VISION VERIFICATION ENHANCED FUSION OF VLMS FOR EFFICIENT VISUAL REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Ensemble of either large language models or large vision models has been studied extensively to date. However, there is an increasing demand on tackling the problem of ensemble of diverse Vision-Language Models (VLMs) based on both vision and language modalities jointly. This paper presents V3Fusion, a Vision-Verification enhanced VLM Fusion framework for high performance visual reasoning with three unique contributions. First, V3Fusion introduces focal error diversity to capture complementary reasoning across VLMs by combining language-based focal error diversity with a Focal Central Kernel Alignment (CKA) based focal diversity metric (CKA-focal) to measure disagreement in visual embeddings. Second, on the constructed ensemble surface from a pool of candidate VLMs, we applied a Genetic Algorithm to effectively prune out those component VLMs that do not add value to the fusion performance. Finally, we identify the best combination for each reasoning task and resolve conflict among the outputs of multiple VLMs in the ensemble model pool, enabling heterogeneous models to capture epistemic uncertainty dynamically while mitigating hallucinations. Our V3Fusion approach is capable of producing dual focal-diversity fused predictions with high performance for vision-language reasoning, even when there is no majority consensus or the majority of VLMs make incorrect predictions. Extensive experiments with four popular VLM benchmarks (A-OKVQA, MMMU, MMMU-Pro, and OCR-VQA) validate the effectiveness of V3Fusion. The results show that V3Fusion outperforms the best-performing VLM on MMMU by 4.54% and MMMU-Pro by 2.29% gain in accuracy. For generative tasks, V3Fusion outperforms Intern-VL2-8b and Qwen2.5-VL-7b, the top-2 VLM performers on both A-OKVQA and OCR-VQA. Code is available at `anonymous.4open.science/r/v3fusion-30D1/`

1 INTRODUCTION

The modern Large Vision Language Models (VLMs) (Wu et al., 2024; Wang et al., 2024; Bai et al., 2025; Liu et al., 2023; Chen et al., 2024b;a) show remarkable performance on visual question answering and visual reasoning tasks. Different from standard text-only LLMs, VLMs are trained jointly with the visual extractor (Li et al., 2023; 2022; Zhai et al., 2023; Radford et al., 2021a; Jia et al., 2021), the language decoder (Touvron et al., 2023; Achiam et al., 2023; Team et al., 2024; Jiang et al., 2024; Vaswani et al., 2017). However, a VLM can extract the visual information thoroughly but fails to make deep connections and reasoning based on the given query; or it may have a deeper reading capacity but fail to detect or misidentify the objects in the image. Despite the existence of these flaws, different VLMs offer diverse generalization performance. Smaller VLMs may outperform larger VLMs on certain questions (Tekin et al., 2024b). In this paper, we argue that one can compare and contrast the capabilities of VLMs with each other to find the best model combination and generate the most refined and accurate output. Based on this comparison, we also demonstrate that one can model the total uncertainty and detect when the model hallucinate and be rectified its illusion. However, this task has significant challenges. First, from a large pool of VLMs, one needs to select the best model combination for the ensemble set by analyzing their outputs. The number of combinations that can be produced grows exponentially with the number of candidate models. Second, each model in the set can produce conflicting outputs; therefore, the generation mechanism should establish the best result for the target task by resolving the conflicts. Third, the source of uncertainty could be inherent in the data or model parameters, and an uncertainty threshold should be dynamically determined to reject highly uncertain answers. Lastly, the rejected samples should be rectified to produce a better answer.

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.

Recent works approach the challenges by prompting an auxiliary LLM to generate an uncertainty score (Manakul et al., 2023; Xia et al., 2025) or creating a debate environment (Zhao et al., 2025; Liang et al., 2023) or target the language embeddings for heterogeneous LLMs (Huang et al., 2024). However, these methods are prompt-dependent, costly, and they do not target modalities in VLMs.

To this end, we introduce Visual Verification VLM Fusion, `V3Fusion`, visual and language diversity-optimized VLM ensemble method with five unique properties: (i) a Focal Central Alignment Kernel (CKA), or *Focal-CKA*, to measure the average similarity between visual encoder embeddings in an ensemble set, (ii) *Focal-Diversity* metric to capture the error diversity and the diversity-performance correlation among component VLMs of an ensemble; (iii) a diversity-optimized ensemble pruning algorithm to identify and select the best sub-ensembles from a pool of N base LLMs; (iv) V3Fusion-MLP and V3Fusion-LED are more specific yet powerful fusion models for multiple-choice and open-ended questions. V3Fusion-MLP leverages the probability distribution assigned to choices by candidate models, learning to detect patterns among model logits and find the correct solution even when the majority of candidate models are incorrect. V3Fusion-LED, learns to generate answers for open-ended questions by comparing and contrasting base-model outputs. (v) Finally, we develop a reflection mechanism based on the epistemic uncertainty of the system to reject or accept the generated output based on an adaptive threshold algorithm, *adaptive epistemic uncertainty*.

