ALLIANCE OF STOCHASTIC ROUTING AND REINFORCEMENT FUSION FOR MULTI-MODEL REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper studies dynamic model routing under uncertainty in multimodal settings, where a system must select and potentially compose multiple models to answer a query. Standard reinforcement learning approaches for routing often suffer from premature convergence to suboptimal models due to noisy rewards and insufficient exploration. We propose a unified vision-language **S**to**c**hastic r**out**ing framework (VL-Scout) for efficient multi-Modal and multi-step reasoning with turn-based reinforcement fusion with three unique features. First, VL-Scout combines Bayesian priors, policy gradient optimization, and sequential decision-making to enable structured exploration and adaptive inference. A context-dependent Dirichlet belief model allows for epistemic uncertainty over model quality and is updated online based on observed rewards and context-specific evidence, enabling Thompson-style exploration while serving as a prior for regularization. Second, a routing policy is trained via a KL-regularized REINFORCE objective by maximizing a reward-weighted evidence lower bound, where the policy acts as an amortized posterior, and the Dirichlet belief serves as a prior. Third, we introduce a fusion policy trained with Group Relative Policy Optimization (GRPO) or Reinforcement Learning with Verifiable Rewards (RLVR), which operates over intermediate model rationales and selects between two actions: CONTINUE, invoking another model via the router, or ANSWER, terminating with a generated output. This enables adaptive computation depth and selective model composition conditioned on reasoning quality. We evaluate VL-Scout on four representative multi-modal benchmarks (MMMU, MMMU-Pro, OKVQA, and RealWorldQA), compared against 11 representative routing and policy optimization baselines. Our method provides consistent performance improvements, while reducing premature policy collapse. These results highlight the importance of integrating uncertainty-aware priors with both amortized policy learning and sequential fusion for robust and efficient model selection in discrete action spaces. Code is available `anonymous.4open.science/r/vl-scout-8EB2/`

1 INTRODUCTION

Multi-modal reasoning is recognized as a challenging problem, as it requires models to understand, interpret, and process information from diverse sources. Advanced visual-language models (VLMs) have shown remarkable performance across different tasks, domains, and query types. However, no single model to date dominates uniformly across all evaluation settings and benchmarks.

Existing research tackles such fundamental heterogeneity in model competencies along two orthogonal directions: Model routing (Ong et al., 2024; Zhang et al., 2025; Li et al., 2025) aims to select the right model for each query, aiming for low cost and high accuracy. However, model routing fails to fully utilize the diversity in model capabilities, as shown by the recent progress in ensemble learning of LLMs/VLMs (Tekin et al., 2025; Wan et al., 2024; Yu et al., 2024; Huang et al., 2024; Mavromatis et al., 2024; Yao et al., 2024b; Jiang et al., 2023; Tekin et al., 2024) and collaboration (Sivakumaran et al., 2025; Liang et al., 2024; Du et al., 2023b). Little efforts have targeted at tackling both routing and fusion together. Recently, (Nguyen et al., 2024) modeled routing as a multi-armed bandit with contextual information and employed a linear relation. However, visual-language modality is too complex to learn through a linear relation. (Wang et al., 2025b) shows that a dynamic routing with a model pool can be maintained through continual learning. (Tekin et al., 2025) utilized reinforcement

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.

learning to select LLMs from a model pool and ensemble their logits. However, the introduced algorithm requires a warm-start dataset and a common value network to stabilize the agents (see Appendix for more detail).

In this paper, we propose VL-SCOUT, an online routing and fusion framework for vision-language models. It aims to combine model routing and model fusion by jointly optimizing both routing function and fusion function together in a dynamic changing environment where supervision is rarely available. We advocate that a practical routing system must be capable of learning effective routing policies online and adapt to new query distributions without relying on pre-collected task-specific training data. Leveraging this routing capability, an efficient ensemble fusion system should learn how to fuse conflicting outputs from multiple diverse base learners, regardless of which of the ensemble sets is chosen to solve a reasoning problem.

The design of VL-Scout has three salient features. First, we formulate routing as a multi-armed bandit problem and learn a non-linear mapping between model reward distributions and encoded visual-language context via a neural network, which is regularized by a Thompson-sampled prior from a category-conditioned Dirichlet belief on the same simplex and serves as global memory over the history of model performance across task categories. Second, we leverage train a routing policy via a KL-regularized REINFORCE objective by maximizing a reward-weighted evidence lower bound. Third, we propose a fusion model that operates on the routed model's rationales and outputs to adaptively decide whether to consult additional models or generate a final answer directly, enabling selective model composition conditioned on reasoning quality. Extensive evaluations across four benchmark datasets demonstrate that VL-Scout surpasses the best-performing base model, outperforms 11 representative VLMs by up to 5.36%, and exceeds the generalization performance of five recent ensemble and collaboration approaches by up to 3% at 8× lower inference cost.

2 PRIOR WORK

Ensemble Learning in Large Vision and Language Models Ensemble training methods and the distillation (Wan et al., 2024; Yu et al., 2024; Huang et al., 2024; Mavromatis et al., 2024; Yao et al., 2024b) and the mixture of experts (MoE) (Jiang et al., 2024) methods require significant computational resources and full access to the model parameters, making it challenging to generalize to diverse contexts and adapt to domain shifts. In the post-inference aggregation ensemble learning category, most LLM ensemble research centered on supervised solutions (Jiang et al., 2023; Tekin et al., 2024) and utilized majority voting to perform inference-time ensemble (Wang et al., 2022b; Fu et al., 2022; Li et al., 2022; Wang et al., 2022a). The downside of majority voting is the poor definition of equality between divergent answers. Two threads of research further improve majority voting, one utilizes the BLEU score as the heuristic to compare answers (Li et al., 2024), and the other enhances the BLEU score-based answer-combination method by assigning weights (Yao et al., 2024a) or creating a debate environment (Liang et al., 2023; Wan et al., 2024; Du et al., 2023a; Chan et al., 2023). LLM-based Multi-Agents (Guo et al., 2024) carry similar motivations in terms of exploiting multiple LLMs to work collaboratively for a particular task. However, neither the dynamic selection of agents nor the aggregation of outputs for a given task have been systematically explored. Similarly. LLM routing aims to identify the most suitable model among the pre-defined set of LLMs for a given prompt query. Unfortunately, routing is inherently limited by the performance of the chosen model and the dependency of using another external LLM to understand/rank the matching of a prompt query to the given pool of LLMs (Ong et al., 2024).

Reinforcement Learning in Model Routing and Ensemble Recently, RL approaches (Chakraborty et al., 2025; Fu et al., 2025) are proposed, which can dynamic adjust ensemble weights for an ensemble of size N (N is fixed for all tasks). Ensemble by RL holds the potential of creating an adaptive ensemble model (Song et al., 2023; Chua et al., 2018), ranging from time-series prediction (Liu et al., 2020; Németh & Szűcs, 2022; Perepu et al., 2020), ensemble pruning (Partalas et al., 2009; Liu & Ramamohanarao, 2020), to Tree-of-Thought (ToT) family of in-context learning of LLMs (Ouyang et al., 2022; Liu et al., 2024a; Monea et al., 2024; Zhang et al., 2021; Sun et al., 2024; Liu et al., 2024b). Yet, these existing RL approaches struggle to tackle the challenges when the ensemble of LLMs (base learners) offers very different inference performance with respect to diverse downstream reasoning tasks, given heterogeneous neural architectures finetuned with different LLM serving objectives as well as the challenges of whether it is feasible to dynamically compose an ensemble

from a pool of base learners on demand to better address the inference performance demand of each downstream user task.

3 PRELIMINARIES AND MOTIVATION

Routing Function. The routing is the subset problem of finding the best possible ensemble set combination for a given query. It selects a single model among candidate models. The objective is to learn a function $f_{\text{route}} : \mathcal{X} \to \{1, \ldots, K\}$ that maps a query x to the most suitable foundation model, e.g., LLM, VLM, in a model pool of size K to achieve the best response with the lowest cost. Following the definition (Nguyen et al., 2024), the routing functions maximize the performance on N queries with the given budget constraint b:

$$\arg\max_{f_{\text{route}}} \sum_{i=1}^{N} a_{f_{\text{route}}(x_i)}(x_i) \text{ s.t. } \sum_{i=1}^{N} c_{f_{\text{route}}(x_i)} \leq b. \tag{1}$$

More precisely, by defining $\mathbf{S} \in \{0,1\}^{N \times K}$ and $\mathbf{A} \in \mathbb{R}^{N \times K}$, where $\mathbf{S}_{i,j} = \mathbb{1}_{\{f_{\text{route}}(x_i)=j\}}$ is representing the choice of routing function f_{route} on N queries and $\mathbf{A}_{i,j} = a_j(x_i)$ is the performance of model j on sample x_i. The optimization goal is to find a choice matrix that maximizes the overall performance:

$$\arg\max_{\mathbf{S}} \sum_{i=1}^{N} \sum_{j=1}^{K} \mathbf{A}_{i,j}\mathbf{S}_{i,j} \text{ s.t. } \sum_{i=1}^{N} c_{f_{\text{route}}(x_i)} \leq b. \tag{2}$$

The optimal $\mathbf{S}$ can be reached when the performance (e.g. accuracy) matrix is known (Chen et al., 2022; 2023). Yet, we do not know the accuracy of a model on a test sample before feeding the sample to the model. Even if we know, *the performance feedback may be delayed, and a model's true performance may only be revealed after multiple queries.* Previous works on LLM-Routing try to overcome the problem by learning a router-model which predicts the performance of each model in the pool. However, this requires the router to comprehend the intricate details of the query and each model's capabilities. The vision modality exacerbates the complexity of untangling these relations. Jointly optimizing these two aspects may limit exploration and adaptive inference, resulting in premature convergence to suboptimal models. Therefore, router models generally require huge parameter sizes or high computational cost due to their cascading query protocols. In this work, we formalize both the routing and ensemble problems within a reinforcement learning framework, enabling the learning of more generalizable, *meta-level* strategies.

