUC comparison could still reflect probe optimization rather than a representation-level failure. We therefore construct matched input conditions on A-OKVQA, MMMU, and the image-containing subset of ScienceQA, intervening only on the visual input: (i) the original image-text input ($\mathbf{x}_{\text{mm}}$), (ii) a caption replacing the image ($\mathbf{x}_{\text{cap}}$), which expresses visual semantics in linguistic form, and (iii) the same prompt with a blank image ($\mathbf{x}_{\text{blank}}$), which retains multimodal formatting while removing useful visual evidence. We measure HSIC and normalized HSIC (CKA) between the last-token state and correctness label. Since these measures require no trained correctness classifier, they directly test whether routing-relevant signal is present in the representation (Guo et al., 2023).

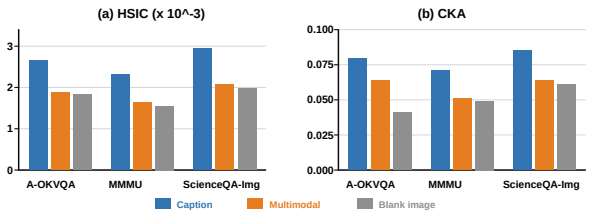


Figure 2: Representation–correctness dependence under visual input interventions. Caption-based inputs consistently preserve stronger correctness cues than raw multimodal or blank image inputs across benchmarks, indicating that visual token processing can dilute routing-relevant information in the final token state.

Figure 2 shows a consistent pattern across datasets: caption-based inputs expose the strongest dependence between the representation and answer correctness, while raw multimodal inputs weaken this signal even though they contain informative visual evidence. Blank-image inputs further confirm that useful visual content remains necessary. These results motivate our design: Attention Probe recovers routing cues dispersed across tokens, while ReLope adapts and compresses the representation so that correctness-relevant information is easier for the router to isolate. The component controls in Table 4 test these two roles directly.

4 Method

Figure 1 illustrates the ReLope routing pipeline, and Algorithm 1 summarizes the training procedures for both routing probes. The colored blocks denote separate training runs: Attention Probe learns an aggregation over frozen hidden states, while ReLope learns a LoRA-adapted bottleneck representation for routing.

4.1 Aggregation Hidden States for Probing

The empirical results in Sec. 3.3 show that the last token hidden state $\mathbf{z}_n^{(l)}$, which is the standard input to probes, provides a weak routing signal in MLLMs. One possible reason is that information relevant to answer correctness is not concentrated in the final token alone. Instead, such information can be distributed across multiple token representations, especially when textual and visual information are jointly encoded.

To address this issue, we construct the probe feature from token-level hidden states in the preceding layer. Let $\mathbf{Z}^{(l-1)} = \{\mathbf{z}_i^{(l-1)}\}_{i=1}^n$ denote the hidden states at layer $l-1$. In a transformer, the last token hidden state at layer l is produced by aggregating information from all tokens' hidden states in layer $l-1$ through the attention module. However, this aggregation is optimized for answer generation by the MLLM, rather than for routing. As a result, the resulting representation may emphasize information useful for producing the next token while weakening signals that are more informative for predicting answer correctness. In MLLMs, this issue can be more severe because irrelevant or redundant visual content may also be mixed into the final-token representation. We therefore propose to replace this implicit aggregation with a learned aggregation tailored to routing.

Attention Probe. We define the probe input as

$$\mathbf{z}_{\mathrm{A}} = \sum_{i=1}^{n} \omega_i \mathbf{z}_i^{(l-1)},$$

where ω_i denotes the contribution weight of token i. The weights are computed with a lightweight attention mechanism:

$$s_i = \frac{\mathbf{q}^\top \mathbf{z}_i^{(l-1)}}{\sqrt{d}}, \quad \omega_i = \frac{\exp(s_i)}{\sum_{j=1}^{n} \exp(s_j)},$$

where $\mathbf{q} \in \mathbb{R}^d$ is a learnable routing query vector and d is the hidden dimension. The aggregated representation $\mathbf{z}_{\mathrm{A}}$ is used as the input to Attention Probe. The single query vector $\mathbf{q}$ is shared across samples and jointly optimized with the probe using the BCE routing objective; the MLLM backbone remains frozen for Attention Probe. This design does not modify the backbone MLLM and only changes how the probe reads intermediate representations. In this way, the probe can focus on routing-relevant cues while reducing the influence of irrelevant visual information that may be carried into the standard last-token state $\mathbf{z}_n^{(l)}$.