2 Related Work

For the Ensemble Learning in LLMs, classical methods such as majority voting are employed to ensemble multiple outputs generated from the same LLM, (Wang et al., 2022; Li et al., 2024a). The confidence score calculated by either a model (Yao et al., 2024) or deduced from the model itself by the logits of tokens produced by the model is also used for voting (Zhang et al., 2024; Li et al., 2024b). Another thread of literature is creating a debate environment (Liang et al., 2023; Wan et al., 2024; Du et al., 2023; Chan et al., 2023) with the cost of lengthy and complex prompt strategies. In addition, several supervised summarization LLM ensemble methods are proposed, such as LLM-Blender (Jiang et al., 2023b) and TOPLA-Summary (Tekin et al., 2024a). These methods formalize the ensemble as a summarization problem using a seq2seq model. Moreover, models with Mixture-of-Expert (MoE) layers are introduced (Shazeer et al., 2017; Jiang et al., 2023a; 2024), which are also used by the authors of (Wu et al., 2024) to create a VLM model and for alignment through fusion in (Tekin et al., 2024b). Alternatively, a distillation strategy is proposed in (Wan et al., 2024) by performing a token alignment on the probability distributions of the models. Regarding the model routing, many works such as (Chen et al., 2023; Zhang et al., 2025; Ong et al., 2024) reduced the cost of inference by performing prompt adaptation, caching, and model tuning to choose the strongest model in the pool. LLM-based Multi-Agents (Guo et al., 2024; Du et al., 2023) carry similar motivations in terms of exploiting multiple LLMs to work collaboratively for a particular task. However, these works do not incorporate the visual modality while performing ensemble, routing, or collaboration.

In the domain of ensemble learning for VLMs, (Lu et al., 2023) made logit level aggregation for zero-few shot learning, (Agnolucci et al., 2023) focused on the visual encoder and learn an ensemble of prompts for image classification, and (Chen et al., 2025d) ensembles prompts in VLM adaptation. Many works (Tran et al., 2025; Guo et al., 2024) employed ensemble learning in their multi-agent frameworks, such as video understanding (Chen et al., 2025a) and propose the majority option as a pseudo-label and evaluates the agents. Authors of (Sivakumaran et al., 2025) introduced multi-VLM debate with external expert tools. Closest to our work, the authors of (Alazraki et al., 2023) explore ensemble strategies for VQA models. However, the authors shows a high potential for ensemble, yet the classical methods applied on cumulative logits of generated tokens, e.g. majority voting, weighted voting, and training a classifier to learn the weights, cannot improve the best model performance. Thus, they use an external LLM (PaLM (Anil et al., 2023)) to evaluate the outputs based on the self-reflection of each candidate model. Differently, as we show in this paper, the probability assigned to tokens representing the choices is highly significant, and a trained ensemble model can improve the base model's performance significantly without the need for an external LLM.

3 Preliminaries

We first define the VLM ensemble problem and our objective. We then examine visual language models in greater detail, focusing on the architectural differences that drive their diverse capabilities. As a preliminary study, we measure the pairwise error correlations across several widely used VLMs, revealing complementary failure patterns (see Appendix B). Motivated by that, our goal is to design