Problem Formulation. Let $(x, \mathbf{I})$ denote an input tuple of a language query and an image, where the query $x = \{\omega_1, \ldots, \omega_L\}$ is composed of L tokens, and the image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$ represents the RGB input. Let $(x_1, \mathbf{I}_1), \ldots, (x_n, \mathbf{I}_n)$ denote a sequence of input queries with length n, where each query $(x_i, \mathbf{I}_i)$ is sampled from a task $\mathcal{T}_j$, denoted by $(x_i, \mathbf{I}) \sim \mathcal{T}_j$. The queries can originate from a single task or from a group of tasks $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_J$, e.g., spatial-visual reasoning, science-math, and social-applied with varying difficulties. Let $\mathcal{E}_t = \{\mathcal{M}_1, \ldots, \mathcal{M}_N\}$ denote a set of VLMs that can be used to obtain the desired output y at time t. For each query x_i, the routing aims to minimize $\mathcal{L}\left(\mathcal{M}_{f_{\text{route}}(x_i)}(x), y\right)$ where $\mathcal{M}_{f_{\text{route}}(x_i)}$ is the routed model and $\mathcal{L}$ a generic loss function such as L2. The ensemble fusion function objectifies $\mathcal{L}\left(\hat{y}_{\text{fusion}}, y\right)$, where $\hat{y}_{\text{fusion}} = f_{\text{ens}}(\mathcal{M}_1(x_i), \ldots, \mathcal{M}_K(x_i))$. The design of VL-Scout aims to route and fuse jointly together. On one hand, we want to select the best models for each incoming query with both textual and visual (image) modality; and on the other hand, we want to fuse the selected model outputs with query-aware adaptability. Therefore, for each query, the common objective is formulated as follows:

$$\min_{\theta,\phi} \mathcal{L}\left(y, f_{\text{ens}}\left(\{\mathcal{M}_k(x_i) : k \sim \hat{f}_{\text{route}}(x_i;\theta)\}, \phi\right)\right) \text{ s.t } \sum_{i=1}^{N} c_{f_{\text{route}}(x_i)} \leq b. \tag{3}$$

Here, $\hat{f}_{\text{route}} : \mathcal{X} \to [0,1]^K$ is the routing function that maps the input query to a probability distribution over K models. In other words, here we sample a possible ensemble set from a universal set $\mathbb{U}$ containing all the possible ensemble combinations with $|\mathbb{U}| = 2^K - K - 1$. Our objective is to minimize the loss by finding the parameters θ and ϕ that parametrize the routing and fusion function. Moreover, the optimization has a budget constraint and the feedback signal y can be delayed or noisy. With these elements, the optimization problem distinguishes itself from the literature by its difficulty, modality, and applicability.

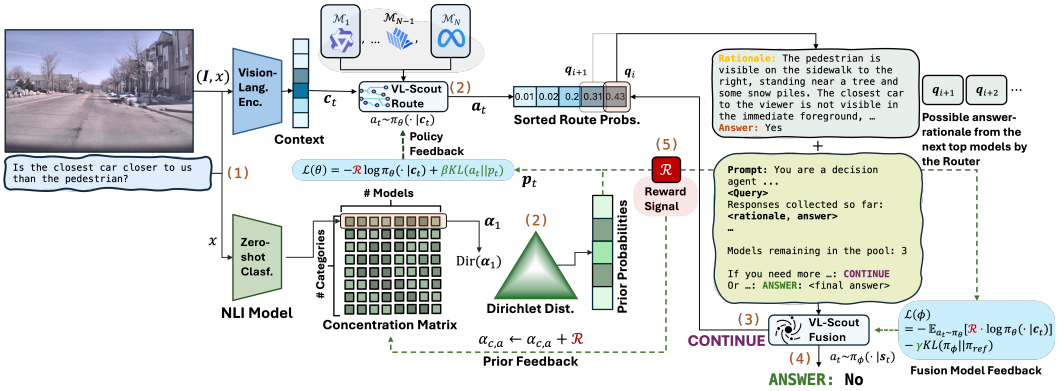


Figure 1: Overview of *VL-Scout* two stage ensemble by reinforcement learning agents. The step sequence number is used to annotate the order of the workflow steps of VL-Scout.

4 VL-SCOUT

We first give an architectural overview of VL-Scout in Figure 1, with a highlight of five steps: (1) The input image and query are passed to the Vision-Language Encoder to obtain its contextual embedding and Natural Language Inference model for zero-shot classification of one of the categories. (2) We obtain probabilities for the most suitable models in the pool based on the context from VL-Scout Route Agent and Prior Distribution. (3) The top selected model's response and rationale is prompted to VL-Scout Fusion Model to generate either two actions CONTINUE, invoking another model via the router, or (4) ANSWER, terminating by generating a final answer. At the final step (5), the reward is calculated based on the final answer and is fed back to the policies and posterior update for new prior. This process iterates until all episodes are finished. In Appendix-B, we introduce four routing techniques that serve as the foundational components of our proposed algorithm, VL-Scout Route. Presenting these methods provides the necessary context for understanding the design and behavior of our approach.

4.1 VL-SCOUT ROUTE: REINFORCE WITH PRIOR REGULARIZATION

The Multi-Armed Bandit routers (see Appendix B) each solve only part of the problem: Thompson Sampling ignores the query context, LinUCB and its Thompson variant rely on a linear reward model that breaks down under domain shift, and REINFORCE captures non-linear context but cannot use prior knowledge of which models suit which query categories and can converge prematurely to suboptimal models under noisy or delayed rewards, so an effective router must be context-aware, non-linear, prior-guided, and exploratory at the same time.

Context and category. We assume a predefined set of categories $\mathcal{C}$ to which a query $x \in \mathcal{X}$ may belong (see the list of categories in Appendix-E). For each category $c \in \mathcal{C}$, we maintain a prior belief over the performance of K models, represented as a probability distribution over arms, $p(a|c(x))$. We map each input query to a category using a function $c : \mathcal{X} \to \mathcal{C}$, instantiated via a Natural Language Inference (NLI) model that assigns the query to the most relevant category with zero-shot classification (see step-1 in Figure 1). Additionally, the router observes $\mathbf{h}_t = g(x_t, \mathbf{I}_t) \in \mathbb{R}^{768}$, the mean-pooled Q-Former output of frozen BLIP-2 encoder (Li et al., 2023).

Policy. The router is a categorical policy over the K models, $\pi_\theta(k \mid \mathbf{h}_t) = \mathrm{softmax}(f_\theta(\mathbf{h}_t))_k$, where f_θ is a two-layer MLP. A top-1 routing decision samples $k_t \sim \pi_\theta(\cdot \mid \mathbf{h}_t)$; a shortlist of length n appends the $n-1$ remaining models in decreasing order of π_θ.

Belief. For each category c, VL-Scout maintains a Dirichlet belief $\mathrm{Dir}(\boldsymbol{\alpha}_c)$, $\boldsymbol{\alpha}_c \in \mathbb{R}^K_{>0}$, over which model to prefer for that category and $\tilde{\mathbf{p}}_t \sim \mathrm{Dir}(\boldsymbol{\alpha}_c)$ is the sampled belief. The entries of $\boldsymbol{\alpha}_c$ act as pseudo-counts: the expected preference for model k is $\alpha_{c,k}/\sum_j \alpha_{c,j}$, and the total $\sum_j \alpha_{c,j}$ sets how firmly the belief is held—the larger it is, the closer the samples drawn from the belief stay to this mean. Giving every model the same initial count α_0 yields an uninformative belief. An operator who expects some models to be strong, e.g., from public leaderboard results, can designate them as a set

$\mathcal{P}$ and give them a larger initial count $\alpha_{\mathcal{P}} > \alpha_0$; this is how prior knowledge about the pool enters the system. The prior is an adaptive head start of pseudo-counts: after each query, every consulted model k contributes signed evidence

$$\alpha_{c_t,k} \leftarrow \alpha_{c_t,k} + e_{t,k}, \qquad e_{t,k} = \begin{cases} 1 & \text{if } \hat{y}_{t,k} = y_t, \\ -\lambda_f & \text{otherwise,} \end{cases} \tag{4}$$

and concentrations that become non-positive are reset to $\alpha_{\min} > 0$. For instance, a non-designated model that answers ten queries of a category correctly reaches $\alpha_{c,k} = 11$ and overtakes a designated model that has gained no evidence in that category. Without the failure term ($\lambda_f = 0$), Eq. (4) is the conjugate update of a categorical observation model whose observations are the identities of models that answered correctly; the failure term additionally removes mass from models that are consulted and wrong. The belief is therefore a pseudo-count *preference* over the models of a category, not an exact posterior over each model's success rate. Because the counts are not normalized by how often a model is consulted, the belief favors models that are both often consulted and often correct; exploration comes from sampling it.

