RELOPE: FROM HIDDEN-STATE PROBING TO A DECISION MODULE FOR MULTIMODAL LLM ROUTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Routing balances performance and cost in hybrid systems by escalating selected queries from a lightweight model to a powerful but expensive one. Hidden-state probes provide an effective routing signal by predicting the small model's correctness from hidden states it already computes. Although effective in text-only settings, their behavior on multimodal LLMs (MLLMs) is less understood. We find that correctness is harder to predict from hidden states when questions are paired with images rather than with captions of those images. Two training-free dependence measures, HSIC and label CKA, show the same pattern. We first investigate whether changing hidden states used by the probe improves correctness prediction for MLLMs. We propose Attention Probe, which learns an attention query to pool frozen token states into a routing feature and improves over the standard probe. However, pooling only recombines hidden states computed for answer generation, not for deciding whether the answer should be trusted. We therefore build a decision layer inside the existing MLLM and train it for the routing decision. The resulting router, ReLope (KL-Regularized LoRA Probe), adapts a separate copy of one MLLM layer with LoRA and regularizes it with a KL-penalized stochastic bottleneck. This decision branch shares the frozen lower layers and leaves the model's answer-generation path unchanged. ReLope attains the highest correctness prediction AUC on all five benchmarks and three backbones we evaluate and improves the accuracy versus cost trade-off at negligible overhead. These results suggest a new perspective and a practical technical path on routing: rather than reading decisions off states built for generation, routing decisions can be learned by a dedicated decision layer inside the model.

1 INTRODUCTION

Routing combines the efficiency of a small language model with the capability of a larger one by escalating only selected queries to the larger model (Chen et al., 2023; Ding et al., 2024; Chen et al., 2025). Hidden-state probing offers an efficient routing signal: a lightweight classifier predicts whether the small model's answer is correct from its hidden states (Kadavath et al., 2022; Azaria & Mitchell, 2023; Chuang et al., 2025). Because these hidden states contain information predictive of correctness (Orgad et al., 2025; Chen et al., 2024a), probes route effectively in text-only settings without extra generations or an additional model call to estimate confidence.

It is less clear whether this approach transfers equally well to multimodal LLMs (MLLMs). Prior work shows that vision-language models are often worse calibrated on visual inputs than on semantically equivalent text (Xuan et al., 2025; Zou et al., 2025), and that internal probes transferred across medical VQA domains can perform near chance (Khanmohammadi et al., 2026). These findings raise concerns about multimodal reliability, but do not directly address how effectively lightweight routing probes can predict correctness from frozen MLLM hidden states.

We find that correctness is harder to predict from hidden states when a question comes with an image than when it comes with a caption of that image. Table 1 shows this on A-OKVQA, where we keep each question fixed and replace its image with a caption. Using the textual caption, the small model answers less accurately (68.73% vs. 78.78%), yet its correctness is easier to predict (probe AUC 85.25 vs. 81.94). The gap is not specific to the trained probe: HSIC (Gretton et al., 2005) and label CKA (Kornblith et al., 2019), which measure the dependence between hidden states and correctness

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Table 1: **Correctness becomes harder to probe with multimodal input.** Three input conditions on A-OKVQA with Qwen2.5-VL-7B-Instruct, differing in evidence source and the presence of image tokens. See Appendix C.3 for examples. HSIC and label CKA are dependence measures between hidden states and correctness labels that require no classifier training.

Input	Evidence	Image tokens	LLM Acc. (%)	Probe AUC	HSIC ($\times 10^{-3}$)	Label CKA
Original input	image	✓	**78.78**	81.94 ± 0.58	2.115	0.1123
Caption only	text	–	68.73	$\mathbf{85.25} \pm 0.52$	**9.842**	**0.2171**
Caption + random image	text	✓	68.30	84.73 ± 1.08	8.364	0.1952

labels without training a classifier, are also higher with captions. The same pattern holds on two other benchmarks (Appendix C). With image inputs, the correctness signal is therefore harder to extract from the hidden states themselves.

Motivated by this finding, we ask whether changing the hidden state the probe reads can improve correctness prediction. A standard probe reads the last-token hidden state at a selected layer. The MLLM produces this state for answer generation, not targeted to expose whether its answer is correct, and with multimodal inputs, the state must encode both visual and textual content. We therefore first ask whether learning how to combine token hidden states helps. We propose Attention Probe, which uses a learned attention query to weigh the frozen token hidden states and passes their weighted sum to the correctness classifier. The query and classifier are trained jointly on correctness labels, while the MLLM remains frozen. On A-OKVQA, Attention Probe reduces the AUC gap between image and caption inputs from 3.31 to 1.25 points (Section 5.4), suggesting that how the hidden states are pooled contributes substantially to the observed gap.

However, learned pooling only changes how frozen hidden states are combined. The states themselves are still shaped for answer generation, not for deciding whether that answer should be trusted. Prior work finds that correctness prediction improves when the model itself is trained for it rather than probed frozen (Kadavath et al., 2022; Kapoor et al., 2024), and Jev (Almeida, 2026) makes a related argument at the system level: LLMs are optimized to generate text, not to make calibrated decisions. Sharing this principle, rather than reading the routing decision off these states, we build a decision layer inside the existing MLLM and train it for the routing decision. We call the resulting router ReLope (KL-Regularized LoRA Probe). It inserts LoRA (Hu et al., 2022) adapters into a separate copy of the selected layer, and trains this copy for correctness classification. The copy reuses the model's frozen lower layers and forms a routing branch, leaving the answer-generating model unchanged. Correctness supervision can thus shape the routing hidden states without changing the small model's answers or their correctness labels. Adaptation, however, also raises the risk of fitting input-specific details that do not generalize. We therefore pass the adapted last-token hidden state through a stochastic bottleneck with a KL penalty (Alemi et al., 2016). Together, the prediction loss and the KL penalty encourage the routing feature to retain information useful for correctness prediction while discarding unnecessary input-specific detail. Neither method requires captions, extra generations, or a large-model call before the routing decision.

Across five multimodal benchmarks and three backbones, ReLope achieves the highest mean correctness prediction AUC among the evaluated methods in all 15 dataset–backbone pairs. It also improves the hybrid system's accuracy–cost trade-off with low inference overhead. Our contributions are:

- An empirical analysis showing that correctness is harder to predict from MLLM hidden states under image inputs than under caption inputs, together with evidence that learned pooling substantially narrows this gap.
- Attention Probe, which learns how to pool frozen hidden states for correctness prediction, and ReLope, which combines adaptation of one layer with a KL-regularized stochastic bottleneck while preserving the small model's answers.
- An evaluation across five benchmarks and three backbones covering correctness prediction, hybrid accuracy and cost, inference overhead, component ablations, robustness, and cross-dataset transfer.