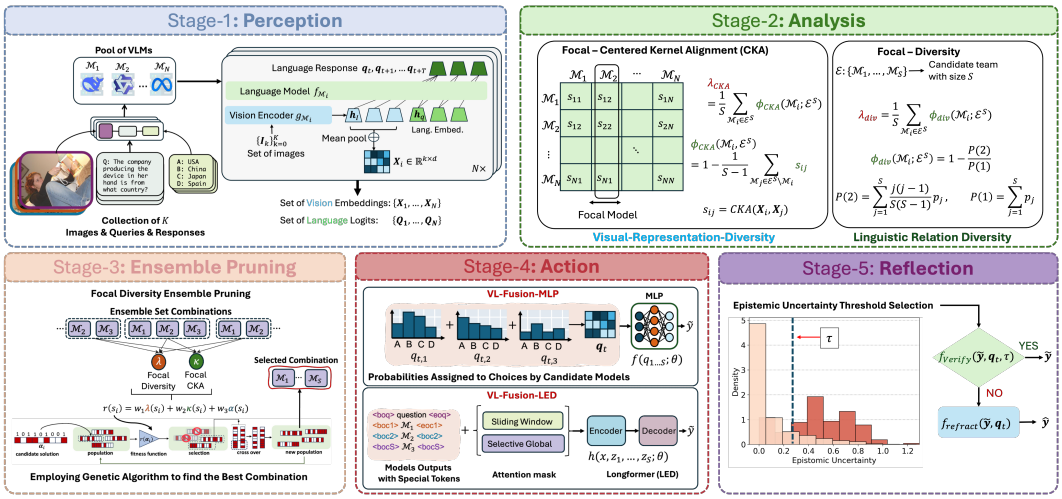


Figure 1: The V3Fusion Framework: An overview.

a mechanism that selects candidate models based on their error correlations while preserving and leveraging their individual expertise.

3.1 Multiple VLMs Ensemble Problem

Let $(\mathbf{x}, \mathbf{I})$ denote an input tuple of a language instruction and an image, where the instruction $\mathbf{x} = \{\omega_1, \ldots, \omega_L\}$ is composed of L tokens ω_i, and the image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$ represents the RGB input. For task T under a VLM $\mathcal{M}$ and let $\mathbf{y}$ denote the desired output. We assume a dataset $\mathcal{D}$ to be the collection of observations for task T, such that $(\mathbf{x}, \mathbf{I}, \mathbf{y}) \in \mathcal{D}$, where $|\mathcal{D}| = K$. For a pool of VLMs with the size N, denoted as $\mathcal{M}_1, \ldots, \mathcal{M}_N$, we utilize $\mathcal{D}$ to find the optimal ensemble function f_{ens}. This function takes outputs of each LLM and yields one final answer, denoted as $\widetilde{y}$, given by $f_{\text{ens}}(\mathcal{M}_1(\mathbf{x}, \mathbf{I}), \ldots, \mathcal{M}_N(\mathbf{x}, \mathbf{I})) = \widetilde{y}$, such that the difference between desired output is minimized, measured by the loss function $\mathcal{L}(\widetilde{y}, y)$. However, based on the task T, the desired output y can represent different solution spaces. For multiple-choice questions (MCQ), y is a single token, $y^{(1)} \in \{1, \ldots, m\}$ representing the choices where m is number of choices. For open-ended questions (OEQ), y consists of multiple tokens representing a single word, a real number, or a sequence of words and is denoted as $y^{(2)} = \{\omega_1, \ldots, \omega_L\}$, where L is the sequence length.

3.2 Closer Look at Visual Language Models

A standard VLM consists of two parts, visual encoder and language decoder as shown in the first step of Figure 1, *Perception*. The visual encoder of model $\mathcal{M}_i$, denoted by $g_{\mathcal{M}_i}$, takes the an input image and produces visual embeddings $g_{\mathcal{M}_i}(\mathbf{I}_k) = [\mathbf{h}'_{I,1}, \ldots, \mathbf{h}'_{I,L'}]$ and most VLMs e.g. (Liu et al., 2023) are using Vision Transformers (ViT) such as CLIP (Radford et al., 2021b) and use a projection matrix to map the image embeddings to language space to generate visual token embeddings. For each token $\mathbf{h}_I = \mathbf{W} \cdot \mathbf{h}'_I$, where $\mathbf{h}_I \in \mathbb{R}^d$, model creates the visual embeddings $\mathbf{X} = [\mathbf{h}_{I,1}, \ldots, \mathbf{h}_{I,L'}]$. In the second phase, the language model takes language token embeddings $\mathbf{Q} = [\mathbf{h}_{q,1}, \ldots, \mathbf{h}_{q,L}]$ together with the visual tokens to generate the first step of the response, t, denoted by $f_{\mathcal{M}_i}(\mathbf{X}, \mathbf{Q}) = \mathbf{q}_t$, where $\mathbf{q}_t \in \mathbb{R}^d$. The language model then, auto regressively predicts the next token, ω_{t+1}, by employing the *softmax* function to model the probability mass conditioned on the input vision and language tokens $\mathbf{X}, \mathbf{Q}$, and the formerly generated tokens, $\omega_{<t}$, i.e. it models: $p(y|\mathbf{I}, \mathbf{x}) = \prod_{t=1}^{L} p_{\mathcal{M}_i}\big(y_t | y_{<t}, g_{\mathcal{M}_i}(\mathbf{I}), f_{\mathcal{M}_i}(\mathbf{x})\big)$. This formulation carries significant dependencies to visual tokens and language tokens. For the input image, if the visual tokens cannot represents the its important aspects, the language model cannot create connections with to input prompt. Since our goal is to compare and contrast different VLMs, we must demystify the visual and language understanding of the model and answer questions, such as: *"Does the model correctly encode the image so that it preserve the semantic structure?"* or *"Does the answer of the model faithfully reflect the content of the image?"* Therefore to answer these questions, we design the *Analysis* step which is the core of our methodology.