Objective. Our goal is to learn a query-dependent routing policy that selects models with high expected reward while remaining robust to noisy and sparse feedback. In particular, we want to avoid two undesirable behaviors: (i) prematurely collapsing onto a model that happens to perform well on a few early queries, and (ii) failing to explore models whose quality is still uncertain. We address these issues by combining a Bayesian belief over model quality with KL-regularized policy optimization:

$$J_t(\theta) = \mathbb{E}_{k\sim\pi_\theta(\cdot|\mathbf{h}_t)}\left[r_t(k)\right] - \beta\mathcal{D}_{KL}(\pi_\theta(\cdot \mid \mathbf{h}_t)||\tilde{\boldsymbol{p}}_t), \tag{5}$$

where $r_t(k) = 1$ if the answer obtained from the routing decision is correct and $r_t(k) = -0.1$ otherwise. Both $\pi_\theta(\cdot \mid \mathbf{h}_t)$ and $\tilde{\mathbf{p}}_t$ are categorical distributions on the simplex Δ^{K-1}, so the KL term is well defined and computed exactly. Equation (5) is the Lagrangian relaxation of maximizing the expected reward subject to $\mathrm{KL}(\pi_\theta\|\tilde{\mathbf{p}}_t) \le \epsilon$ (Boyd & Vandenberghe, 2004), with a fixed multiplier β. Thus, the router seeks high reward while remaining within a controlled distance of the sampled belief. In the following preposition, we show that the posterior is reward-weighted version of the belief by following standard identity (Levine, 2018) which makes precise in what sense the trained router is an amortized posterior.

Proposition 1 (Gibbs posterior) *Let $\tilde{\mathbf{p}}$ lie in the interior of Δ^{K-1}, $r : \{1,\dots,K\} \to \mathbb{R}$, $\beta > 0$, and $\pi^\star(k) = \tilde{p}_k\, e^{r(k)/\beta}/Z$, where the normalizing constant $Z = \sum_j \tilde{p}_j\, e^{r(j)/\beta}$ ensures $\sum_k \pi^\star(k) = 1$. For every $\pi \in \Delta^{K-1}$,*

$$\mathbb{E}_{k\sim\pi}[r(k)] - \beta\,\mathrm{KL}(\pi\|\tilde{\mathbf{p}}) = \beta\log Z - \beta\,\mathrm{KL}(\pi\|\pi^\star),$$

so $\pi^\star$ is the unique maximizer. Moreover, for a binary success variable O with likelihood $P(O{=}1 \mid k) \propto e^{r(k)/\beta}$ and prior $\tilde{\mathbf{p}}$ over models, the objective divided by β equals, up to an additive constant, the evidence lower bound on $\log P(O{=}1)$ with variational distribution π, and $\pi^\star$ is the exact posterior $P(k \mid O{=}1)$.

The proof of the preposition is in Appendix-C. The trained policy maps $\mathbf{h}_t$ directly to the distribution and amortizes the per-query posterior $\pi_t^* \propto \tilde{\mathbf{p}}_t\, e^{r_t/\beta}$ across queries. The belief proposes, the reward reweights, and the policy generalizes the result through context embedding. Since calculating posterior requires observation of r_t for all the models, we estimate the gradient of the first term in 5 with REINFORCE (Williams, 1992) and differentiate the KL term exactly:

$$\nabla_\theta J_t = \mathbb{E}_{k\sim\pi_\theta}\left[r_t(k)\,\nabla_\theta \log \pi_\theta(k \mid \mathbf{h}_t)\right] - \beta\sum_{k=1}^{K} \nabla_\theta\pi_\theta(k \mid \mathbf{h}_t)\,\log\frac{\pi_\theta(k \mid \mathbf{h}_t)}{\tilde{p}_{t,k}}. \tag{6}$$

4.2 VL-SCOUT FUSION

Routing struggles with the early stopping by picking one model and stopping. The strength of combining multiple model outputs and their relations is never discovered. The routed model may make an error that is irreversible. It is possible that consulting another model yields a correct solution or a different perspective by leveraging complementary strengths of multiple models and

their rationales. Use of multiple models, however, creates two additional questions: (i) how to create an ensemble set and (ii) how to combine the models' output with a generalizable solution described in Section 3. VL-Scout Fusion introduces a fusion model which is parameterized by a pre-trained VLM that learns to defer decision and refine predictions by outputting CONTINUE, defer decision and gather more evidence, and ANSWER, stop and generate new refined predictions. Due to the autoregressive nature, the VL-Scout can fuse the answers given to the open-ended questions. As shown in Figure 1, the fusion model exploits the rationales of the candidate models, sorted by their routing probabilities. Given the prompt, the fusion model follows a sequential decision process to determine when to stop or ask for more information. The fusion model thus turns static routing into adaptive, multi-step decision-making with learned stopping. The objective can be formalized as an MDP with the following elements. **State**: $s_t = (x, h_t)$, where x is the input query with its image and h_t is the history up to step t, consisting of the selected models, their outputs, and rationales. **Action**: $a_t \in \{\text{CONTINUE}, \text{ANSWER}\}$. **Transition**:

$$s_{t+1} = \begin{cases} \big(x,\ h_t \cup \mathcal{M}_j(x)\big) & \text{if } a_t = \text{CONTINUE}, \\ \hat{y}_{\text{fusion}} & \text{if } a_t = \text{ANSWER}, \end{cases} \tag{7}$$

where $\mathcal{M}_j$ is the next routed model. **Reward**: the reward can be binary feedback testing whether $\hat{y}_{\text{fusion}} = y$, or it can be measured by a reward model. To incentivize consulting more models and to respect the budget constraint in Eq. (3), we add the following penalties:

$$r_t = \begin{cases} -\dfrac{n}{K}\,\kappa & \text{if } a_t = \text{ANSWER and } \hat{y}_{\text{fusion}} \neq y, \\ r' & \text{if } n > n', \end{cases} \tag{8}$$

where n is the number of models queried, $\kappa \in [0, 1]$ is the early-stop penalty, and r' is a large negative reward for exceeding the budget of n' models; we set $n' = 3$ and $r' = -5$. We use model count for coarse budget control; token-level costs enable finer-grained rewards. The **objective** is to maximise $\mathbb{E}_{\tau\sim\pi_\phi}[\mathcal{R}(\tau)]$ with trajectory $\tau = (s_1, a_1, \ldots, s_T, a_T)$. We adapt two post-training techniques for VL-Scout Fusion.

Reinforcement Learning with Verifiable Rewards (RLVR) (Wen et al., 2025) defines reward via a verifier function: $\mathcal{R}(\tau) = V(\hat{y}_{\text{fusion}}, y)$, where $V \in [0, 1]$ is the correctness/accuracy function. Then, the model is trained via policy gradient $\nabla_\phi J = \mathbb{E}\left[\mathcal{R}(a, x)\nabla_\phi \log \pi_\phi\right]$. RLVR offers a flexible, modular, and simple reward design and directly optimizes the final task objective. However, sparse rewards such as binary may cause a high-variance effecting early actions to get weak signal.

Group Relative Policy Optimization (GRPO) (Shao et al., 2024) reduces the variance by using relative rewards within a group of trajectories. For each query x the policy samples G trajectories $\tau_1, \ldots, \tau_G \sim \pi_\phi$ and calculate their rewards. Then the model is trained via GRPO loss function:

$$\mathcal{L}_{\text{GRPO}}(\phi) = \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|\tau_i|}\sum_{t=1}^{|\tau_i|}\Big\{\min\Big(\hat{r}_t(\phi)\hat{A}_{i,t}, \text{clip}\big(\hat{r}_t(\phi), 1-\epsilon, 1+\epsilon\big)\hat{A}_{i,t}\Big) - \beta\mathcal{D}_{\text{KL}}(\pi_\phi||\pi_{\text{ref}})\Big\}, \tag{9}$$

where $\hat{r}_t(\theta) = \frac{\pi_{\theta_{\text{new}}}(a_t|\mathbf{s}_t)}{\pi_{\theta_{\text{old}}}(a_t|\mathbf{s}_t)}$ is the ratio term, $\hat{A}_t = \mathcal{R}_t - \bar{R}$ is the estimated advantage function to reduce variance and stabilize learning under sparse rewards. GRPO with the ratio term and reference model penalty encourages structured exploration, best when the reward function is relative and noisy.

5 Experimental Evaluations

We evaluate VL-Scout across four benchmarks in Table 1 — MMMU (Yue et al., 2024a), MMMU-Pro (Yue et al., 2024b), A-OKVQA (Schwenk et al., 2022), and RealWorldQA (xAI, 2024) — comparing against a broad set of pretrained vision-language models and assessing performance under two practical deployment scenarios: online routing and fusion without training data, and online routing and fusion with cross-domain training data available.