2 Related Work

Model selection and routing. Input-based routers choose a model from the input query alone, using routers trained on benchmark evaluation results, reward signals, preference data, or learned

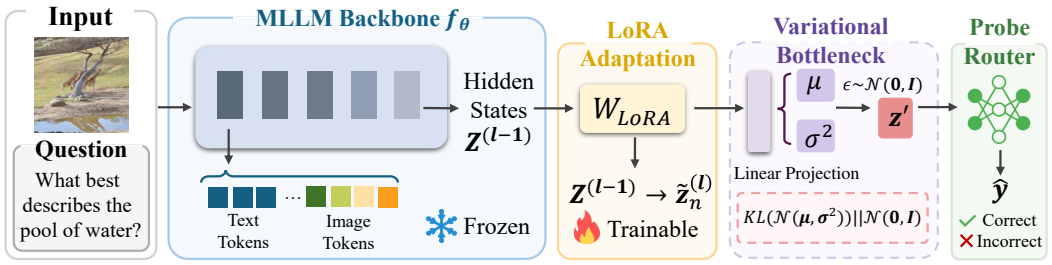


Figure 1: **ReLope overview.** Token states $\mathbf{Z}^{(l-1)}$ of the frozen small model enter a copy of layer l with LoRA adapters. The adapted last-token state is projected to a Gaussian $(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$, a sample from which is classified for correctness. A KL penalty to a standard normal prior regularizes the latent feature. LoRA weights, projection heads, and classifier are trained together with the routing loss.

taxonomies (Shnitzer et al., 2023; Lu et al., 2023; Ong et al., 2024; Mohammadshahi et al., 2024; Shah & Shridhar, 2025; Panda et al., 2025). Eagle ranks models without training a router (Zhao et al., 2024), and Router-R1 chains several model calls and aggregates their answers (Zhang et al., 2025). Universal routers treat routing and cascading as one problem (Jitkrittum et al., 2025). We study the two-model decision after the small model has run, using features it has already computed.

Cascades and probes. Cascades defer selected queries after a smaller model has answered (Chen et al., 2023; Yue et al., 2023), relating routing to selective prediction (Chow, 1970; Geifman & El-Yaniv, 2017). Deferral signals include output probabilities, token-level uncertainty and answer agreement (Huang et al., 2023; Gupta et al., 2024), output embeddings, and hidden-state probes (Kadavath et al., 2022; Azaria & Mitchell, 2023; Chuang et al., 2025). Their reliability for factual correctness estimation has also been studied directly (Mahaut et al., 2024). Probe performance depends on both the hidden state and the classifier, so limited decoding accuracy does not by itself establish the absence of information (Alain & Bengio, 2016; Belinkov, 2022; Hewitt & Liang, 2019). We study readout learning and supervised adaptation of the hidden states for multimodal routing.

Closest to our design, Kapoor et al. (2024) find that probes on frozen features and prompting give weaker uncertainty estimates than fine-tuning on a small set of graded answers. Their estimator is text-only and works by prompting the fine-tuned model with an uncertainty query. ReLope instead adapts a copy of a single layer on top of the frozen lower layers without interfering with the model itself. At the system level, Jev (Almeida, 2026) argues that decision tasks call for models trained for calibrated decisions rather than for text generation. ReLope applies a similar principle within a lightweight one-layer branch of an existing MLLM.

Multimodal confidence and hidden states. MLLMs form hidden states from both visual and textual tokens (Yin et al., 2024), while some benchmark questions remain answerable without the image (Chen et al., 2024b). Reliability therefore requires more than high answer accuracy. Xuan et al. (2025) show that verbalized calibration is often worse with visual inputs than with semantically equivalent text. Khanmohammadi et al. (2026) find that confidence estimation remains unreliable across medical VQA domains when probes trained on natural images are transferred without adaptation. VIB-Probe (Zhang et al., 2026) uses a variational bottleneck on attention-head outputs for hallucination detection and generation-time intervention. ReLope instead predicts correctness from prefill hidden states for multimodal routing, jointly training an adapted layer and a stochastic bottleneck in a separate branch while keeping the answer-generating model frozen.

3 Preliminaries

Problem statement. We consider a hybrid MLLM system consisting of a small, cheap model and a larger, more capable model. Given a multimodal input $\mathbf{x}$ containing a text prompt and an image, the small model produces an answer $h_s(\mathbf{x})$. The binary label y indicates whether the small model's answer is correct. A router assigns a correctness score $s(\mathbf{x})$, with higher values indicating greater confidence in the small model's answer. The system retains that answer when $s(\mathbf{x}) \geq \tau$ and otherwise escalates the query to the large model. The threshold τ controls the escalation rate.

Hidden-state probe. At layer l the model produces token hidden states $\mathbf{Z}^{(l)} = (\mathbf{z}_1^{(l)}, \ldots, \mathbf{z}_n^{(l)}) \in \mathbb{R}^{n \times d}$. The standard probe (Alain & Bengio, 2016; Chuang et al., 2025) reads the last-token state and trains a classifier g_ϕ with binary cross-entropy (BCE) against y: $s_{\mathrm{P}}(\mathbf{x}) = \mathrm{sigmoid}\big(g_\phi(\mathbf{z}_n^{(l)})\big)$. The backbone stays frozen, and the last-token readout is fixed by the probe design. The classifier learns to predict correctness from this hidden state without changing how it is formed. Its performance reflects both the available signal and how readily the chosen classifier can extract it.

Correctness prediction under multimodal input. We compare image and caption inputs to examine how the input modality affects hidden-state probing. Using Qwen2.5-VL-7B-Instruct on the same A-OKVQA questions, we consider three conditions: the original image, a caption replacing the image, and the caption paired with an unrelated image. The third condition tests whether adding image tokens affects correctness prediction when the relevant evidence is still provided as text. Answers and correctness labels are recomputed for each condition.

As shown in Table 1, replacing images with captions increases probe AUC from 81.94 to 85.25 while lowering answer accuracy from 78.78% to 68.73%. Since probe AUC also depends on the classifier and its training, we additionally measure dependence between hidden states and correctness labels using HSIC (Gretton et al., 2005) and label CKA (Kornblith et al., 2019). Neither measure requires classifier training, and both yield higher values in the caption-only condition, consistent with the probe results. The same image-versus-caption pattern appears on two additional benchmarks (Appendix C). These comparisons are descriptive, as changing the input also changes the answers and correctness labels.

4 Learning the Routing Feature

Table 1 shows that correctness is harder to probe under multimodal input, and the classifier-free measures place the difficulty outside the classifier. What is left is the hidden state the probe reads directly from the MLLM, the last-token state of layer l. Motivated by these observations, we investigate two ways to learn the hidden state used for correctness prediction. Both methods use the frozen token hidden states $\mathbf{Z}^{(l-1)}$ as their starting point. Attention Probe learns an attention query that pools these hidden states into an input for the correctness classifier. ReLope instead passes them through a separately adapted copy of layer l, then applies a stochastic bottleneck to the adapted last-token hidden state. Both methods use the same classifier architecture g_ϕ, with BCE supervising correctness prediction; ReLope additionally uses KL regularization. Figure 1 illustrates ReLope, and Algorithm 1 summarizes training for both methods.

Algorithm 1: Training Attention Probe or ReLope (separate runs)

Input: Correctness dataset $\mathcal{D} = \{(\mathbf{x}, y)\}$; frozen backbone f_θ; read layer l; chosen router
for *each training epoch* **do**
  for *each minibatch* $(\mathbf{x}, y)$ **do**
    $\mathbf{Z}^{(l-1)} \leftarrow f_\theta^{(<l)}(\mathbf{x})$;
    if *the chosen router is Attention Probe* **then**
      Compute $\boldsymbol{\omega}$ and $\mathbf{z}_{\mathrm{A}}$ with Eq. 1;
      $\mathcal{L} \leftarrow \mathrm{BCE}(y, \mathrm{sigmoid}(g_\phi(\mathbf{z}_{\mathrm{A}})))$; update $(\mathbf{q}, \phi)$;
    else
      $\tilde{\mathbf{Z}}^{(l)} \leftarrow f^{(l)}_{\theta, \theta_{\mathrm{LoRA}}}(\mathbf{Z}^{(l-1)})$; $\mathbf{z}_{\mathrm{L}} \leftarrow \tilde{\mathbf{z}}_n^{(l)}$;
      $(\boldsymbol{\mu}, \log \boldsymbol{\sigma}^2) \leftarrow \psi(\mathbf{z}_{\mathrm{L}})$; sample $\mathbf{z}_{\mathrm{B}}$ with Eq. 2;
      Compute $\mathcal{L}_{\mathrm{R}}$ with Eq. 3; update $(\theta_{\mathrm{LoRA}}, \psi, \phi)$;

// At inference ReLope uses $\mathbf{z}_{\mathrm{B}} = \boldsymbol{\mu}$.