4 VISUAL–LINGUISTIC ANALYSIS OF VLMS

Given a pool of candidate (base) VLMs as an ensemble, V3Fusion first performs the focal diversity-based ensemble pruning for two reasons: First, the diversity among base models improves the ensemble performance (Breiman, 1996; Dietterich, 2000). Second, as we add more models to the ensemble pool, it becomes more expensive to prompt each model, and the input length of the ensemble model increases. Thus, the model selection for an ensemble set is essential, and when you consider a pool of N base models, the total number of possible ensemble teams with size S ($2 \leq S \leq N$) is $2^N - N - 1$. A key question is how to perform ensemble pruning efficiently. For better generation performance by an ensemble, we argue that the models in the ensemble must have diverse perception and must capture diverse linguistic relations; thus, a designed metric for ensemble pruning in a multi-vision-language system must respect these aspects. Before defining such metrics, we first determine the subset of the input space on which they should be evaluated. A model in the ensemble set $\mathcal{M}_i$ sees kinds of examples where difficulty varies e.g. $\mathcal{D} = \mathcal{D}_{\text{hard}} \cup \mathcal{D}_{\text{easy}}$. If a diversity metric $\phi(\mathcal{M}_i, \mathcal{M}_j))$ that measures a representation dissimilarity between models i and j uses the whole observation, $\mathbb{E}_{\mathcal{D}}\left[\phi(\mathcal{M}_i, \mathcal{M}_j)\right]$, then it estimates the dissimilarity for the global representation. If we condition on focal failures, $\mathbb{E}\left[\phi(\mathcal{M}_i, \mathcal{M}_j) \mid \mathcal{M}_i(\mathbf{x}, \mathbf{I}) \neq \mathbf{y}\right]$ changes the quantity from global representation dissimilarity to failure-conditioned representation dissimilarity. However, this yields a pairwise dissimilarity measure between two models. To obtain a single score for a candidate ensemble $\mathcal{E}$, we choose one of the S base models each turn as the focal model and aggregate its diversity with respect to the remaining models over its negative episodes. We denote this general aggregation as $\rho^{focal}(\mathcal{E}, \phi)$, where ϕ represents the underlying diversity metric, then:

$$\rho^{focal}\left(\mathcal{E}, \phi\right) = \frac{1}{|\mathcal{E}|} \sum_{\mathcal{M}_i \in \mathcal{E}_i} \phi(\mathcal{M}_i; \mathcal{E} \setminus \mathcal{M}_i) \tag{1}$$

And subsequently, we introduce two metrics $\rho^{\text{focal}}(\mathcal{E}, \phi_{\text{CKA}})$ for visual understanding diversity and $\rho^{\text{focal}}(\mathcal{E}, \phi_{\text{div}})$ for capturing linguistic task relations.