Comparison with Pretrained Baselines. Among the 11 pre-trained baselines, InternVL3-8B outperforms the other 10 VLMs on three of the four benchmarks with 57.76 on MMMU, 51.82 on MMMU-Pro and 88.47 on OKVQA. However, on RealWorldQA, InternVL3-8B (63.53) falls behine Qwen3-VL-8B (71.11) and InternVL2-8B (70.10). In comparison, Without access to any training

Table 1: VL-Scout selection and ensemble composition for the evaluated VLM benchmarks.

Settings	VLMS	MMMU (Acc %)↑	MMMU-Pro (Acc %)↑	A-OKVQA (Acc %)↑	RealWorldQA (Acc %)↑
Pretrained	llava-v1.6-vicuna-7b	36.27	35.42	82.62	38.30
	llava-v1.6-vicuna-13b	36.77	33.06	83.41	48.37
	Llama-3.2-11B-Vision	39.00	38.94	81.48	33.86
	deepseek-vl2-tiny	39.75	37.96	80.09	64.2*
	deepseek-vl2-small	39.01	34.63	76.68	65.4*
	Qwen2.5-VL-7B	51.55	46.95	87.25	69.15
	Qwen3-VL-8B	56.64	50.56	87.94	71.11
	Kimi-K2.5	30.18	24.87	80.96	64.97
	gemma-3n-E4B	48.57	43.03	83.06	56.47
	InternVL2-8B	51.30	43.12	85.33	70.10*
	InternVL3-8B	57.76	51.82	88.47	63.53
Online w/o Training Data	**Routing Only (Top-1)**				
	TS-MA	55.40	48.36	86.38	62.75
	LinUCB	48.08	42.39	84.62	59.73
	TS+LinUCB	49.32	44.78	88.21	57.51
	REINFORCE	52.91	48.05	85.15	67.45
	VL-Scout Route (REINFORCE+Prior)	56.65	50.13	86.72	68.15
	Routing + Fusion				
	Voting+Bernoulli	54.04	50.34	85.76	67.45
	Voting-Top-3	57.64	51.38	89.78	71.24
	VL-Scout Fusion (RLVR)	52.67	46.76	89.00	72.55
	VL-Scout Fusion (GRPO)	**60.37**	**52.01**	**90.13**	**76.47**
Online w/ Cross-Domain Training Data	**Routing Only(Top-1)**	MMMU-pro → MMMU	MMMU →MMMU-pro	RealWorldQA →OKVQA	A-OKVQA →RealWorldQA
	TS-MA	55.90	50.00	87.77	69.02
	LinUCB	41.20	41.27	74.41	56.99
	TS+LinUCB	35.62	36.80	72.31	58.82
	REINFORCE	48.57	50.65	86.37	63.52
	VL-Scout Route (REINFORCE+Prior)	55.78	49.82	86.20	71.11
	Routing + Fusion				
	Voting+Bernoulli	54.65	51.63	87.42	68.11
	Voting-Top-3	57.28	50.31	88.04	72.42
	VL-Scout Fusion (RLVR)	56.65	49.37	89.00	72.29
	VL-Scout Fusion (GRPO)	**62.86**	**53.52**	**90.57**	**74.51**
	Oracle	81.98	78.14	95.89	92.54

data, VL-Scout (GRPO) outperforms all 11 VLMs in the pool on all four benchmarks, with 60.37 on MMMU, 52.01 on MMMU-Pro, 90.13 on OKVQA, and 76.47 on RealWorldQA, surpassing by a clear margin on MMMU and OKVQA. This result underscores a central thesis of our work: principled routing and fusion of pre-trained VLMs can be a strong alternative to scale individual model's capacity and yield consistent gains without any additional model training.

Routing-Only Strategies: The Case for REINFORCE+Prior. For the family of routing-only (Top-1) methods, performance results vary considerably across bandit-based and policy gradient approaches, revealing important challenges in how well each strategy adapts to the online routing problem. Standard bandit methods, e.g., LinUCB, MA-TS, achieve moderate performance with 48.08 and 55.40 on MMMU, respectively, but struggle to capture the nuanced, query-dependent preferences that effective routing demands. Policy gradient methods are more effective in this regard. VL-Scout (REINFORCE) improves upon the bandit baselines by a meaningful margin, while VL-Scout (REINFORCE+Prior) is the strongest routing-only method by a consistent margin across all 4 benchmarks, achieving 56.65 on MMMU, 50.129 on MMMU-Pro, 86.72 on OKVQA, and 68.15 on RealWorldQA. The incorporation of a prior over model preferences provides a crucial inductive bias that stabilizes early-stage routing decisions and accelerates convergence, particularly in low-data regimes where pure exploration is costly. These results establish REINFORCE+Prior as a competitive solution in settings where fusion is computationally prohibitive or architecturally constrained.

Routing + Fusion: VL-Scout (GRPO) Sets a New Standard. The full potential of VL-Scout is realized through its fusion-based variants, which aggregate predictions across multiple models rather than committing to a single selection. Across all fusion strategies, VL-Scout (GRPO) achieves the state-of-the-art results in online without training data setting, with 60.37 on MMMU, 52.01 on MMMU-Pro, 90.13 on OKVQA, and 76.47 on RealWorldQA, outperforming all routing-only methods and all pretrained VLM baselines by substantial margins. Compared to Voting-Top-3, VL-Scout (GRPO) improves MMMU by 2.73 points and RealWorldQA by 5.23 points, showing that naive ensemble voting is suboptimal and that learning to combine model outputs via group relative

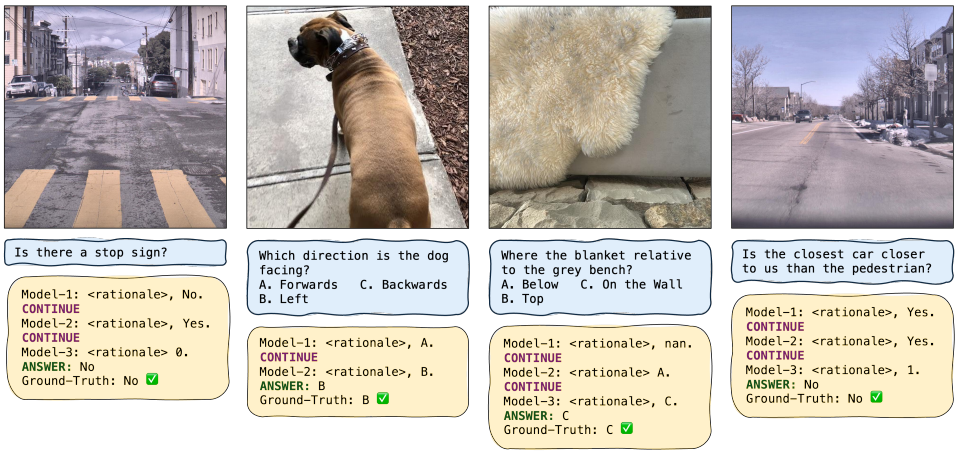


Figure 2: Example illustration of VL-Scout reasoning. Example-1 reaches the correct answer after evaluating the first model's output. Example-2 makes the correct decision after consulting the second model. Example-3 continues consulting models until it finds a correct rationale and answer. Example-4 makes correct decision when all consulted models are incorrect. Detailed responses in Appendix-G.

Table 2: VL-Scout against routing, collaboration, and ensemble baselines.

Method	Routing	Fusion	Infer. Time (s)	MMMU Acc (%)	A-OKVQA Acc (%)
LLM-Blender (Pair R.)(Jiang et al., 2023)	✓	✗	19.1s	22.36	70.56
Majority Voting	✗	✓	0.001	41.98	85.76
Plurality Voting	✗	✓	0.001	52.54	86.81
RouteLLM (Random)(Ong et al., 2024)	✓	✗	0.0007	26.45	68.90
MAD w/ Consensus(Sivakumaran et al., 2025)	✗	✓	21.84	53.2	57.4
MAD w/ Judge(Sivakumaran et al., 2025)	✗	✓	27.92	54.8	59.7
DART (Best Model) (Sivakumaran et al., 2025)	✓	✓	32.75	61.6	65.3
VL-Rethinker-7B (Wang et al., 2025a)	✗	✗	0.16	56.7	87.28
VL-Scout	✓	✓	4.09s	**62.86**	**90.13**

policy optimization yields meaningfully stronger aggregations. Compared to RLVR on RealWorldQA, the gains of GPRO are more pronounced (76.47 vs. 72.55), as GRPO's optimization objective better captures the distributional shift between benchmark domains.

Cross-Domain Training Further Amplifies VL-Scout's Advantages. Incorporating cross-domain training data provides additional, consistent improvements across all VL-Scout variants, and VL-Scout (GRPO) leads all methods compared in Table 1 by a decisive margin, achieving 62.86 on MMMU, 53.52 on MMMU-Pro, 90.57 on OKVQA, and 74.51 on RealWorldQA. The improvement from the no-training setting is consistent across benchmarks, particularly notable on MMMU (+2.49 points) and MMMU-Pro (+1.51 points). It indicates that cross-domain routing signals transfer effectively via Prior and VL-Scout is capable of exploiting auxiliary supervision to refine its model selection strategy. Also GRPO consistently outperforms RLVR, voting-based methods, REINFORCE+Prior, and bandit methods.