4.1 Attention Probe: learn where to read

Motivation. The hidden state the probe reads is the last-token state of layer l, the strongest fixed readout in our sweep over positions and layers (Section 5.4). However, the MLLM produces this hidden state for answer generation, without an explicit objective to make answer correctness easy to predict. We therefore investigate whether learning how to pool token hidden states specifically for correctness prediction improves probe performance.

Design. Attention Probe applies attention pooling to the frozen token hidden states at layer $l-1$. A learned query $\mathbf{q} \in \mathbb{R}^d$ assigns a score to each token hidden state. A softmax converts these scores into weights, and the resulting weighted sum is passed to the correctness classifier:

$$\omega_i = \frac{\exp(\mathbf{q}^\top \mathbf{z}_i^{(l-1)}/\sqrt{d})}{\sum_{j=1}^{n} \exp(\mathbf{q}^\top \mathbf{z}_j^{(l-1)}/\sqrt{d})}, \qquad \mathbf{z}_{\mathrm{A}} = \sum_{i=1}^{n} \omega_i \mathbf{z}_i^{(l-1)}, \qquad s_{\mathrm{A}}(\mathbf{x}) = \mathrm{sigmoid}\big(g_\phi(\mathbf{z}_{\mathrm{A}})\big). \quad (1)$$

The sum includes all valid text and image tokens, allowing the probe to draw on hidden states from both modalities and across token positions. To keep the pooling mechanism lightweight, we use a single learned query $\mathbf{q}$ shared across samples and token positions. This introduces only d trainable pooling parameters and applies the same scoring rule to every token, while the resulting weights adapt to each input's hidden states. We scale the dot-product scores by $1/\sqrt{d}$ to reduce the risk of softmax saturation. Initializing $\mathbf{q}$ near zero gives approximately uniform weights, so the probe starts from a readout close to mean pooling and learns to emphasize different token hidden states through correctness supervision.

Training updates only the query $\mathbf{q}$ and classifier parameters ϕ. The MLLM and its hidden states remain frozen. The classifier architecture matches that of the standard probe, allowing us to compare a fixed hidden state with learned attention pooling while keeping the backbone unchanged. Section 5.4 examines how learned pooling affects correctness prediction under image and caption inputs.

4.2 ReLope: learn what to encode

Motivation. Attention Probe learns how to combine frozen hidden states, but the states themselves are still computed for answer generation. Sharing the principle of training models specifically for decisions (Almeida, 2026; Kapoor et al., 2024), we build a decision layer inside the existing MLLM and train it for the routing decision, so that correctness supervision shapes what the routing feature encodes. To keep this cheap, the decision layer is a separate copy of layer l with LoRA adapters that reuses the frozen lower layers, leaving the model's original answer path unchanged.

Adaptation must preserve the small model's answers and add little computation. Prior work shows that visual tokens can contain redundancy and information irrelevant to the question (Yang et al., 2025; Cao et al., 2023). Motivated by these findings, we hypothesize that retaining information relevant to correctness while limiting unrelated visual details can make correctness easier to predict from hidden states. ReLope therefore combines single-layer adaptation with a KL-regularized stochastic bottleneck. Correctness supervision guides the adapted hidden state toward correctness prediction, while the bottleneck limits how much input-specific information the routing feature retains.

Layer adaptation. We insert LoRA adapters into a separate copy of layer l while freezing its original weights. For an adapted weight matrix, the update is $\Delta\mathbf{W} = (\alpha/r)\mathbf{B}\mathbf{A}$ with rank at most r (Hu et al., 2022). The adapted layer maps $\mathbf{Z}^{(l-1)}$ to $\tilde{\mathbf{Z}}^{(l)}$, from which we take the last-token state $\mathbf{z}_{\mathrm{L}} = \tilde{\mathbf{z}}_n^{(l)}$. The routing branch shares the lower layers with the original answer path and adds one layer computation. Retaining the last-token interface allows us to evaluate adaptation of the hidden states without simultaneously changing the token selection rule.

Stochastic bottleneck. Our HSIC and label CKA analysis examines the dependence between frozen hidden states and correctness labels. This motivates learning a routing feature that retains information predictive of correctness. To regularize this learning process, we adopt the information bottleneck principle: preserve information useful for correctness prediction while limiting how much information the feature retains from the adapted hidden state (Tishby & Zaslavsky, 2015; Alemi et al., 2016). We implement this trade-off with a stochastic bottleneck trained using correctness supervision and a KL penalty. Specifically, two linear heads parameterized by ψ map $\mathbf{z}_{\mathrm{L}}$ to $\boldsymbol{\mu}$ and $\log \boldsymbol{\sigma}^2$, defining a Gaussian distribution:

$$q_\psi(\mathbf{z}_{\mathrm{B}} \mid \mathbf{z}_{\mathrm{L}}) = \mathcal{N}(\boldsymbol{\mu}, \mathrm{diag}(\boldsymbol{\sigma}^2)), \qquad \mathbf{z}_{\mathrm{B}} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \quad (2)$$

During training, we sample $\mathbf{z}_{\mathrm{B}}$ using the reparameterization trick (Kingma & Welling, 2014), allowing gradients to propagate through the stochastic bottleneck. At inference, we use the mean $\boldsymbol{\mu}$ instead of sampling, yielding the deterministic score $s_{\mathrm{R}}(\mathbf{x}) = \mathrm{sigmoid}(g_\phi(\boldsymbol{\mu}))$.

Training objective. We minimize the correctness loss with a KL penalty toward a fixed prior,

$$\mathcal{L}_{\mathrm{R}} = \mathbb{E}_{(\mathbf{x},y)}\Big[\mathbb{E}_{q_\psi}\big[\mathrm{BCE}\big(y, \mathrm{sigmoid}(g_\phi(\mathbf{z}_{\mathrm{B}}))\big)\big] + \beta D_{\mathrm{KL}}(q_\psi(\mathbf{z}_{\mathrm{B}} \mid \mathbf{z}_{\mathrm{L}}) \,\|\, \mathcal{N}(\mathbf{0}, \mathbf{I}))\Big]. \quad (3)$$

Table 2: **Correctness prediction.** Routing AUC (%) on five benchmarks and three small backbones, mean ± standard deviation over three seeds. Bold marks the best mean per column.