The results in Table 2 show that the Attention Probe consistently improves routing AUC over the standard last token probe across most datasets and backbone models. These results suggest that correctness-related signals in MLLMs are not well captured by the final-token representation alone, and that a learned token-level aggregation provides a stronger feature for routing.

Algorithm 1: Attention Probe and ReLope

Input: Training set $\mathcal{D}$, small MLLM f_θ, probe g_ϕ, layer l, epochs T, KL weight $\beta(t)$
Output: Attention parameters $(\mathbf{q}, \phi)$; ReLope parameters $(\theta_{\mathrm{LoRA}}, \psi, \phi)$

```
// The two colored blocks specify separate training runs.
```

for $t = 1$ **to** T **do**

```
// For each (x, y) ∈ D, optimize the selected block.
// Attention Probe (Sec. 4.1)
```

$\mathbf{Z}^{(l-1)} \leftarrow f_\theta(\mathbf{x})$;
$\omega_i \leftarrow \mathrm{softmax}_i(\mathbf{q}^\top \mathbf{z}_i^{(l-1)}/\sqrt{d})$;
$\mathbf{z}_{\mathrm{A}} \leftarrow \sum_i \omega_i \mathbf{z}_i^{(l-1)}$;
$\hat{y}^{\mathrm{A}} \leftarrow \mathrm{sigmoid}(g_\phi(\mathbf{z}_{\mathrm{A}}))$;
Update $(\mathbf{q}, \phi)$ using $\mathrm{BCE}(y, \hat{y}^{\mathrm{A}})$;

```
// ReLope (Sec. 4.2)
```

$\tilde{\mathbf{Z}}^{(l)} \leftarrow f_{\theta,\theta_{\mathrm{LoRA}}}(\mathbf{x})$;
$\mathbf{z}_{\mathrm{L}} \leftarrow \tilde{\mathbf{z}}_n^{(l)}$;
$(\boldsymbol{\mu}, \log \boldsymbol{\sigma}^2) \leftarrow \psi(\mathbf{z}_{\mathrm{L}})$;
$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$;
$\mathbf{z}_{\mathrm{B}} \leftarrow \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$;
$\hat{y}^{\mathrm{R}} \leftarrow \mathrm{sigmoid}(g_\phi(\mathbf{z}_{\mathrm{B}}))$;
$\mathcal{L}_{\mathrm{R}} \leftarrow \mathrm{BCE}(y, \hat{y}^{\mathrm{R}}) + \beta D_{\mathrm{KL}}(q_\psi(\mathbf{z}_{\mathrm{B}} \mid \mathbf{z}_{\mathrm{L}}) \,\|\, \mathcal{N}(\mathbf{0}, \mathbf{I}))$;
Update $(\theta_{\mathrm{LoRA}}, \psi, \phi)$ using $\mathcal{L}_{\mathrm{R}}$;

```
// ReLope inference : z_B ← μ and ŷ^R ← sigmoid(g_φ(z_B)).
```

4.2 ReLope

The Attention Probe reweights existing features but does not alter the representation space itself. We next consider a more expressive approach that directly adapts the hidden state for routing. To this end, we introduce **ReLope** (KL-Regularized LoRA Probe), which combines LoRA adaptation with a variational bottleneck objective. The LoRA adapters provide an efficient way to reshape the hidden representation without full fine-tuning, while the bottleneck encourages the adapted feature to retain information related to answer correctness and discard irrelevant variation.