Gap to Oracle. Despite these strong results, a non-trivial gap remains between VL-Scout (GRPO) and the Oracle upper bound, which assumes perfect per-instance model selection and achieves 81.98 on MMMU, 78.14 on MMMU-Pro, 95.89 on OKVQA, and 92.54 on RealWorldQA. Even though there is a relatively high gap in MMMU and MMMU-Pro, VL-Scout achieves up to 5% difference in OKVQA. Figure 2 shows examples of how VL-Scout operates.

Performance against Prior Work. Table 2 benchmarks VL-Scout against a broad spectrum of multi-model inference strategies, and the results consistently favor VL-Scout across both accuracy and efficiency dimensions. Naive ensemble methods such as Majority Voting (41.98%) and Plurality Voting (52.54%) fall well short on MMMU despite their near-instantaneous inference times, demonstrating that undiscriminating aggregation fails to exploit the complementary strengths of heterogeneous VLMs. Multi-agent collaboration approaches close some of this gap, MAD w/ Judge and DART (Best Model) reach 54.8% and 61.6% on MMMU, respectively, yet with a prohibitive computational cost, requiring up to 32.75 seconds per query, nearly 8× the inference time of VL-Scout. Against VL-Rethinker-7B, VL-Scout delivers gains of 6.16 points on MMMU and 2.85 points

on A-OKVQA. Ultimately, VL-Scout achieves state-of-the-art results on both benchmarks, 62.86% on MMMU and 90.13% on A-OKVQA, at just 4.09 seconds per query.

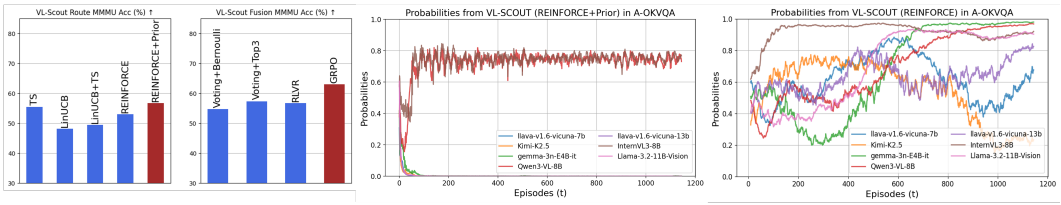


Figure 3: The left shows an ablation study on routing and fusion in the MMMU dataset. The last two plots show how the prior affects the convergence on the stochastic model selection.

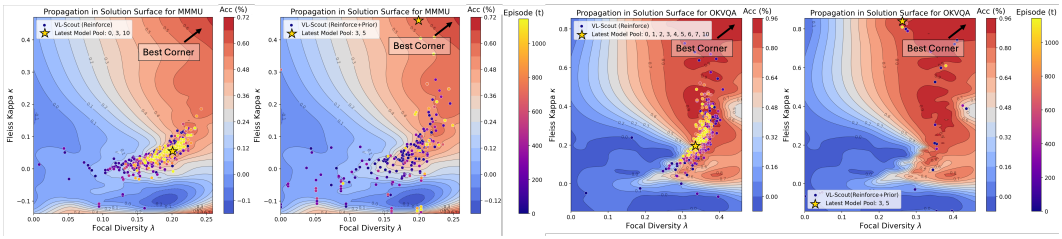


Figure 4: All candidate ensemble teams from the model pool of size $K = 11$ are plotted with their focal diversity scores(Tekin et al., 2024), Fleiss Kappa, and Plurality Voting Accuracy using two popular datasets. We use cubic interpolation with RBF smoothing to create a surface, and the dark red represents a higher performance score. Each dot represents the explored ensemble team by the routing models REINFORCE and REINFORCE + PRIOR, colored by their appearances during testing.

Ablation Study. The bar chart in Figure 3 reports the results of experiments on the effect of Thompson sampling, linear context modeling, and non-linear context modeling of VL-Scout Route on the MMMU dataset. VL-Scout reaches its best performance when non-linear modeling of the context information is combined with Bayesian priors. This effect is also visible on the probability distribution change shown in the curve plots. The prior regularizes the model selection distribution, whereas vanilla REINFORCE exhibits abrupt fluctuations. The second bar chart illustrates the impact of fusion modeling on performance, comparing Bernoulli sampling with voting, top-3 selection with voting, RLVR-based selection and fusion, and GRPO-trained fusion and selection. Notably, VL-Scout Fusion with GRPO achieves the best performance. These results highlight the critical role of principled exploration and fusion strategies, demonstrating that combining non-linear context modeling with Bayesian priors and GRPO-based fusion yields the most stable and effective performance gains. Figure 4 shows the effect of Prior to the REINFORCE, where the vanilla model cycles at the mediocre solution sets, Prior initialized REINFORCE algorithm quickly converges to the best model set selection. Without the prior, REINFORCE's sampled teams stay in a narrow band through the middle of the surface and end at mediocre teams on both benchmarks. With the Dirichlet prior, the router still explores widely in early episodes but then moves into the best corner and ends at the compact pair $3, 5$ on both benchmarks; because these are the two models the operator prior up-weights, this reflects both the informative initialization and the KL regularization toward the belief.

6 Conclusion

We presented VL-Scout, a unified stochastic routing framework for efficient multi-Modal and multi-step reasoning with turn-based reinforcement fusion. This paper makes three contributions. First, we combine Bayesian priors, policy gradient optimization, and sequential decision-making to enable structured exploration and adaptive inference. Second, a routing policy is trained via a KL-regularized REINFORCE objective by maximizing a reward-weighted evidence lower bound. Third, we introduce a fusion policy trained with GRPO or RLVR, which selects between two actions: CONTINUE, invoking another model via the router, or ANSWER, terminating with a generated output. Evaluated on four representative multi-modal benchmarks (MMMU, MMMU-Pro, OKVQA, and RealWorldQA), compared against 11 representative routing and policy optimization baselines, we show that VL-Scout offers up to a 5% improvement over the best individual model and prior-free routing baselines, while reducing premature policy collapse. We provide a reproducibility statement in Section 6.1, more details on experimental setup and datasets in Appendix D.

AI USE STATEMENT

In this work, we used generative AI tools for one task with required disclosure. We used Claude to help us to discover, navigate, and understand a proposition, for example, by explaining individual steps and clarifying the background results the proof relies on. We have not used generative AI tools for any other task with required disclosure. Specifically, we have not used them to generate synthetic datasets, develop theoretical models or conceptual frameworks, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, clean or reformat datasets, analyze data, or interpret results. Translation is not applicable to this work.

Additionally, we used generative AI tools (Claude and ChatGPT) for two tasks with recommended disclosure. First, we used them to edit the paper to improve readability. This included checking grammar, spelling, and phrasing in text written by the authors, reformatting equations and tables, suggesting corrections to mathematical notation, and drafting short connecting passages (such as a section introduction and a transition sentence) that summarize content already written by the authors. Second, we occasionally used them to create or modify scientific figures, either by generating plotting code to visualize results we had already obtained or by changing the visual style (e.g., colors, fonts, layout) of existing figures. These tools did not generate any of the research results or conclusions presented in this paper.

We have reviewed all AI-assisted work. Every suggested text edit was reviewed and accepted or rejected by the authors to make sure the original meaning and technical content were preserved, and every suggested notation correction was checked against our derivations. We independently verified each step of the proof and did not rely on AI-generated explanations without checking them. For every AI-assisted figure, we checked the plotting code and verified that the rendered figure accurately reflects the underlying experimental data. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

6.1 REPRODUCIBILITY STATEMENT

For VL-Scout implementation, a link to a downloadable source repository is included in our abstract. The source includes links for all the datasets, and we also provide the VLM outputs for each dataset. The details of our experiment are provided in Appendix D, which includes the selected hyperparameters and hardware specifications. We also provide examples of the outputs and prompts used in our paper in Appendix G.

REFERENCES

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Souradip Chakraborty, Sujay Bhatt, Udari Madhushani Sehwag, Soumya Suvra Ghosal, Jiahao Qiu, Mengdi Wang, Dinesh Manocha, Furong Huang, Alec Koppel, and Sumitra Ganesh. Collab: Controlled decoding using mixture of agents for llm alignment. *arXiv preprint arXiv:2503.21720*, 2025.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. Chateval: Towards better llm-based evaluators through multi-agent debate. *arXiv preprint arXiv:2308.07201*, 2023.

Lingjiao Chen, Matei Zaharia, and James Zou. Efficient online ml api selection for multi-label classification tasks. In *International conference on machine learning*, pp. 3716–3746. PMLR, 2022.

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.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in neural information processing systems*, 31, 2018.

J Russo Daniel, Van Roy Benjamin, Kazerouni Abbas, Osband Ian, and Wen Zheng. A tutorial on thompson sampling. *Foundations and Trends® in Machine Learning*, 11(1):1–99, 2018.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*, 2023a.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate, 2023b. URL https://arxiv.org/abs/2305.14325.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. Complexity-based prompting for multi-step reasoning. In *The Eleventh International Conference on Learning Representations*, 2022.

Yuqian Fu, Yuanheng Zhu, Jiajun Chai, Guojun Yin, Wei Lin, Qichao Zhang, and Dongbin Zhao. Rlae: Reinforcement learning-assisted ensemble for llms. *arXiv preprint arXiv:2506.00439*, 2025.

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V Chawla, Olaf Wiest, and Xiangliang Zhang. Large language model based multi-agents: A survey of progress and challenges. *arXiv preprint arXiv:2402.01680*, 2024.