Method	MMMU	A-OKVQA	ScienceQA	ChartQA	MathVision
Qwen2.5-VL-7B-Instruct					
MMR-MLP	63.50 ± 0.23	61.40 ± 0.57	80.05 ± 0.13	71.54 ± 0.18	77.58 ± 1.19
MMR-CMR	62.11 ± 1.15	64.77 ± 2.04	82.95 ± 0.29	73.73 ± 0.27	80.25 ± 2.04
MoT	66.97 ± 1.24	72.26 ± 0.62	71.83 ± 0.89	76.22 ± 2.08	81.89 ± 1.64
Cascade Routing	68.41 ± 1.14	73.13 ± 0.08	76.15 ± 0.47	69.75 ± 2.21	78.20 ± 1.00
RouteLLM	71.02 ± 1.36	68.74 ± 0.65	82.78 ± 0.15	75.35 ± 1.10	87.38 ± 0.88
Post-Hoc Embed	70.29 ± 0.03	69.97 ± 0.76	75.41 ± 0.35	72.01 ± 2.24	87.73 ± 1.99
Probe	76.61 ± 0.20	81.94 ± 0.58	83.65 ± 0.86	81.59 ± 2.35	90.03 ± 2.05
BEST-Route	77.51 ± 0.77	80.01 ± 0.37	82.73 ± 0.69	82.51 ± 2.03	90.33 ± 2.59
Attention Probe (ours)	78.36 ± 1.02	84.06 ± 0.65	86.11 ± 0.79	85.45 ± 1.43	89.16 ± 2.51
ReLope (ours)	**80.79 ± 0.40**	**86.24 ± 0.63**	**91.78 ± 0.52**	**85.94 ± 0.51**	**93.56 ± 0.95**
Gemma3-12B					
MMR-MLP	64.23 ± 3.89	61.62 ± 0.52	84.34 ± 0.04	72.10 ± 0.04	68.71 ± 0.25
MMR-CMR	64.49 ± 1.44	64.84 ± 1.43	86.70 ± 0.21	74.05 ± 0.27	72.28 ± 2.09
MoT	62.25 ± 0.47	68.11 ± 0.29	72.85 ± 0.07	69.75 ± 1.44	65.66 ± 0.96
Cascade Routing	62.47 ± 0.15	64.76 ± 0.62	75.85 ± 0.70	73.08 ± 0.71	67.75 ± 1.35
RouteLLM	65.25 ± 1.39	66.11 ± 0.13	78.90 ± 0.34	74.51 ± 2.39	70.37 ± 2.05
Post-Hoc Embed	65.57 ± 0.31	71.71 ± 0.47	76.13 ± 0.60	73.51 ± 1.63	68.38 ± 2.50
Probe	68.10 ± 0.46	75.10 ± 0.09	80.41 ± 0.26	77.21 ± 2.47	71.71 ± 1.26
BEST-Route	69.23 ± 0.41	71.90 ± 0.62	78.85 ± 1.00	75.75 ± 1.85	73.00 ± 2.99
Attention Probe (ours)	70.64 ± 1.46	74.49 ± 0.34	85.29 ± 1.04	81.02 ± 0.75	70.39 ± 2.65
ReLope (ours)	**76.82 ± 0.86**	**85.78 ± 0.12**	**88.65 ± 0.24**	**87.35 ± 1.38**	**81.26 ± 1.89**
Phi-4-Multimodal-Instruct					
MMR-MLP	56.44 ± 4.61	58.17 ± 1.18	75.07 ± 1.73	60.96 ± 1.37	75.81 ± 0.28
MMR-CMR	58.90 ± 1.44	59.33 ± 1.93	77.66 ± 1.98	63.06 ± 1.13	78.74 ± 0.52
MoT	64.41 ± 1.27	68.43 ± 0.15	74.08 ± 0.81	66.57 ± 1.70	73.61 ± 2.16
Cascade Routing	63.87 ± 0.30	70.11 ± 0.71	75.38 ± 0.67	63.41 ± 2.12	76.70 ± 1.13
RouteLLM	66.50 ± 1.05	71.04 ± 0.62	84.20 ± 0.32	70.31 ± 1.55	78.42 ± 0.94
Post-Hoc Embed	66.85 ± 1.00	72.75 ± 0.68	77.83 ± 0.03	70.00 ± 1.91	77.18 ± 1.28
Probe	69.66 ± 1.05	75.98 ± 0.58	82.00 ± 0.94	73.99 ± 0.53	81.86 ± 2.75
BEST-Route	71.55 ± 0.13	75.39 ± 0.77	80.71 ± 0.76	70.24 ± 2.18	80.68 ± 1.95
Attention Probe (ours)	72.51 ± 1.47	80.12 ± 0.69	87.20 ± 0.51	78.33 ± 0.89	83.82 ± 0.87
ReLope (ours)	**78.64 ± 0.54**	**85.89 ± 0.10**	**92.55 ± 0.63**	**81.64 ± 0.75**	**90.09 ± 2.16**

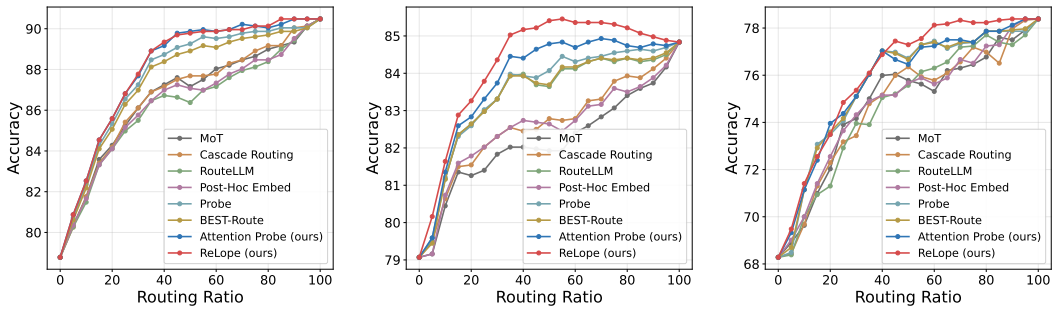
The BCE term encourages accurate correctness prediction. The expected KL term upper-bounds the mutual information between the adapted hidden state and the stochastic bottleneck output, thereby regularizing how much input-specific information the routing feature retains. The coefficient β controls the trade-off between prediction and compression. We jointly optimize the LoRA adapters, projection heads, and classifier, while keeping the answer-generating MLLM frozen. Section 5.5 evaluates prediction of clean-input correctness under image perturbations and finds that ReLope has the smallest mean AUC drop among the tested probes.

5 Experiments

We ask four questions. Does the learned feature predict correctness better than existing routers? Does that prediction improve a hybrid system, and at what cost? Where does the gain come from? And how far does it carry beyond the training distribution?

5.1 Setup

Datasets and models. We evaluate on five multimodal benchmarks: MMMU for general reasoning (Yue et al., 2024), A-OKVQA for knowledge-based visual question answering (Schwenk et al., 2022), the image subset of ScienceQA for science questions (Lu et al., 2022), ChartQA for chart



(a) A-OKVQA (b) ScienceQA (c) ChartQA

Figure 2: **Hybrid-system accuracy against escalation rate** with Qwen2.5-VL-7B-Instruct as the small model. At rate $h\%$ each router sends its $h\%$ least confident queries to the large model. The endpoints are the small model alone and the large model alone. A higher curve means more correct system answers for the same number of large-model calls.

understanding (Masry et al., 2022), and MathVision for visual mathematics (Wang et al., 2024). We use three small backbones: Qwen2.5-VL-7B-Instruct (Bai et al., 2025), Gemma3-12B (Team et al., 2025), and Phi-4-Multimodal-Instruct (Abouelenin et al., 2025).

Baselines. We compare with eight routers: MoT (Yue et al., 2023), Cascade Routing (Chen et al., 2023), RouteLLM (Ong et al., 2024), Post-Hoc Embed (Gupta et al., 2024), BEST-Route (Ding et al., 2025), MMR-MLP and MMR-CMR from MMR-Bench (Ma et al., 2026), and the standard hidden-state probe (Chuang et al., 2025). These methods use signals ranging from sampled-answer agreement and output statistics to query embeddings and hidden states. We adapt MMR-MLP and MMR-CMR to binary correctness prediction using frozen OpenCLIP ViT-B/32 embeddings.

Evaluation protocol. Correctness labels are obtained by comparing model answers with gold answers using dataset-specific evaluation rules. We evaluate correctness prediction using ROC-AUC, reporting the mean and standard deviation over three seeds unless otherwise stated. We assess routing utility through hybrid-system accuracy at different escalation rates and measure inference time and peak memory usage. Appendix A details data splits, training configurations, and answer evaluation.