4.1 VISUAL-REPRESENTATION-DIVERSITY METRIC

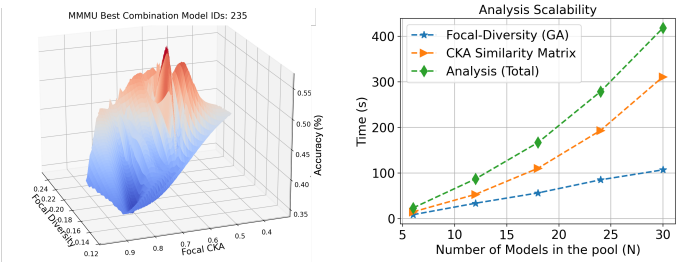
We seek to identify informative complementary representations where a model made a mistake. During forward propagation of each VLM in the pool of N models, we hook the visual-encoders $g_{\mathcal{M}_1}, \ldots, g_{\mathcal{M}_N}$ and obtain the visual token embeddings. However, each model utilizes a different ViT with various structural differences, e.g., different patch sizes and pooling mechanisms, and transformations. Thereby, each model generates embeddings with different dimensions. To address these challenges, we employ Central Kernel Alignment (CKA) (Kornblith et al., 2019; Cortes et al., 2012; Raghu et al., 2021), which enables quantitative comparison by looking at the relative relation between the embeddings of samples extracted by different models. First, we take the mean-pool of vision token embeddings of model i, denoted by $\widetilde{\mathbf{h}}_i = \text{MeanPool}([\mathbf{h}_{\text{I},1}, \ldots, \mathbf{h}_{\text{I,L}'}])$. We repeat for K samples to obtain $\widetilde{\mathbf{X}}_i \in \mathbb{R}^{K \times d_i}$ and d_i is the embedding dimension of model-i. Let $\widetilde{\mathbf{X}}_j$ be the embedding of model j. Gram matrices of both embeddings are then computed as $\mathbf{K} = \widetilde{\mathbf{X}}_i \widetilde{\mathbf{X}}_i^\top$ and $\mathbf{L} = \widetilde{\mathbf{X}}_j \widetilde{\mathbf{X}}_j^\top$ capturing the internal similarity structure of each embedding space. Then, CKA compares the pairwise similarity by calculating:

$$\text{CKA}(\widetilde{\mathbf{X}}_i, \widetilde{\mathbf{X}}_j) = \frac{\text{HSIC}(\mathbf{K}, \mathbf{L})}{\sqrt{\text{HSIC}(\mathbf{K}, \mathbf{K})\text{HSIC}(\mathbf{L}, \mathbf{L})}}, \tag{2}$$

where HSIC is the Hilbert Schmidt Independence Criterion which is calculated as $\text{HSIC}(\mathbf{K}, \mathbf{L}) = \frac{\text{vec}(\mathbf{K}') \cdot \text{vec}(\mathbf{L}')}{n-1}$. Here $\mathbf{K}' = \mathbf{HKH}$ and $\mathbf{L}' = \mathbf{HLH}$ are centered by $\mathbf{H} = \mathbb{I}_n - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$ ensuring the invariance to isotropic scaling. CKA, also, invariant to orthogonal transformation of embeddings, enabling meaningful comparison and analysis of extracted sample embeddings.

Focal-CKA: As shown in the second stage of Figure 1, we first obtain a CKA similarity matrix $\mathbf{S} \in [0, 1]^{N \times N}$, where $s_{ij} = \text{CKA}(\widetilde{\mathbf{X}}_i, \widetilde{\mathbf{X}}_j)$, $s_{ij} = s_{ji}$, and $s_{ii} = 1$. Next, we select a focal model $\mathcal{M}_i$ inside the ensemble set and compute the average pairwise CKA similarity, ϕ_{CKA}, between the focal model and the other models in the set, denoted by:

$$\phi_{\text{CKA}}\left(\mathcal{M}_i; \mathcal{E}\right) = 1 - \frac{1}{|\mathcal{E}| - 1} \sum_{\mathcal{M}_j \in \mathcal{E} \setminus \mathcal{M}_i} s_{ij} \tag{3}$$



(a) (b)

Figure 2: (a) All candidate ensemble teams from the model pool are plotted with their Focal Diversity, Focal-CKA, and Plurality Voting metrics for MMMU dataset. (b) Scalability of the analysis stage with breakdown.

Table 1: Brute Force (BF) and Genetic Alg. (GA) pruning comparison with time reduction percentage.

VLM Count	Time(s) BF↓	Time(s) GA↓	Reduction (%)↑
$N = 5$	9.4	9.9	−5.31
$N = 10$	228.2	24.5	89.26
$N = 15$	508.99	41.8	91.78
$N = 20$	16201.51	54.9	99.66

Here, we subtract the average from 1 to obtain a score representing visual diversity, where 1 represents the most dissimilar and 0 represents the most similar visual representation. We repeat this process for each model in the ensemble set to obtain Focal-CKA, $\rho^{\text{focal}}(\mathcal{E}, \phi_{\text{CKA}}) = \lambda_{\text{CKA}}$, using equation 1.