Yichong Huang, Xiaocheng Feng, Baohang Li, Yang Xiang, Hui Wang, Ting Liu, and Bing Qin. Ensemble learning for heterogeneous large language models with deep parallel collaboration. *Advances in Neural Information Processing Systems*, 37:119838–119860, 2024.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. *arXiv preprint arXiv:2306.02561*, 2023.

Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.

Dawei Li, Bohan Jiang, Liangjie Huang, Alimohammad Beigi, Chengshuai Zhao, Zhen Tan, Amrita Bhattacharjee, Yuxuan Jiang, Canyu Chen, Tianhao Wu, et al. From generation to judgment: Opportunities and challenges of llm-as-a-judge. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 2757–2791, 2025.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PmLR, 2023.

Junyou Li, Qin Zhang, Yangbin Yu, Qiang Fu, and Deheng Ye. More agents is all you need. *arXiv preprint arXiv:2402.05120*, 2024.

Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pp. 661–670, 2010.

Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. On the advance of making language models better reasoners. *arXiv preprint arXiv:2206.02336*, 2022.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. Encouraging divergent thinking in large language models through multi-agent debate. *arXiv preprint arXiv:2305.19118*, 2023.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. Encouraging divergent thinking in large language models through multi-agent debate. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pp. 17889–17904, 2024.

Hui Liu, Chengqing Yu, Haiping Wu, Zhu Duan, and Guangxi Yan. A new hybrid ensemble deep reinforcement learning model for wind speed short term forecasting. *Energy*, 202:117794, 2020.

Shaoteng Liu, Haoqi Yuan, Minda Hu, Yanwei Li, Yukang Chen, Shu Liu, Zongqing Lu, and Jiaya Jia. Rl-gpt: Integrating reinforcement learning and code-as-policy. *arXiv preprint arXiv:2402.19299*, 2024a.

Zhengshang Liu and Kotagiri Ramamohanarao. Instance-based ensemble selection using deep reinforcement learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7. IEEE, 2020.

Zijun Liu, Yanzhe Zhang, Peng Li, Yang Liu, and Diyi Yang. A dynamic llm-powered agent network for task-oriented agent collaboration. In *First Conference on Language Modeling*, 2024b.

Costas Mavromatis, Petros Karypis, and George Karypis. Pack of llms: Model fusion at test-time via perplexity optimization. *arXiv preprint arXiv:2404.11531*, 2024.

Giovanni Monea, Antoine Bosselut, Kianté Brantley, and Yoav Artzi. Llms are in-context reinforcement learners. 2024.

Marcell Németh and Gábor Szűcs. Split feature space ensemble method using deep reinforcement learning for algorithmic trading. In *Proceedings of the 2022 8th International Conference on Computer Technology Applications*, pp. 188–194, 2022.

Quang H Nguyen, Thinh Dao, Duy C Hoang, Juliette Decugis, Saurav Manchanda, Nitesh V Chawla, and Khoa D Doan. Metallm: A high-performant and cost-efficient dynamic framework for wrapping llms. *arXiv preprint arXiv:2407.10834*, 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.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Ioannis Partalas, Grigorios Tsoumakas, and Ioannis Vlahavas. Pruning an ensemble of classifiers via reinforcement learning. *Neurocomputing*, 72(7-9):1900–1909, 2009.

Satheesh K Perepu, Bala Shyamala Balaji, Hemanth Kumar Tanneru, Sudhakar Kathari, and Vivek Shankar Pinnamaraju. Reinforcement learning based dynamic weighing of ensemble models for time series forecasting. *arXiv preprint arXiv:2008.08878*, 2020.

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.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Nithin Sivakumaran, Justin Chih-Yao Chen, David Wan, Yue Zhang, Jaehong Yoon, Elias Stengel-Eskin, and Mohit Bansal. Dart: Leveraging multi-agent disagreement for tool recruitment in multimodal reasoning. *arXiv preprint arXiv:2512.07132*, 2025.

Yanjie Song, Ponnuthurai Nagaratnam Suganthan, Witold Pedrycz, Junwei Ou, Yongming He, Yingwu Chen, and Yutong Wu. Ensemble reinforcement learning: A survey. *Applied Soft Computing*, pp. 110975, 2023.

Chuanneng Sun, Songjun Huang, and Dario Pompili. Llm-based multi-agent reinforcement learning: Current and future directions. *arXiv preprint arXiv:2405.11106*, 2024.

Selim Furkan Tekin, Fatih Ilhan, Tiansheng Huang, Sihao Hu, and Ling Liu. LLM-TOPLA: Efficient LLM ensemble by maximising diversity. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 11951–11966, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.698. URL https://aclanthology.org/2024.findings-emnlp.698/.

Selim Furkan Tekin, Fatih Ilhan, Gaowen Liu, Ramana Rao Kompella, and Ling Liu. Dynamic optimizations of llm ensembles with two-stage reinforcement learning agents. *arXiv preprint arXiv:2502.04492*, 2025.

Fanqi Wan, Xinting Huang, Deng Cai, Xiaojun Quan, Wei Bi, and Shuming Shi. Knowledge fusion of large language models. *arXiv preprint arXiv:2401.10491*, 2024.

Haozhe Wang, Chao Qu, Zuming Huang, Wei Chu, Fangzhen Lin, and Wenhu Chen. Vl-rethinker: Incentivizing self-reflection of vision-language models with reinforcement learning. *arXiv preprint arXiv:2504.08837*, 2025a.

Xinyuan Wang, Yanchi Liu, Wei Cheng, Xujiang Zhao, Zhengzhang Chen, Wenchao Yu, Yanjie Fu, and Haifeng Chen. Mixllm: Dynamic routing in mixed large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 10912–10922, 2025b.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, and Denny Zhou. Rationale-augmented ensembles in language models. *arXiv preprint arXiv:2207.00747*, 2022a.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*, 2022b.

Xumeng Wen, Zihan Liu, Shun Zheng, Shengyu Ye, Zhirong Wu, Yang Wang, Zhijian Xu, Xiao Liang, Junjie Li, Ziming Miao, et al. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base llms. *arXiv preprint arXiv:2506.14245*, 2025.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

xAI. Realworldqa. https://huggingface.co/datasets/xai-org/RealworldQA, 2024. Accessed: [Insert Date].

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36, 2024a.

Yuxuan Yao, Han Wu, Mingyang Liu, Sichun Luo, Xiongwei Han, Jie Liu, Zhijiang Guo, and Linqi Song. Determine-then-ensemble: Necessity of top-k union for large language model ensembling. *arXiv preprint arXiv:2410.03777*, 2024b.

Yao-Ching Yu, Chun-Chih Kuo, Ziqi Ye, Yu-Cheng Chang, and Yueh-Se Li. Breaking the ceiling of the llm community by treating token generation as a classification for ensembling. *arXiv preprint arXiv:2406.12585*, 2024.

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, 2024a.

Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, et al. Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark. *arXiv preprint arXiv:2409.02813*, 2024b.

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.

Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of reinforcement learning and control*, pp. 321–384, 2021.

Appendix

This appendix is organized by the following sections:

APPENDIX CONTENTS

- Appendix A: Impact Statement
- Appendix B: Active Model Routing: Design Guidelines
- Appendix C: Proof of Preposition 1
- Appendix D: Dataset and Framework Parameters
- Appendix E: Zero-shot Category Classification of Queries
- Appendix F: Prompts Used for VLMs and Fusion Model
- Appendix G: Examples of VL-Scout Fusion

A IMPACT STATEMENT

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

B ACTIVE MODEL ROUTING: DESIGN GUIDELINES

In this section, we build up the design of an active router step by step, starting from the simplest formulation and relaxing one assumption at a time. We consider a pool of K vision-language models and a stream of queries $(x_t, \mathbf{I}_t)$ that arrive online, each consisting of a text prompt and an image. For every query, the router selects one or more models from the pool and receives a reward that may be delayed, partial, or noisy; no labeled training data or prior knowledge of the models is assumed. We first cast routing as a context-free Multi-Armed Bandit solved with Thompson Sampling, which learns each model's marginal accuracy but ignores the query itself. We then introduce query context through LinUCB, a contextual bandit with a linear reward model, and add stochastic exploration by combining it with Thompson Sampling. Finally, we replace the linear model with a neural policy trained via REINFORCE to capture non-linear relationships between queries and models. For each design, we discuss its limitations, which together motivate the design of VL-Scout Route.

B.1 MULTI-ARMED BANDIT (MA) AND THOMPSON SAMPLING (TS).