Model Structure. As depicted in Figure 1, ReLope first adapts a routing-relevant hidden representation with LoRA, maps it through a stochastic bottleneck, and then predicts correctness from the bottleneck feature. We insert LoRA adapters into the selected transformer layer l and keep the original backbone parameters frozen. The LoRA adapters take the hidden states $\mathbf{Z}^{(l-1)} \in \mathbb{R}^{n \times d}$ at transformer layer $l-1$ to produce adapted hidden states $\tilde{\mathbf{Z}}^{(l)}$, from which we obtain the adapted last-token feature

$$\mathbf{z}_{\mathrm{L}} = \tilde{\mathbf{z}}_n^{(l)}$$

for LoRA. Instead of sending $\mathbf{z}_{\mathrm{L}}$ directly to the probe, ReLope maps it to a stochastic bottleneck. Two linear projection heads, jointly denoted by ψ, produce the parameters of a diagonal Gaussian:

$$\boldsymbol{\mu} = \mathbf{W}_{\mu}\mathbf{z}_{\mathrm{L}} + \mathbf{b}_{\mu}, \quad \log \boldsymbol{\sigma}^2 = \mathbf{W}_{\log \sigma^2}\mathbf{z}_{\mathrm{L}} + \mathbf{b}_{\log \sigma^2}.$$

These parameters define a variational posterior

$$q_{\psi}(\mathbf{z}_{\mathrm{B}} \mid \mathbf{z}_{\mathrm{L}}) = \mathcal{N}(\boldsymbol{\mu}, \mathrm{diag}(\boldsymbol{\sigma}^2)),$$

where $\mathbf{z}_{\mathrm{B}}$ denotes the bottleneck feature supplied to the ReLope probe. During training, it is sampled using the reparameterization trick:

$$\mathbf{z}_{\mathrm{B}} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

The probe prediction is always computed from this bottleneck feature:

$$\hat{y}(\mathbf{z}_{\mathrm{B}}) = \mathrm{sigmoid}(g_{\phi}(\mathbf{z}_{\mathrm{B}})).$$

Specifically, training uses a reparameterized sample $\mathbf{z}_{\mathrm{B}} \sim q_{\psi}(\cdot \mid \mathbf{z}_{\mathrm{L}})$, whereas inference deterministically sets $\mathbf{z}_{\mathrm{B}} = \boldsymbol{\mu}$. Thus the probe receives the same feature type in both phases and never directly classifies $\mathbf{z}_{\mathrm{L}}$.

Training Objective. ReLope jointly optimizes probe prediction and representation compression:

$$\mathcal{L} = \mathbb{E}_{(\mathbf{x},y)}\Big[\, \mathbb{E}_{\mathbf{z}_{\mathrm{B}} \sim q_{\psi}(\cdot \mid \mathbf{z}_{\mathrm{L}})}\big[\mathrm{BCE}(y, \hat{y}(\mathbf{z}_{\mathrm{B}}))\big] + \beta D_{\mathrm{KL}}(q_{\psi}(\mathbf{z}_{\mathrm{B}} \mid \mathbf{z}_{\mathrm{L}}) \,\|\, \mathcal{N}(\mathbf{0}, \mathbf{I}))\Big]. \tag{1}$$

The first term is the binary cross-entropy loss evaluated on the sampled bottleneck feature $\mathbf{z}_{\mathrm{B}}$, not on the intermediate LoRA feature $\mathbf{z}_{\mathrm{L}}$. The second term regularizes the bottleneck distribution by encouraging its posterior to stay close to a standard Gaussian prior, with β denoting the KL regularization weight. Following the information bottleneck view (Tishby and Zaslavsky, 2015; Alemi et al., 2016), this regularization encourages the representation to preserve information that is useful for predicting correctness while discarding irrelevant details in the hidden states.

Unlike existing routing methods, which make routing decisions from fixed model inputs or outputs, ReLope learns routing-oriented features by adapting the hidden states themselves. This design allows ReLope to make decisions based on features that are more informative for the answer's correctness in MLLMs and more robust to irrelevant multimodal variation.