5.2 Correctness prediction

Table 2 shows that frozen hidden states already provide useful routing signals: the standard probe outperforms most output-, sampling-, and preference-based baselines. Building on this baseline, Attention Probe learns to combine frozen token hidden states and improves over the standard probe in 12 of the 15 dataset–backbone pairs. ReLope improves over Attention Probe in all 15 pairs and achieves the highest mean AUC among the evaluated methods. These results suggest benefits from adapting and regularizing routing hidden states beyond just combining frozen states. Section 5.4 examines adaptation, the bottleneck, parameter count, and alternative readouts of frozen states.

5.3 From prediction to a hybrid system

A better prediction is only useful if it buys accuracy per large-model call. Figure 2 plots hybrid accuracy against escalation rate on A-OKVQA, ScienceQA, and ChartQA, using Qwen2.5-VL-7B-Instruct as the small model. The large model is GPT-4.1. Figure 4 in Appendix D.2 covers MMMU and MathVision. ReLope gives favorable trade-offs on all five tasks: for a given budget of large-model calls it returns more correct system answers than the other routers over most of the range. The curves do not follow AUC alone, since the large model can replace a correct small answer with a wrong one. Appendix D.1 converts the same comparison from escalation rate into large-model cost.

The router's own overhead is small: with the backbone forward pass included, ReLope adds under two percent to probe's per-example time and about a tenth of a gigabyte of peak memory (Appendix E.1). What grows is the training footprint, about thirty times more parameters than probe, and the next section asks whether the parameters, rather than the design, explain the gain.

Table 3: **Comparison in text-only and multimodal conditions.** Routing AUC (%) on A-OKVQA under the two evidence conditions.

Router	Original input	Caption only	Δ
Probe	81.94 ± 0.58	85.25 ± 0.52	$+3.31$
Attention Probe	84.06 ± 0.65	85.31 ± 0.47	$+1.25$
ReLope	86.24 ± 0.63	87.17 ± 0.82	$+0.93$

Table 4: **Routing scores under image perturbations.** AUC (%) for predicting clean-input correctness on A-OKVQA.

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: **Capacity and adaptation controls** with Qwen2.5-VL-7B-Instruct (AUC, %). LoRA + probe feeds the adapted last-token state to the classifier. VIB + probe applies the bottleneck to the frozen last-token state. Widened MLP enlarges the classifier to ReLope's parameter count.

Method	Trainable params	MMMU	A-OKVQA	ScienceQA
Probe	0.96M	76.61 ± 0.20	81.94 ± 0.58	83.65 ± 0.86
LoRA + Probe	3.84M	77.92 ± 0.51	83.17 ± 0.71	85.25 ± 0.73
VIB + Probe	26.66M	76.82 ± 0.87	83.05 ± 0.75	84.44 ± 1.05
Widened MLP	30.31M	78.65 ± 0.29	83.87 ± 0.21	86.13 ± 0.36
ReLope	29.54M	$\mathbf{80.79} \pm 0.40$	$\mathbf{86.24} \pm 0.63$	$\mathbf{91.78} \pm 0.52$

5.4 Where does the gain come from?

Learned pooling narrows the image–caption gap. To examine whether the fixed probe input limits correctness prediction, we compare the standard probe with Attention Probe while keeping the backbone frozen. Attention Probe improves AUC by 2.12 points with images but only 0.06 with captions, narrowing the gap from 3.31 to 1.25 points (Table 3). This suggests that how token hidden states are combined contributes substantially to the observed gap.

ReLope improves both conditions, with larger gains for images. Compared with the standard probe, ReLope increases AUC by 4.30 points with images and 1.92 points with captions, narrowing the image–caption gap to 0.93 points (Table 3). It also improves over Attention Probe in both conditions, suggesting benefits from adapting and regularizing the routing hidden states beyond learned pooling. The larger gain with images is consistent with our focus on multimodal routing.

Parameter count alone does not explain the gains. To test whether ReLope benefits mainly from additional parameters, we widen the classifier on frozen hidden states to a comparable parameter count. This control improves over the standard probe but still trails ReLope by 2.14–5.65 AUC points (Table 5). VIB + Probe also uses more parameters than LoRA + Probe while gaining less. These comparisons suggest that how the parameters are used matters beyond their number.

Adaptation and bottleneck together. We evaluate LoRA + Probe and VIB + Probe separately to examine the contribution of each component (Table 5). They improve AUC by 1.4 and 0.7 points on average over the standard probe, respectively. Combining them in ReLope outperforms both controls on all three datasets, with gains larger than the sum of the two individual gains. This pattern suggests complementary benefits from adapting and regularizing the routing hidden states.

Layer and readout on frozen states. We test whether a different layer or fixed readout would provide a stronger standard probe, comparing four readouts at seven layers with the same classifier (Table 6). The layer-24 last-token hidden state achieves the highest AUC, while means over text, image, or all tokens do not improve on it. Thus, our gains are measured against the strongest fixed configuration tested in this sweep. Appendix B.1 details the single-seed protocol.

KL regularization. To isolate the KL penalty, the $\beta = 0$ condition retains the stochastic Gaussian bottleneck but removes the KL term. Adding the penalty at $\beta = 1$ improves AUC by 2.16 (Table 7), which supports the contribution of KL regularization beyond the stochastic bottleneck alone.

LoRA rank. We vary the adaptation capacity while keeping α/r and the adapted projections fixed (Table 7). Rank 128 achieves the highest mean AUC, while increasing the rank to 512 uses four times as many LoRA parameters as rank 128 but yields lower mean AUC. Thus, greater adaptation capacity does not consistently improve correctness prediction.

Table 6: **The probe uses the best tested fixed hidden state.** A-OKVQA AUC (%), Qwen2.5-VL-7B-Instruct.

	Last	Mean over		
Layer	token	Text	Image	All
4	64.19	67.26	56.23	56.87
8	66.33	69.06	62.46	63.22
12	68.28	70.08	66.95	67.17
16	71.88	70.65	70.77	71.04
20	81.84	77.20	74.33	75.54
24	**82.03**	76.40	71.82	72.91
28	80.07	71.92	69.20	69.13

Table 7: **KL and LoRA rank sweeps.** AUC (%) on A-OKVQA (Qwen2.5-VL-7B-Instruct). Counts cover trainable LoRA parameters; bold marks the best mean.

KL coefficient		LoRA rank		
β	AUC (%)	r	LoRA params.	AUC (%)
0	84.08 ± 0.26	16	360,448	85.89 ± 0.27
10^{-3}	84.18 ± 0.61	64	1,441,792	85.37 ± 0.33
10^{-2}	85.17 ± 0.35	128	2,883,584	$\mathbf{86.24} \pm 0.63$
10^{-1}	86.06 ± 0.26	256	5,767,168	85.18 ± 0.48
1	$\mathbf{86.24} \pm 0.63$	512	11,534,336	85.05 ± 0.72

Table 8: **Cross-dataset transfer** with Qwen2.5-VL-7B-Instruct. ROC-AUC (%) on each held-out dataset (column) for routers trained on the other four datasets. Bold marks the best in each column.

Method	**MMMU**	**A-OKVQA**	**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

5.5 Robustness and transferability

Image perturbations. One motivation for the bottleneck is to limit reliance on input-specific information. We therefore test whether each router can still distinguish originally correct from incorrect answers when the image appearance changes. We apply Gaussian noise, JPEG compression, and Gaussian blur to A-OKVQA images while keeping clean-input correctness labels fixed. As shown in Table 4, the mean AUC drop across the three corruptions is 5.30 points for the probe, 2.69 for Attention Probe, and 1.44 for ReLope. The smaller drop suggests that ReLope more stably predicts clean-input correctness under image perturbations. This fixed-label test does not assess the correctness of answers generated from corrupted images.