Based on our motivations in Section 2, we first approach the problem as a dynamic environment that can be modeled with only two elements: action and reward. From the user's perspective, we assume that there is no prior knowledge of the models in the pool or how they behave, and the queries are independent of each other. The problem is a natural fit to the Multi-Armed Bandit, where each arm represents a model we prompt. Let a_t be the action at time t and r_t the reward (feedback), which may be delayed, partial, or noisy. The reward is given by $r_t = \mathcal{R}(a_t, y_t) + \epsilon_t$, where ϵ_t captures delay, partiality, or noise. For example, for binary feedback we define the reward function as

$$\mathcal{R}(a_t, y_t) = \begin{cases} 1 & \text{if } \hat{y}_t = y_t, \\ 0 & \text{otherwise}, \end{cases} \tag{10}$$

where $\hat{y}_t$ is the routed model's output. In this setting, the goal is to maximize the cumulative reward over horizon T:

$$\max \ \mathbb{E}\left[\sum_{t=1}^{T} r_t\right]. \tag{11}$$

This is equivalent to minimizing the regret, i.e., selecting the most rewarding model (arm) at each time step. As we do not know the models' reward distributions, Thompson Sampling (Daniel et al., 2018) can be used to balance exploration and exploitation. At each query, we maintain a posterior over each arm's expected reward γ_k, modeled with a Beta distribution as the prior belief, $\gamma_k \sim \mathrm{Beta}(\alpha_k, \beta_k)$.

The Beta distribution is also a natural fit when the reward is binary. For each arm $k \in \{1, \ldots, K\}$ we maintain counts α_k, the number of successes, and β_k, the number of failures of model k. For each query x, we sample a reward estimate from each arm's posterior and select the arm with the highest sample:

$$k^* = \arg\max_k \gamma_k, \qquad \gamma_k \sim \text{Beta}(\alpha_k, \beta_k). \tag{12}$$

After observing the reward $r \in \{0, 1\}$, the posterior is updated by

$$\alpha_k \leftarrow \alpha_k + r, \qquad \beta_k \leftarrow \beta_k + (1 - r). \tag{13}$$

Therefore, we stochastically sample plausible models and act optimally with respect to the sample. After each observation, the posterior update makes the distribution wider or narrower, resulting in more exploration or exploitation, respectively. However, the caveat of this approach is that it is independent of the query context, which tends to fail under unseen queries and unknown context. Thompson Sampling (Daniel et al., 2018) pools all the queries together, so α_k and β_k converge to the model's marginal accuracy.

B.2 Learnable Contextual Multi-Armed Bandits (LinUCB)

The contextual information should drive model routing; for example, a vision model that excels at OCR questions may fail at spatial reasoning. Previous works (Nguyen et al., 2024; Ong et al., 2024) showed that encoded semantic information of language queries can be used to decide which model should be queried. With a vision-language encoder, the input query can be mapped to a latent space, $g_{enc}(x, \mathbf{I}) = \mathbf{h}$, $\mathbf{h} \in \mathbb{R}^d$. Based on this context, a bandit can learn the reward distribution to select models. Therefore, we employ a contextual bandit algorithm, LinUCB (Li et al., 2010), which selects actions by combining the predicted reward with an uncertainty bonus. It learns a linear function that maps context to expected reward for each model, and uses an upper confidence bound (UCB) to favor actions that are either promising or uncertain. This yields a principled exploration–exploitation trade-off with strong regret guarantees. For each action $k \in \{1, \ldots, K\}$ and context $\mathbf{h}$, LinUCB assumes $\mathcal{R}(\mathbf{h}; \theta_k) = \mathbf{h}^\top \theta_k$, where θ_k is the unknown coefficient vector. For each model k we maintain the feature covariance matrix $\mathbf{A}_k \in \mathbb{R}^{d \times d}$ and the reward-weighted feature sum $\mathbf{b}_k \in \mathbb{R}^d$, and obtain the coefficients from the closed-form solution $\theta_k = \mathbf{A}_k^{-1}\mathbf{b}_k$. Given a new query x_t, the next arm is chosen with the UCB selection strategy:

$$k^* = \arg\max_k \quad \underbrace{\theta_k^\top \mathbf{h}}_{\text{predicted reward}} + \underbrace{\alpha\sqrt{\mathbf{h}^\top \mathbf{A}_k^{-1}\mathbf{h}}}_{\text{uncertainty}}, \tag{14}$$

where the uncertainty term is large when the context points in a direction the bandit has not yet explored. After observing reward r_t for the selected arm, the update rule is:

$$\mathbf{A}_k \leftarrow \mathbf{A}_k + \mathbf{h}\mathbf{h}^\top, \qquad \mathbf{b}_k \leftarrow \mathbf{b}_k + r_t\mathbf{h}, \qquad \theta_k \leftarrow \mathbf{A}_k^{-1}\mathbf{b}_k. \tag{15}$$

LinUCB provides a strong baseline with theoretical regret guarantees under linear realizability assumptions. However, its assumption of a linear relationship between context and reward can be restrictive, particularly under cross-domain shifts. In addition, the UCB bonus is fully deterministic given the data, which may lead to the same arm being chosen whenever the same context is given, as there is no randomness at all. This means that the policy cannot explore when two arms have identical UCB scores. One of our design goals is to create a more robust algorithm against contextual domain shifts.

B.3 LinUCB + Thompson Sampling

We can replace the deterministic UCB score with a stochastic perturbation by sampling from a Bayesian posterior over the weight vectors:

$$\tilde{\theta}_k \sim \mathcal{N}\big(\theta_k,\ v^2\mathbf{A}_k^{-1}\big)\,, \tag{16}$$

where $v^2 \in \mathbb{R}$ scales the uncertainty. Each arm is then scored with its sampled weights, $\tilde{\theta}_k^\top \mathbf{h}$, and the update rule is the same as in Eq. (15). As a result, we get randomized exploration from Thompson Sampling (TS) on top of the deterministic safety floor from UCB.

B.4 REINFORCE FOR CONTEXTUAL BANDIT ROUTING.

There are two remaining concerns. First, the relationship between the context $\mathbf{h}$ and the action can be non-linear, which could be modeled via a multi-layer neural network with gradient-based optimization. Second, the previous algorithms use the immediate rewards in the update, whereas buffering rewards yields more stable updates. Therefore, we adopt REINFORCE (Williams, 1992), a policy-gradient algorithm, for contextual bandit routing. Instead of maintaining a posterior over weights, it directly parameterizes a stochastic policy π_θ as a neural network and optimizes it via gradient ascent on the expected reward:

$$J(\theta) = \mathbb{E}_{a\sim\pi_\theta(\cdot|x)}\left[\mathcal{R}(a,x)\right]. \tag{17}$$

Here, we use a two-layer MLP as the policy, with a sigmoid function σ at the output layer:

$$\pi_\theta(a_t \mid \mathbf{h}_t) = \sigma(\mathbf{W}_L\,(\ldots \mathrm{ReLU}(\mathbf{W}_1\mathbf{h}_t)\ldots)) = [p_1, p_2, \ldots, p_K]. \tag{18}$$

The output has K dimensions, where $p_k = \Pr(a_k = 1 \mid \mathbf{h}_t;\theta)$ is the probability of model k being included in the selected pool, and action a_k is drawn from a Bernoulli distribution with success probability p_k, i.e., $a_k \sim \mathrm{Bernoulli}(p_k)$. By the log-derivative trick, the policy gradient is estimated over a buffer of T transitions:

$$\nabla_\theta J(\theta) \approx \frac{1}{T}\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_t \mid \mathbf{h}_t)\, r_t. \tag{19}$$

We also use reward normalization to reduce gradient variance. Compared to a Markov Decision Process, the contextual bandit has no transition model to learn, since each context is independent of its predecessor. However, REINFORCE leaves two issues open. First, the user who submits a query may have a prior belief about the models for some categories, e.g., 'Deepseek-Math is more successful than Gemma-7B for math-science related tasks', which the policy cannot exploit. Second, due to the stochastic model selections made under noisy or delayed rewards, REINFORCE may converge prematurely to suboptimal models with insufficient exploration.

These design considerations serve as guidelines for developing our VL-Scout Route, which enables Thompson-style exploration while serving as a prior for regularization. Our experiments section will show the advantages of VL-Scout Route compared to the above four baselines.

C PROOF OF PROPOSITION 1

The normalizer Z. The constant

$$Z = \sum_{j=1}^{K} \tilde{p}_j\, e^{r(j)/\beta} = \mathbb{E}_{j\sim\tilde{\mathbf{p}}}\left[e^{r(j)/\beta}\right]$$

is the normalizing constant of $\pi^\star$: the prior-weighted average of the exponentiated rewards, which ensures $\sum_k \pi^\star(k) = 1$. It depends on $\tilde{\mathbf{p}}$, r, and β, but not on π. Rearranging the definition of $\pi^\star$ gives, for every k,

$$\tilde{p}_k\, e^{r(k)/\beta} = Z\,\pi^\star(k). \tag{20}$$

For example, with two models, $\tilde{\mathbf{p}} = (0.5, 0.5)$, $r = (1, -0.1)$, and $\beta = 0.8$, we get $Z = 0.5\,e^{1.25} + 0.5\,e^{-0.125} \approx 2.19$ and $\pi^\star \approx (0.80, 0.20)$: the reward tilts the prior's even split toward the model that answered correctly.