5 Experiment

5.1 Datasets

We evaluate ReLope routing performance on five widely used multimodal benchmarks. **MMMU** (Yue et al., 2024) is designed to evaluate advanced multimodal reasoning across multiple academic disciplines and contains thousands of expert-level questions spanning more than 30 subjects. **AOKVQA** (Schwenk et al., 2022) contains over 25,000 questions and requires both external world knowledge and image understanding ability to answer correctly. **ScienceQA** (Saikh et al., 2022) is a science question answering benchmark consisting of over 21,000 middle school science problems. Here, we only use its multimodal subset. **ChartQA** (Masry et al., 2022) is a benchmark for question answering over charts and tables, requiring models to interpret structured visual data and perform logical reasoning with it. **MathVision** (Wang et al., 2024) contains visually grounded mathematics problems, requiring models to interpret diagrams and symbolic expressions in order to solve complex mathematical questions.

5.2 Baselines

We compare ReLope with several representative routing strategies. **MoT** (Yue et al., 2023) performs routing based on multiple reasoning traces generated by the model. The final routing decision is obtained by aggregating signals from different reasoning paths, which provides a more stable estimate of task difficulty. **Cascade Routing** (Chen et al., 2023) adopts a staged routing framework in which inputs are first processed by the small model and are escalated to stronger models only when necessary. We implement Strategy 3 from the original paper, which determines routing decisions based on

Method	MMMU	AOKVQA	ScienceQA	ChartQA	MathVision	Avg.
Qwen2.5-VL-7B-Instruct						
MoT	67.52	72.35	72.08	75.38	82.59	73.18
Cascade Routing	68.32	73.15	76.31	69.58	78.25	73.12
RouteLLM	71.33	68.51	82.83	74.89	87.53	77.02
Post-Hoc Embed	70.30	69.92	75.45	72.75	88.14	75.31
Probe	76.51	82.03	83.95	82.51	90.28	83.86
BEST-Route	77.36	80.10	83.02	83.11	90.46	82.81
Attention Probe (ours)	77.10	84.34	86.43	85.43	88.92	84.84
ReLope (ours)	**80.89**	**86.17**	**91.79**	**86.51**	**95.68**	**88.21**
Gemma3-12B						
MoT	62.37	68.05	72.84	70.16	65.39	67.76
Cascade Routing	62.52	64.91	75.58	73.29	68.10	68.88
RouteLLM	64.83	66.10	78.92	74.65	70.45	70.59
Post-Hoc Embed	65.68	71.92	76.19	73.42	69.07	71.26
Probe	68.24	75.13	80.47	76.88	72.26	74.60
BEST-Route	69.12	72.08	78.61	76.02	72.34	73.63
Attention Probe (ours)	70.39	74.58	84.93	81.32	71.19	76.48
ReLope (ours)	**76.23**	**85.73**	**88.73**	**87.95**	**81.77**	**84.88**
Phi-4-Multimodal-Instruct						
MoT	63.91	68.42	74.03	67.17	73.28	69.36
Cascade Routing	63.79	70.04	75.56	62.93	76.84	69.83
RouteLLM	66.12	70.81	84.13	70.46	78.55	74.41
Post-Hoc Embed	67.28	72.66	77.84	70.58	77.12	73.10
Probe	69.97	76.23	82.08	73.76	81.03	76.61
BEST-Route	71.50	75.12	80.90	70.11	81.46	75.82
Attention Probe (ours)	72.68	80.27	87.24	78.63	83.47	80.46
ReLope (ours)	**78.75**	**85.90**	**92.36**	**81.47**	**91.16**	**85.93**