Held-out datasets. To test transfer, we hold out each dataset in turn, train each learned router on the other four, and report its AUC on the held-out dataset in the corresponding column of Table 8. ReLope has the highest AUC on four of five targets, showing an advantage in cross-dataset transfer. MathVision is the exception: its visual-math reasoning may pose a larger domain shift, and training-free MoT outperforms ReLope (82.59 versus 66.83 AUC). The results suggest that the routing features learned by ReLope capture correctness signals that transfer across diverse multimodal domains, rather than relying primarily on dataset-specific patterns.

6 Conclusion

Hidden-state probes provide an answer correctness signal for routing with little cost, but they rely on hidden states produced by models trained for answer generation. We find that correctness is harder to predict with image inputs than with captions, and that Attention Probe substantially narrows this gap by learning to combine frozen token hidden states. Based on these findings, we propose ReLope, which learns hidden states for routing through a dedicated decision layer using a separate LoRA-adapted layer and a KL-regularized stochastic bottleneck, while preserving the answer-generation path. Across five benchmarks and three backbones, ReLope achieves the highest AUC among the evaluated methods in all 15 dataset–backbone pairs and improves hybrid accuracy–cost trade-offs with low inference overhead. These results support learning a dedicated decision layer within an MLLM to improve correctness prediction and routing accuracy–cost trade-offs.

AI USE STATEMENT

We used generative AI tools to help rephrase the manuscript and check the writing errors. We have not used generative AI tools to formulate claims or proofs, or to implement the methods. The authors verified the technical content and are responsible for it.

REFERENCES

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

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

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

Diogo Almeida. Introducing system one models & Jev. TypeSafe AI, 2026. URL `https://typesafe.ai/blog/introducing-system-one-models-and-jev`.

Amos Azaria and Tom Mitchell. The internal state of an LLM knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 967–976, 2023.

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

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022.

Qingqing Cao, Bhargavi Paranjape, and Hannaneh Hajishirzi. PuMer: Pruning and merging tokens for efficient vision language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12890–12903, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.721. URL `https://aclanthology.org/2023.acl-long.721/`.

Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. INSIDE: LLMs' internal states retain the power of hallucination detection. In *International Conference on Learning Representations*, 2024a.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. Are we on the right way for evaluating large vision-language models? In *Advances in Neural Information Processing Systems*, volume 37, 2024b.

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

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

C. K. Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, 16(1):41–46, 1970.

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

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

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. BEST-Route: Adaptive LLM routing with test-time optimal compute. *arXiv preprint arXiv:2506.22716*, 2025.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with Hilbert–Schmidt norms. In *International Conference on Algorithmic Learning Theory*, pp. 63–77. Springer, 2005.

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

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2733–2743, 2019.

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

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

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

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

Sanyam Kapoor, Nate Gruver, Manley Roberts, Katherine M. Collins, Arka Pal, Umang Bhatt, Adrian Weller, Samuel Dooley, Micah Goldblum, and Andrew Gordon Wilson. Large language models must be taught to know what they don't know. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=QzvWyggrYB.

Reza Khanmohammadi, Kundan Thind, and Mohammad M. Ghassemi. Calibrated triage, not autonomy: Confidence estimation for medical vision-language models. *arXiv preprint arXiv:2606.15910*, 2026. URL https://arxiv.org/abs/2606.15910.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pp. 3519–3529. PMLR, 2019.

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

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2521, 2022.

Haoxuan Ma, Guannan Lai, and Han-Jia Ye. MMR-Bench: A comprehensive benchmark for multimodal LLM routing. *arXiv preprint arXiv:2601.17814*, 2026.

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

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

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

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

Hadas Orgad, Michael Toker, Zorik Gekhman, Roi Reichart, Idan Szpektor, Hadas Kotek, and Yonatan Belinkov. LLMs know more than they show: On the intrinsic representation of LLM hallucinations. In *International Conference on Learning Representations*, 2025.

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

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

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

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

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

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

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

Weihao Xuan, Qingcheng Zeng, Heli Qi, Junjue Wang, and Naoto Yokoya. Seeing is believing, but how much? a comprehensive analysis of verbalized calibration in vision-language models. *arXiv preprint arXiv:2505.20236*, 2025. URL https://arxiv.org/abs/2505.20236.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. VisionZip: Longer is better but not necessary in vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19792–19802, June 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Yang_VisionZip_Longer_is_Better_but_Not_Necessary_in_Vision_Language_CVPR_2025_paper.html.

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

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

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. 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*, pp. 9556–9567, 2024.

Feiran Zhang, Yixin Wu, Zhenghua Wang, Xiaohua Wang, Changze Lv, Xuan-Jing Huang, and Xiaoqing Zheng. Vib-probe: Detecting and mitigating hallucinations in vision-language models via variational information bottleneck. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23509–23521, 2026.

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

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

Xiaohan Zou, Jian Kang, George Kesidis, and Lu Lin. Understanding and rectifying safety perception distortion in VLMs. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=KAMsbarp3w.

A Implementation Details

A.1 Training configuration

The current implementation uses PyTorch, Transformers, and PEFT. Probe-based routers use four Linear layers, $d \to 256 \to 128 \to 64 \to 1$, with ReLU activations. We train for three epochs with AdamW, learning rate 10^{-4}, and batch size 8. The official training pool is split by correctness label into 90% train and 10% development data; the checkpoint is selected only by development AUC and then evaluated once on the official validation split. The read layer is 24 for Qwen2.5-VL and Phi-4 and 40 for Gemma3. ReLope adapts the `q/k/v/o` projections of a separate copy of the read layer with LoRA rank 128, $\alpha = 256$, and dropout 0.05. Its KL coefficient is 1.0 with one-epoch linear annealing; KL is summed over latent dimensions and averaged over the batch. Training samples the bottleneck, while reported inference uses the posterior mean. Algorithm 1 lists the training steps.

A.2 Correctness labels

For MMMU, ScienceQA, and A-OKVQA, the model output is normalized and its extracted option is compared with the gold option. ChartQA uses relaxed exact matching for numerical answers and normalized string matching for text answers. MathVision uses final-answer extraction and symbolic-equivalence checking. No judge model is used.

A.3 Data splits and reproducibility

Each current run stores the model revision, command-line configuration, immutable example IDs and split hashes, per-epoch development predictions, selected checkpoint, and final evaluation predictions. For Qwen2.5-VL on A-OKVQA, the official training pool yields 15,350 training and 1,706 development examples; final evaluation uses all 1,145 examples in the official validation split. Principal comparisons use seeds 42, 43, and 44; deviations are sample standard deviations across seeds.

Features are read at the last valid prompt token. Probe and Attention Probe may reuse cached frozen states. ReLope keeps the answer-generating backbone frozen and evaluates only the separately adapted routing layer after the shared lower layers, so matched routers score the same small-model answers and correctness labels. Single-seed analyses (Tables 6 and 4) are labeled as such and are not mixed with the three-seed results.

A.4 Baseline inputs

RouteLLM and Post-Hoc Embed both receive image information. RouteLLM is implemented as a similarity-weighted ranking router whose query representation concatenates a CLIP text embedding of the question and its options with a CLIP embedding of the image. The matrix-factorization variant uses the same representation. Post-Hoc Embed averages the final decoder layer of the small model over all non-padding tokens, so it reads the full multimodal context including image tokens, and it pools the last hidden layer rather than layer 24. These two rows therefore differ from the probe-based rows of Table 2 in both input and pooling.

MMR-MLP and MMR-CMR are adapted to binary correctness prediction using frozen OpenCLIP ViT-B/32 embeddings.