First statement. Write both terms of the objective as sums over the models,

$$\mathbb{E}_\pi[r] = \sum_k \pi(k)\, r(k), \qquad \beta\,\mathrm{KL}(\pi\|\tilde{\mathbf{p}}) = \beta\sum_k \pi(k)\log\frac{\pi(k)}{\tilde{p}_k},$$

and write each reward as a logarithm, $r(k) = \beta\log e^{r(k)/\beta}$, so that $\mathbb{E}_\pi[r] = \beta\sum_k \pi(k)\log e^{r(k)/\beta}$. Both terms now have the form $\beta\sum_k \pi(k)\log(\cdot)$ and combine into a single sum:

$$\mathbb{E}_\pi[r] - \beta\,\mathrm{KL}(\pi\|\tilde{\mathbf{p}}) = \beta\sum_k \pi(k)\Big[\log e^{r(k)/\beta} - \log\pi(k) + \log\tilde{p}_k\Big] = -\beta\sum_k \pi(k)\log\frac{\pi(k)}{\tilde{p}_k\, e^{r(k)/\beta}}.$$

The expected reward has not disappeared: it is carried by the factor $e^{r(k)/\beta}$ in the denominator. Substituting Eq. (20) and splitting the logarithm gives

$$-\beta \sum_k \pi(k) \log \frac{\pi(k)}{Z\,\pi^\star(k)} = -\beta \sum_k \pi(k) \log \frac{\pi(k)}{\pi^\star(k)} + \beta \log Z \sum_k \pi(k) = \beta \log Z - \beta\, \mathrm{KL}(\pi \| \pi^\star),$$

where the last step uses $\sum_k \pi(k) = 1$. Since Z does not depend on π, and $\mathrm{KL}(\pi\|\pi^\star) \geq 0$ with equality if and only if $\pi = \pi^\star$, $\pi^\star$ is the unique maximizer. The maximum value, $\beta \log Z$, is a soft maximum of the rewards under the prior: it lies between $\mathbb{E}_{k\sim\tilde{\mathbf{p}}}[r(k)]$ (by Jensen's inequality) and $\max_k r(k)$.

Second statement. Let $P(O{=}1 \mid k) = e^{r(k)/\beta}/C$ with $C \geq \max_k e^{r(k)/\beta}$, so that these are valid probabilities. Then $P(O{=}1) = \sum_k \tilde{p}_k\, P(O{=}1 \mid k) = Z/C$, so $\log P(O{=}1) = \log Z - \log C$, and for every π, by Jensen's inequality,

$$\log P(O{=}1) \;\geq\; \mathbb{E}_\pi\big[\log P(O{=}1 \mid k)\big] - \mathrm{KL}(\pi\|\tilde{\mathbf{p}}) = \frac{1}{\beta}\Big(\mathbb{E}_\pi[r] - \beta\, \mathrm{KL}(\pi\|\tilde{\mathbf{p}})\Big) - \log C,$$

with equality if and only if $\pi(k) = P(k \mid O{=}1) = \tilde{p}_k\, e^{r(k)/\beta}/Z = \pi^\star(k)$. □

D Dataset and Framework Parameters

Datasets. The experiments are performed on four popular benchmark datasets: MMMUYue et al. (2024a), MMMU-Pro Yue et al. (2024b), A-OKVQA Schwenk et al. (2022), and RealWorldQA xAI (2024). in multiple-choice question format. MMMUYue et al. (2024a) dataset is designed to evaluate VLMs on massive multi-discipline tasks containing 30 different subjects. The questions are at the college level and contain multiple reasoning steps. The dataset only provides 5 development and 900 validation samples with the labels. However, the labels of the test split are hidden. MMMU-Pro dataset Yue et al. (2024b), on the other hand, is the robust version of MMMU and provides only a test dataset with labels. A-OKVQA dataset Schwenk et al. (2022) provides visuals with questions that require reasoning to solve. The expected answer is either the choice from the multiple choices given or a direct answer. There are 17,056 training pairs, 1,146 validation pairs, and 6,70 test samples. We use the validation set for evaluation. Similar to the MMMU dataset, the A-OKVQA benchmark did not share the test labels. Thus, we used a validation split as the test data. RealWorldQA is a benchmark designed for real-world understanding. The dataset consists of anonymized images taken from vehicles, in addition to other real-world images. It contains spatial understanding that requires OCR and object detection. It contains 764 images and questions, where questions have 3-4 choices.

Evaluation Metrics. To evaluate the performance of VLMs and V3Fusion methods, we use accuracy on MMMU, MMMU-Pro and A-OKVQA datasets.

VLM selection for the candidate model pool. In our model pool selection, we choose pre-trained individual VLMs based on the following three elements and their effect on performance: (i) size of the model (less than 16b), (ii) model variety, and (iii) being open-source. The size was the limiting factor for us since the modern LLMs are huge, and our experiments were conducted on a single H100 GPU of 80GB.

VL-Scout Policy Models and Structural Details. VL-Route REINFORCE model contains two fully connected hidden layers with 100 neurons and ReLU activations between the layers. The model weights start from Xavier initialization and are optimized by Adam during online training. For the VL-Scout Fusion, we have used `Qwen/Qwen3-VL-4B-Instruct` and during training we have used LORA with rank 8 and $\mathrm{lora_}\alpha = 16$ and trained with Decoupled Adam.

E Zero-shot Category Classification of Queries

The Table 3 shows the main categories and the descriptions that fall under each category. We have used `nli-MiniLM2-L6-H768` zero-shot language classifier to classify each question into one of the categories.

Table 3: Question Categories and Descriptions

Category	Description
spatial_and_visual	Where, position, layout, scene understanding
counting_and_quantity	How many, amounts, numbers
text_and_document	OCR, reading, documents, charts
science_and_math	Physics, chemistry, biology, math, engineering
humanities_and_arts	History, literature, music, art, philosophy
social_and_applied	Economics, law, medicine, psychology, business
perception_and_attribute	Color, shape, material, appearance, action
knowledge_and_reasoning	World knowledge, causal, comparison, why

Table 4: Prompt Configurations

Prompt	Description	Content
VQA System Prompt	Base prompt. Instructs the model to output the final answer only, no reasoning.	You are a visual question answering assistant. For every question, first write a short rationale describing what you observe and how you reason, then give your final answer. Always use this format: Rationale: [your reasoning] Answer: [your final answer]]
Decision Agent Prompt	Dynamic multi-model prompt. Decides whether to STOP and answer or CONTINUE querying more models.	You are a decision agent that determines whether to STOP and answer, or CONTINUE querying more models. Query: {query} Responses collected so far: Model {i+1}: Answer: {resp['answer']} Rationale: {resp['rationale']} Models remaining in pool: {remaining} Think about whether the current responses are sufficient: - Do the models agree? - Is the rationale strong and complete? - Would another model's perspective help? If sufficient → ANSWER: <final answer> If not → CONTINUE

F Prompts Used for VLMs and Fusion Model

The Table 4 shows the prompts we have used for inferring the base models in the pool (first row) and the prompt for the VL-Scout Fusion model to decide on CONTINUE or ANSWER (second row).

Table 5: Model Rationales and Answers for Example-4 for Figure-2

Question: Is the closest car closer to us than the pedestrian?				
Model	**Rationale**	**Answer**	**Decision**	**Correct?**
`Qwen` (Model 1)	The pedestrian is visible on the sidewalk to the right, standing near a tree and some snow piles. The closest car is further down the street, behind the pedestrian. Since the pedestrian is on the sidewalk and the car is on the road and further away, the pedestrian is closer to the viewer than the car.	Yes	CONTINUE	×
`InternVL` (Model 2)	The closest car is on the left side of the road, while the pedestrian is on the right side. By comparing their relative positions to the camera's viewpoint, the car appears closer to the viewer than the pedestrian.	Yes	CONTINUE	×
`Gemma` (Model 3)	The image shows a street scene. The closest car is driving down the middle of the road, relatively far away. The pedestrian is walking on the sidewalk on the right side, and is much closer to the camera than the car. Therefore, the pedestrian is closer to us than the closest car.	1	ANSWER: No	✓
Ground Truth		**No**		✓

Table 6: Model Rationales and Answers Example-1 for Figure-2

Question: Is there a stop sign?				
Model	**Rationale**	**Answer**	**Decision**	**Correct?**
`InternVL` (Model 1)	To determine if there is a stop sign, I examine the street for any octagonal signs with a red border and white background. Upon reviewing the image, I see several signs along the street, but none match the description of a stop sign. The signs visible are either rectangular or do not have the characteristic red border and white center.	No	CONTINUE	✓
`Qwen` (Model 2)	I can see a white SUV parked on the left side of the street, and behind it, there appears to be a stop sign mounted on a pole near the sidewalk. The sign is partially obscured by the SUV and the angle of the photo, but its shape and position are consistent with a stop sign. Therefore, based on the visual evidence, there is at least one stop sign present.	Yes	CONTINUE	×
`Gemma` (Model 3)	I see a street with cars parked on the side and yellow crosswalk markings. I do not see any stop signs visible in the image.	0	ANSWER: No	✓
Ground Truth		**No**		✓

G EXAMPLES OF VL-SCOUT FUSION

We show the rationales and answers by each base VLM in Tables 6 and 5 for cases in Figure 2. These two examples illustrate how the sequential querying strategy handles both model disagreement and majority error. In the first example (closest car vs. pedestrian), Models 1 and 2 confidently agree on the wrong answer, yet Model 3 correctly identifies the depth relationship, steering the decision agent toward the correct final answer. In the second example (stop sign detection), Model 2 hallucinates a

partially obscured stop sign, while Models 1 and 3 independently and correctly find none present. In both cases, the decision agent successfully resolves conflicts and overrides incorrect responses by continuing to query additional models, demonstrating the system's robustness against individual model failures and hallucinations.