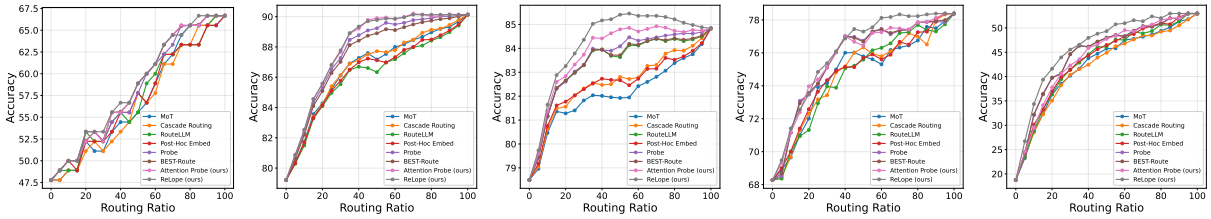
Table 2: AUC (%) comparison on five multimodal benchmarks and three MLLM backbones.

model confidence signals. **RouteLLM** (Ong et al., 2024) learns a routing function that ranks candidate models according to expected performance and cost. We use the SW Ranking strategy to estimate routing preferences, which is generally better than other variants according to results in the original work. **Post-Hoc Embed** (Gupta et al., 2024) extracts the output embedding of the small model after generation. A lightweight classifier is trained on these embeddings to estimate whether the small model can correctly solve the input problem. **Probe** (Chuang et al., 2025) trains a supervised probe on intermediate hidden states of the model to estimate correctness probability, which serves as the routing signal. **BEST-Route** (Ding et al., 2025) allocates test-time compute adaptively through a performance–cost objective.

5.3 Comparison with Baselines

AUC performance comparison. Table 2 reports routing performance on five datasets and three backbone MLLMs: Qwen2.5-VL-7B-Instruct (Bai et al., 2025), Gemma3-12B (Team et al., 2025), and Phi-4-Multimodal-Instruct (Abouelenin et al., 2025). ReLope achieves the highest AUC on every dataset and backbone. On Qwen2.5-VL-7B-Instruct, ReLope attains an average AUC of 88.21, exceeding Probe and Attention Probe by 4.35 and 3.37. The improvement over Probe is also consistent on Gemma3-12B and Phi-4-Multimodal-Instruct. These results indicate that routing-aware hidden state adaptation improves routing performance beyond compute allocation alone.

Routing performance in the hybrid MLLM system. Following (Chuang et al., 2025), we eval-



(a) MMMU (b) AOKVQA (c) ScienceQA (d) ChartQA (e) MathVision

Figure 3: Overall system accuracy as a function of routing ratio. The routing ratio denotes the percentage of queries routed to the large model in a hybrid MLLM system. Higher curves indicate better accuracy–escalation-rate trade-offs by sending only the most difficult queries to the large model.

Router	ms/sample	Trainable params	Peak GB
Probe	97.6	0.96M	16.89
Attention Probe	98.2	0.96M	16.96
ReLope	99.0	29.54M	17.00

Table 3: Routing overhead on MMMU with Qwen2.5-VL-7B-Instruct (A100-80GB, fp16, batch size 4).

Router	LoRA	VIB	AOKVQA	MMMU
Probe	–	–	82.03	76.51
LoRA-only Probe	Yes	–	85.12	78.00
VIB-only Probe	–	Yes	84.45	78.97
ReLope	Yes	Yes	**86.17**	**80.89**

Table 4: Component controls with Qwen2.5-VL-7B-Instruct at layer 24 (AUC, %).

uate the accuracy–routing-rate trade-off in a hybrid MLLM system composed of Qwen2.5-VL-7B-Instruct and a stronger large model. We use GPT-4.1 for most datasets and GPT-5.1 (OpenAI, 2026) for MathVision. Given a routing ratio of $h\%$, each method ranks queries by its routing score and sends the top-$h\%$ hardest queries to the large model, while the remaining queries are answered by the small model. When escalation uses a paid API and local inference cost is fixed, this ratio is a proxy for deployment cost; it is not a direct monetary measurement.

Figure 3 reports overall system accuracy as the routing ratio varies. When the ratio is 0, all queries are handled by the small model; when it is 100, all queries are routed to the large model. ReLope consistently traces the strongest tradeoff across all datasets. Its advantage is most evident at low routing ratios, where it gains higher accuracy under the same escalation budget.