B Readout and Hyperparameter Analysis

B.1 Layer and token readout

Table 6 compares four frozen-state readouts at layers 4, 8, 12, 16, 20, 24, and 28 of Qwen2.5-VL-7B-Instruct on all 1,145 A-OKVQA validation examples. Each MLP probe follows Appendix A and is trained with one seed. The layer-24 last-token hidden state gives the highest AUC among these configurations. Its AUC of 82.03 differs slightly from the three-seed mean of 81.94 in Table 2.

B.2 KL COEFFICIENT AND LORA RANK

We separately sweep the KL coefficient β and LoRA rank r on A-OKVQA with Qwen2.5-VL-7B-Instruct; Table 7 in the main text reports the results. The $\beta = 0$ condition retains the stochastic Gaussian bottleneck and removes only the KL penalty. In the rank sweep, we set $\alpha = 2r$ to keep α/r constant and adapt the same `q/k/v/o` projections. Parameter counts include only trainable LoRA modules.

C IMAGE–CAPTION INPUT COMPARISONS

C.1 COMPARISON PROTOCOL

Table 1 uses all 1,145 A-OKVQA validation examples and the pinned Qwen2.5-VL-7B-Instruct revision. The same example IDs, answer parser, prompt template, and layer-24 last-token readout are used in every condition. Captions were generated by the pinned Qwen2.5-VL-7B-Instruct model with the prompt *Describe this image in detail, focusing on visual facts needed to answer questions about it*. In the random-image condition the image is replaced by one drawn at random from another example of the dataset. Answers and correctness labels are recomputed per condition. Probe AUC follows the protocol of Table 2: the MLP probe of Appendix A is trained on the training pool of each condition, evaluated on the layer-24 last-token states of the validation examples, and reported as the mean and sample standard deviation over training seeds 42, 43, and 44. HSIC and label CKA are computed on the same states and labels. Caption input lowers answer accuracy by 10.05 points and raises probe AUC by 3.31 points, and HSIC and label CKA follow the same order. Because answers and labels are regenerated per condition, the comparison is descriptive rather than causal.

C.2 RESULTS ON MMMU AND SCIENCEQA

We repeat the three input conditions of Table 1 on MMMU and on the image subset of ScienceQA, keeping the pinned backbone, the caption prompt, the layer-24 last-token readout, and the per-condition retraining unchanged. Table 9 reports the result.

Table 9: **The comparison of Table 1 repeated on two further benchmarks.** Three input conditions on 900 MMMU and 2,097 ScienceQA validation examples with a frozen Qwen2.5-VL-7B-Instruct, under the protocol of Table 1. Probe AUC is the mean ± SD over three seeds. Bold marks the best value per column within each benchmark.

Input	Acc. (%)	Probe AUC	HSIC ($\times 10^{-3}$)	Label CKA
		MMMU		
Original input	**47.11**	76.61 ± 0.20	4.332	0.1106
Caption only	38.56	$\mathbf{78.02} \pm 0.48$	**4.969**	**0.1158**
Caption + random image	40.89	75.74 ± 0.64	3.193	0.0904
		ScienceQA		
Original input	**79.07**	83.65 ± 0.86	3.183	0.1242
Caption only	70.62	$\mathbf{88.42} \pm 0.03$	**8.063**	**0.2151**
Caption + random image	70.53	88.17 ± 0.27	7.188	0.2056

On both benchmarks, replacing images with captions lowers answer accuracy but raises probe AUC, HSIC, and label CKA, reproducing the A-OKVQA pattern. Adding an unrelated image has little effect on ScienceQA but reduces probe AUC and both dependence measures on MMMU. Thus, the claim that evidence source matters more than image-token presence is benchmark-dependent. Section 5.4 separately tests the role of the probe's readout.

C.3 ILLUSTRATIVE INPUT CONDITIONS

In Table 1, *Input* names the prompt construction: the question and answer options, together with any image or caption. *Evidence* identifies how information about the original scene is supplied, through

the image or its generated caption. *Image tokens* are visual tokens derived from an image supplied to the MLLM; caption words are text tokens.

Table 10: **Concrete input constructions for the birthday question.** The question and answer options are unchanged across conditions. The cupcake photograph is an illustrative unrelated image, not a recorded random draw from the experiment.

Input condition	Content supplied with the question and options	Evidence	Image tokens
Original input	The original birthday photograph, which shows the number 30	image	✓
Caption only	Its generated caption, which explicitly mentions 30; no image	text	–
Caption + random image	The same birthday caption plus an unrelated photograph, e.g., the cupcake photograph	text	✓

Figure 3 shows two concrete examples. For the cupcake question, the caption describes the topping as white frosting, supporting the gold answer, icing. For the birthday question, it explicitly states the number 30. Thus, the relevant clue is supplied visually in the original-input condition and textually in the caption-only condition. Adding an unrelated image introduces image tokens without restoring visual evidence from the original scene. These examples illustrate the input constructions, not the model's predictions or a guarantee that all captions preserve the required evidence.

D Hybrid-System Evaluation

D.1 Accuracy and cost at fixed escalation rates

We evaluate fixed escalation rates on all 1,145 A-OKVQA validation examples by aligning cached GPT-4.1 answers and token counts with each router's scores (Table 11). At each rate, the lowest-scoring examples are escalated; rates are specified in advance without test labels. Accuracy is reported as mean ± SD over seeds 42, 43, and 44, and API cost is averaged across seeds. Costs cover strong-model API usage only; local overhead is reported separately in Appendix E.1.

Table 11: **Hybrid accuracy and large-model cost** on the A-OKVQA validation set with Qwen2.5-VL-7B-Instruct and GPT-4.1, at escalation rates fixed in advance. Accuracy is mean ± SD over three router seeds. Cost saved is relative to strong-only inference. Prices follow the published `gpt-4.1-2025-04-14` rates with no prompt-caching discount. Bold marks the best hybrid accuracy per budget.

Router	Escalation	Hybrid acc.	GPT calls	API cost ($)	Cost saved
Small only	0%	78.78	0	0.000	100.00%
Strong only	100%	90.48	1,145	1.098	0.00%
Probe	20%	85.33 ± 0.10	229	0.214	80.54%
Probe	40%	88.68 ± 0.25	458	0.439	59.98%
Probe	60%	89.46 ± 0.22	687	0.654	40.45%
Attention Probe	20%	$\mathbf{85.59 \pm 0.46}$	229	0.220	80.00%
Attention Probe	40%	89.14 ± 0.13	458	0.437	60.20%
Attention Probe	60%	89.84 ± 0.13	687	0.656	40.24%
ReLope	20%	$\mathbf{85.59 \pm 0.13}$	229	0.213	80.57%
ReLope	40%	$\mathbf{89.32 \pm 0.20}$	458	0.437	60.16%
ReLope	60%	$\mathbf{89.87 \pm 0.26}$	687	0.654	40.45%

ReLope gives the highest mean accuracy at 40% and 60% escalation and ties Attention Probe at 20%. At 60% escalation, it reaches 89.87% accuracy versus 90.48% for strong-only inference while saving 40.45% of API cost, and at 40% it is within 1.2 points of strong-only accuracy at 60.16% cost saved. Appendix D.3 complements this budget-matched comparison with fixed-threshold routing.



(a) Cupcakes
ID: `2C8riXpRLX3CyM5jDz23m7`
Question: What is the white substance on top of the cupcakes?
Options: A. butter; B. mayo; C. ice cream; D. icing.
Gold answer: D. icing.