Table 3 measures the previously implicit efficiency claim. ReLope adds 1.4 ms (1.4%) and 0.11 GB over a standard Probe in this end-to-end measurement; its parameter increase has limited latency effect because the MLLM forward pass dominates runtime.

5.4 Ablations and Discussion

Separating design from additional capacity. Table 4 directly controls the two additions to the frozen-feature Probe. Both LoRA-only ($\beta = 0$) and VIB-only improve over Probe, while their combination performs best on both datasets. Together with the matched representation intervention in Figure 2, these results associate each design choice with the observed failure mode: LoRA improves separability of routing features, while the VIB objective suppresses variation that is unhelpful for correctness prediction.

Additional analyses. ReLope remains stable over three seeds on MMMU (AUC 80.79 ± 0.40, versus 76.61 ± 0.20 for Probe). Results under visual perturbation, leave-one-dataset-out evaluation, and parameter sensitivity are provided in Appendix B to preserve main-paper space.

6 Conclusion

We study the routing problem in MLLMs and show that probe-based methods, which are effective in text-only LLMs, degrade substantially in the multimodal setting. To address this issue, two methods are proposed. The Attention Probe aggregates hidden states through the attention mechanism to enrich the probe input, while ReLope introduces a LoRA adapter with a KL regularizer to learn routing-aware representations. Extensive experiments across multiple benchmarks and backbones show consistent improvements over strong baselines. These results highlight the importance of learning a routing-aware probe in MLLMs. More broadly, the findings suggest that improving the structure of hidden states is a promising direction for reliable routing in multimodal systems.

7 Limitations

This work has several limitations. First, although the experiments cover multiple benchmarks and backbone MLLMs, the evaluation is still limited to a small set of tasks and model families. Second, ReLope relies on supervised correctness labels and introduces additional tuning overhead, which may limit scalability in new domains. Finally, the HSIC/CKA analysis provides evidence that multimodal last-token states carry weaker correctness dependence than caption-based inputs, but it does not fully establish causality or characterize all sources of multimodal routing difficulty.

References

Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, and 1 others. 2025. Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras. *arXiv preprint arXiv:2503.01743*.

Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. 2016. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

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

Zhijun Chen, Jingzheng Li, Pengpeng Chen, Zhuoran Li, Kai Sun, Yuankai Luo, Qianren Mao, Dingqi Yang, Hailong Sun, and Philip S Yu. 2025. Harnessing multiple large language models: A survey on llm ensemble. *arXiv preprint arXiv:2502.18036*.

Yu-Neng Chuang, Leisheng Yu, Guanchu Wang, Lizhe Zhang, Zirui Liu, Xuanting Cai, Yang Sui, Vladimir Braverman, and Xia Hu. 2025. Confident or seek stronger: Exploring uncertainty-based on-device llm routing from benchmarking to generalization. *arXiv preprint arXiv:2502.04428*.

Dujian Ding, Ankur Mallick, Shaokun Zhang, Chi Wang, Daniel Madrigal, Mirian Del Carmen Hipolito Garcia, Menglin Xia, Laks V. S. Lakshmanan, Qingyun Wu, and Victor Rühle. 2025. BEST-Route: Adaptive LLM routing with test-time optimal compute. *arXiv preprint arXiv:2506.22716*.

Song Guo, Lei Zhang, Xiawu Zheng, Yan Wang, Yuchao Li, Fei Chao, Chenglin Wu, Shengchuan Zhang, and Rongrong Ji. 2023. Automatic network pruning via Hilbert–Schmidt independence criterion lasso under information bottleneck principle. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17458–17469.

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

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Yuheng Huang, Jiayang Song, Zhijie Wang, Shengming Zhao, Huaming Chen, Felix Juefei-Xu, and Lei Ma. 2023. Look before you leap: An exploratory study of uncertainty measurement for large language models. *arXiv preprint arXiv:2307.10236*.

Wittawat Jitkrittum, Harikrishna Narasimhan, Ankit Singh Rawat, Jeevesh Juneja, Congchao Wang, Zifeng Wang, Alec Go, Chen-Yu Lee, Pradeep Shenoy, Rina Panigrahy, and 1 others. 2025. Universal model routing for efficient llm inference. *arXiv preprint arXiv:2502.08773*.

Zijing Liang, Yanjie Xu, Yifan Hong, Penghui Shang, Qi Wang, Qiang Fu, and Ke Liu. 2024. A survey of multimodel large language models. In *Proceedings of the 3rd international conference on computer, artificial intelligence and control engineering*, pages 405–409.

Ji Lin, Hongxu Yin, Wei Ping, Yao Lu, Pavlo Molchanov, Andrew Tao, Huizi Mao, Jan Kautz, Mohammad Shoeybi, and Song Han. 2024. VILA: On pre-training for visual language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

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

Matéo Mahaut, Laura Aina, Paula Czarnowska, Momchil Hardalov, Thomas Müller, and Lluís Màrquez. 2024. Factual confidence of llms: on reliability and robustness of current estimators. *arXiv preprint arXiv:2406.13415*.

Ahmed Masry, Xuan Long Do, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the association for computational linguistics: ACL 2022*, pages 2263–2279.

Alireza Mohammadshahi, Arshad Rafiq Shaikh, and Majid Yazdani. 2024. Routoo: Learning to route to large language models effectively. *arXiv preprint arXiv:2401.13979*.

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

OpenAI. 2026. Chatgpt. https://chat.openai.com/.

Pranoy Panda, Raghav Magazine, Chaitanya Devaguptapu, Sho Takemori, and Vishal Sharma. 2025. Adaptive llm routing under budget constraints. *arXiv preprint arXiv:2508.21141*.

Tanik Saikh, Tirthankar Ghosal, Amish Mittal, Asif Ekbal, and Pushpak Bhattacharyya. 2022. Scienceqa: A novel resource for question answering on scholarly articles. *International Journal on Digital Libraries*, 23(3):289–301.

Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. 2022. A-okvqa: A benchmark for visual question answering using world knowledge. In *European conference on computer vision*, pages 146–162. Springer.

Soham Shah and Kumar Shridhar. 2025. Select-then-route: Taxonomy guided routing for llms. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 425–441.

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

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, and 1 others. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Naftali Tishby and Noga Zaslavsky. 2015. Deep learning and the information bottleneck principle. In *2015 ieee information theory workshop (itw)*, pages 1–5. Ieee.

Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. 2024. Measuring multimodal mathematical reasoning with math-vision dataset. *Advances in Neural Information Processing Systems*, 37:95095–95169.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. 2024. A survey on multimodal large language models. *National Science Review*, 11(12):nwae403.

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

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, and 1 others. 2024. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9556–9567.

Haozhen Zhang, Tao Feng, and Jiaxuan You. 2025. Router-r1: Teaching llms multi-round routing and aggregation via reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, and 1 others. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2).

Zesen Zhao, Shuowei Jin, and Z Morley Mao. 2024. Eagle: Efficient training-free router for multi-llm inference. *arXiv preprint arXiv:2409.15518*.