Generated caption (201 tokens):
The image shows the viewfinder of a camera displaying a close-up shot of three cupcakes being decorated. The cupcakes appear to be yellow cake with white frosting. Each cupcake is topped with a dollop of white frosting and a small red fruit, possibly a raspberry or strawberry, placed on top of the frosting. A hand is visible in the frame, holding what looks like a small tool or utensil, possibly used for placing the fruit toppings. The background within the camera's viewfinder includes a dark surface that appears to be a baking tray or a similar flat surface, and there is some powdered sugar scattered around. The camera's display also shows various details such as the battery level indicator, the number "100-0237" which likely refers to the file number, and the "CF" symbol indicating CompactFlash memory card usage. The camera model appears to be a Canon EOS 5D Mark II, as indicated by the branding at the bottom left corner of the viewfinder.

Answer clue: White frosting corresponds to icing.



(b) Birthday
ID: `2Aq5RiEn7eyfWjEbpuYT2o`
Question: Which number birthday is probably being celebrated?
Options: A. one; B. ten; C. nine; D. thirty.
Gold answer: D. thirty.

Generated caption:
The image shows a birthday cake placed on a wooden table. The cake is designed to look like a plush teddy bear, with a grayish-brown color and a soft texture that mimics the appearance of a stuffed toy. The teddy bear has a small blue bow tie around its neck and a tag attached to one of its paws, which reads "Happy Birthday."

To the right of the teddy bear-shaped cake, there is a smaller round cake with a purple base and a white top. The number "30" is prominently displayed on the top of this smaller cake, indicating that it might be part of a 30th birthday celebration. The smaller cake also has candles inserted into it.

The table is covered with a green tablecloth featuring a leafy pattern. In the background, there is a person sitting at the table, partially visible, wearing a red and black outfit. There are also some plates and glasses on the table, suggesting that the setting is a casual dining environment. The overall atmosphere appears festive and celebratory.

Answer clue: The caption explicitly identifies the number 30.

Figure 3: **Visual evidence and generated captions.** Two A-OKVQA examples with their images, questions, and full generated captions. Both captions retain the key clue needed for the corresponding question. Captions are reproduced verbatim. Gold answers are dataset annotations.

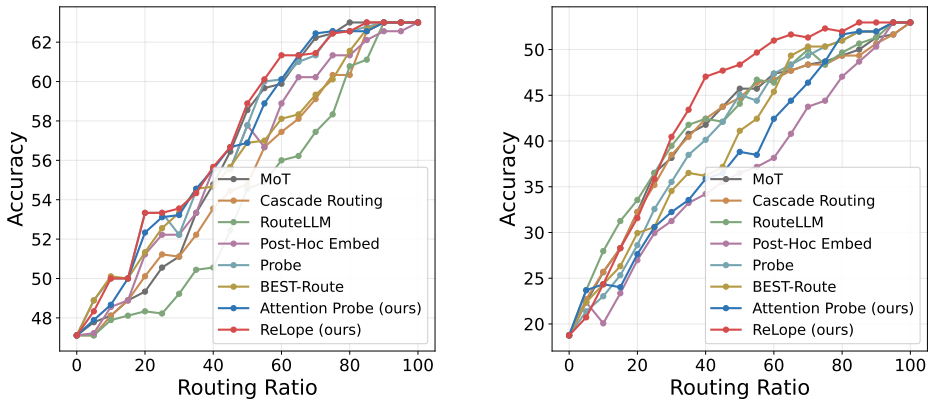
D.2 Additional routing curves on MMMU and MathVision

Both panels show the same pattern as Section 5.3. For a given budget of large-model calls ReLope returns at least as many correct system answers as the other routers over most of the range, and the curves cross in places because the large model can overturn a correct small-model answer.

D.3 Calibration and threshold-based routing

Fixed-threshold routing depends on score calibration as well as ranking. We evaluate ECE, Brier score, and escalation to GPT-4.1 when the routing score falls below h, following Appendix A with seeds 42, 43, and 44. ECE uses 15 equal-width bins. Table 12 reports hybrid accuracy at each threshold; because the same threshold escalates a different share of queries for each router, these accuracies are not budget-matched.

ReLope has the lowest ECE and Brier score on A-OKVQA; on MMMU, Attention Probe has the lowest ECE and ReLope the lowest Brier score. On MMMU, Attention Probe matches the large-only accuracy of 63.00 at $h = 0.7$. However, the same threshold can yield different escalation rates across routers, so these accuracies do not establish performance at a matched budget; Table 11 provides that comparison.



(a) MMMU (b) MathVision

Figure 4: **Hybrid-system accuracy against escalation rate** on the two benchmarks not shown in Figure 2, with Qwen2.5-VL-7B-Instruct as the small model. The large model is GPT-4.1 on both MMMU and MathVision. Axes and endpoints follow Figure 2.

Table 12: **Calibration and routing at a fixed score threshold.** A query is escalated when its routing score falls below h. Each cell gives the hybrid accuracy at that threshold, mean over three seeds; the share of queries escalated at a given h differs across routers.

Router	ECE	Brier	h=0.5	h=0.6	h=0.7	h=0.8	h=0.9
A-OKVQA (N = 1,145, small 78.78, large 90.48)							
Probe	0.0633	0.1257	82.47	85.15	86.90	87.98	89.34
Attention Probe	0.0480	0.1238	82.94	85.15	87.07	88.44	88.68
ReLope	**0.0409**	**0.1214**	84.13	85.62	87.42	89.61	89.75
MMMU (N = 900, small 47.11, large 63.00)							
Probe	0.0948	0.2166	53.59	55.96	58.63	61.56	62.85
Attention Probe	**0.0453**	0.2385	54.48	59.85	63.00	63.00	63.00
ReLope	0.0731	**0.2130**	54.85	58.30	60.78	62.59	63.00

E Computational Overhead

E.1 GPU inference overhead

We benchmark Qwen2.5-VL-7B-Instruct on ScienceQA on one NVIDIA H200 NVL (fp16), excluding preprocessing and host-to-device transfer but including CUDA synchronization. Head-only and backbone-plus-router timings average 200 and 30 repetitions, respectively. ReLope uses the posterior mean.

Table 13: Local routing cost at batch size 4 on an NVIDIA H200 NVL (fp16). Router time covers the scoring heads; total time and throughput include the backbone and full routing computation.

Router	Head ms/sample	Total ms/sample	Samples/s	Trainable params
Probe	0.055	17.383	57.53	0.96M
Attention Probe	0.047	17.372	57.56	0.96M
LoRA + Probe	0.054	17.455	57.29	3.84M
ReLope	0.065	17.695	56.51	29.54M

At batch size 4, ReLope adds 0.31 ms per example (1.79%) over probe, with peak memory of 16.35 versus 16.24 GiB. Differences remain small at batch sizes 1 and 8, but differing sequence lengths preclude a scaling comparison.

E.2 CPU ROUTING-BRANCH COST

To separate scoring-head cost from the full routing branch, we replay the adapted layer and scoring heads at batch size one on a memory-bandwidth-limited host CPU in float32, using the same inputs as in a forward pass (Table 14).

Table 14: **Routing cost at batch size one** on a host CPU (float32), separating the answer path, the routing branch, and the scoring heads.

Dataset	Image size	Seq. len.	Answer path (ms)	Branch (ms)	Heads (ms)	Branch+heads / answer	Heads / branch
ChartQA	native	244	7,160.84	324.85	6.106	4.62%	1.88%
ScienceQA	native	104	3,953.45	239.76	4.569	6.18%	1.91%
ScienceQA	644	584	20,318.90	584.05	7.372	2.91%	1.26%
ScienceQA	896	1,079	43,154.09	1,110.66	4.742	2.58%	0.43%
ScienceQA	1288	2,171	118,573.52	2,359.37	4.322	1.99%	0.18%

The branch plus heads cost 1.99–6.18% of the answer path, with a smaller share for longer ScienceQA prompts. Heads account for under 2% of branch time, distinguishing head-only timings from full-branch overhead.