A Implementation Details

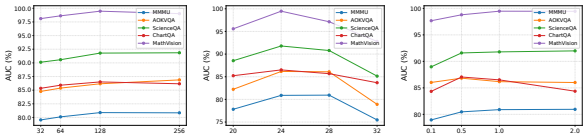
All experiments are implemented in PyTorch using the transformers and peft libraries. For probe-based methods (Probe, Post-Hoc Embed, Attention Probe, and ReLope), a five-layer MLP probe is used. Training uses the AdamW optimizer with learning rate 1×10^{-4}. Hidden states are extracted from an intermediate transformer layer l of the small model f_θ. Layer $l = 24$ is used for Qwen2.5-VL-7B-Instruct and Phi-4-Multimodal-Instruct (32 layers), and layer $l = 40$ for Gemma3-12B (48 layers). For ReLope, a LoRA adapter with rank $r = 128$ and scaling factor $\alpha = 256$ is inserted into the selected transformer layer. Two linear heads predict $\boldsymbol{\mu}$ and $\log \boldsymbol{\sigma}^2$ from the LoRA-adapted feature $\mathbf{z}_\mathrm{L}$. The probe receives $\mathbf{z}_\mathrm{B}$ in both phases: a reparameterized sample during training and $\mathbf{z}_\mathrm{B} = \boldsymbol{\mu}$ during inference. The loss combines binary cross-entropy and KL divergence with $\beta = 1.0$. For Attention Probe, its single routing query vector is shared by all examples and trained jointly with the MLP while the backbone remains frozen. All experiments run on NVIDIA A100 GPUs with 80 GB memory.

Correctness labels. Routing labels indicate whether the small MLLM answers an example correctly. For multiple-choice tasks (MMMU, ScienceQA, and A-OKVQA), we normalize model output and match the extracted option to the gold option. For ChartQA, we use relaxed exact matching with numerical tolerance for numeric answers and normalized string matching for textual answers. For MathVision, we extract the final answer and apply symbolic-equivalence checking. No judge model is used to generate routing labels.

Latency measurement. The runtime experiment in Table 3 uses Qwen2.5-VL-7B-Instruct on MMMU with an NVIDIA A100-80GB GPU, fp16 inference, and batch size 4. It measures end-to-end wall-clock latency including the backbone forward pass and router computation. The bottleneck heads parameterize $q_\psi(\mathbf{z}_\mathrm{B} \mid \mathbf{z}_\mathrm{L})$; at inference we set $\mathbf{z}_\mathrm{B} = \boldsymbol{\mu}$ before applying g_ϕ.

B Additional Results

Parameter sensitivity. Figure 4 shows that increasing the LoRA rank improves routing AUC up to $r = 128$, after which gains are marginal. Intermediate and later transformer layers provide stronger routing signals than early layers. Moderate bottleneck weights ($\beta \in [0.5, 1.0]$) yield the best trade-off between compression and prediction.



(a) LoRA rank r (b) Layer l (c) Coefficient β

Figure 4: Parameter sensitivity of ReLope: LoRA rank, probe layer, and bottleneck coefficient. Results are AUC on five multimodal benchmarks.

Method	Clean	Noise	JPEG	Blur	ΔAUC↓
Probe	82.03	77.31	76.84	76.05	5.30
Attention Probe	84.24	82.11	81.65	80.90	2.69
ReLope	**86.17**	**86.05**	**84.21**	**83.92**	**1.44**

Table 5: Routing AUC under image perturbations on A-OKVQA. ΔAUC is the average drop from clean inputs.

Robustness to visual perturbations. We add Gaussian noise, JPEG compression, and Gaussian blur to A-OKVQA test images while retaining prompts and labels. As shown in Table 5, ReLope has the smallest mean AUC drop, consistent with its bottleneck suppressing some task-irrelevant visual variation.

Method	MMMU	AOKVQA	ScienceQA	ChartQA	MathVision
MoT	67.52	72.35	72.08	75.38	**82.59**
Cascade Routing	67.15	72.94	75.33	65.47	63.28
RouteLLM	69.28	65.57	75.98	64.38	61.16
Post-Hoc Embed	67.55	66.18	69.81	65.75	64.39
Probe	71.78	81.92	80.57	73.05	66.35
Attention Probe	73.22	81.87	81.14	73.98	66.42
ReLope	**76.66**	**83.68**	**81.58**	**75.57**	66.83

Table 6: Leave-one-dataset-out OOD routing AUC: each target dataset is excluded from probe training.

Cross-dataset generalization. Table 6 reports leave-one-dataset-out evaluation. ReLope performs best on four targets and is the strongest probe-based approach on MathVision, although the non-learned MoT baseline is substantially stronger there. This gap suggests that mathematical visual reasoning remains a challenging domain shift for learned routers.