4.2 Capturing Error Diversity for Linguistic Relations

The Focal-CKA alone can identify what is different and give ensemble sets with diverse perceptions. To measure the usefulness of the difference, we introduce Focal-Diversity, $\lambda_{\text{div}} = \rho^{\text{focal}}(\mathcal{E}, \phi_{\text{div}})$, which quantifies the level of error diversity among the component models of an ensemble with respect to each model, thereby capturing the potential for failure recovery. As shown in the second stage of Figure 1, we calculate the focal negative correlation score ϕ_{div} by calculating two probability terms of the focal model using its negative episodes in a candidate ensemble set $\mathcal{E}$:

$$\phi_{\text{div}}(\mathcal{M}_i; \mathcal{E}) = 1 - \frac{P(2)}{P(1)} \quad \text{, where } P(1) = \sum_{j=1}^{S} \frac{j}{N} p_j \text{ and } P(2) = \sum_{j=1}^{S} \frac{j(j-1)}{S(S-1)} p_j. \tag{4}$$

The term $P(2)$ represents the probability of two randomly chosen models simultaneously failing on an episode, while the denominator, $P(1)$, represents the probability of one randomly chosen model failing on an episode. Here, p_j is the probability that j models fail together on a randomly chosen episode. We calculate as $p_j = n_j/K$ where n_j is the total number of episodes that j number of models failed together on the observation set $\mathcal{D}$ and K is the total number of validation episodes. The terms beneath p_j values are the probability of the chosen model being one of the failures. For example, when $S = 3$, there are three cases of model failures; one, two, or three models can fail simultaneously. If one model fails, the chance of selecting the failed model is $1/3$. Similarly, for two models, it is $2/3$, and for three models, it is 1. In the case of minimum diversity, the probability of two randomly chosen models failing together comes down to the probability of one of them failing, which makes the fraction term equal to 1 and $\phi_{\text{div}} = 0$. Similarly, in the case of maximum diversity, there are no simultaneous failures. Hence, the numerator equals 0 and $\phi_{\text{div}} = 1$. The definition of error changes according to the type of task and its solution set y. For the MCQs and OEQs, the errors are inequalities between the prediction of the model and the label; for the GQs, the errors are missed 1-grams between the prediction and the label. Thus, the focal diversity captures member models that are not correlated solely by their error diversity. For each model in the ensemble set, ϕ_{div} is calculated to obtain Focal-Diversity, λ_{div}, using equation 1.

5 Ensemble Pruning with Focal Diversity

Figure 7 shows the surface created by Focal-CKA, Focal-Diversity, and the accuracy of every ensemble combination for a given pool of $N = 6$ base models with MMMU (Yue et al., 2024a). In Figure 6 at Appendix, we plot them separately. We make three observations: (i) the Focal-CKA and Focal-Diversity metric is correlated with the model performance, (ii) there are multiple sub-ensemble teams of size 2-4 that outperform the largest ensemble of size 6, and (iii) a majority of the smaller ensemble teams also outperform the best-performing individual model in the base model pool.

The best combination we need to select is lying at the summit of the surface in Figure 7. However, we need to compute the focal diversity scores for all $2^N - N - 1$ sub-ensemble teams when given

Algorithm 1 Adaptive Epistemic Uncertainty

Require: Uncertainties $\{e_i\}_{i=1}^{N}$, significance threshold α
$\mu, \sigma \leftarrow \text{FitGaussian}(\{e_i\})$ ▷ Fit single Gaussian
$\text{Log}L_1 \leftarrow \sum_{i=1}^{N} \log \mathcal{N}(e_i \mid \mu, \sigma^2)$
$(\pi, \mu_1, \sigma_1, \mu_2, \sigma_2) \leftarrow \text{FitGMM}(\{e_i\})$ ▷ Fit 2-comp. GMM
$\text{Log}L_2 \leftarrow \sum_{i=1}^{N} \log[\pi \mathcal{N}(e_i \mid \mu_1, \sigma_1^2) + (1-\pi)\mathcal{N}(e_i \mid \mu_2, \sigma_2^2)]$
If $\text{Log}L_2 - \text{Log}L_1 > \alpha$
    Predict cluster labels c_i
    $G_0 \leftarrow \{e_i : c_i = 0\}, \quad G_1 \leftarrow \{e_i : c_i = 1\}$
    $\tau \leftarrow \min\{\max(G_0), \max(G_1)\}$
else $t \leftarrow \mu + 2\sigma$
return Threshold t

a pool of N base models. The brute force (BF) approach requires computing the focal diversity for each candidate ensemble of size S ($2 \leq S \leq N$). For $N = 20$, we need to compute the focal diversity score for all $1,048,555$ candidate ensemble teams. To speed up this process, we leverage the Genetic Algorithm (GA) (Mirjalili & Mirjalili, 2019), which significantly reduces the time required to reach the optimal combination (the summit). The process is shown at the third step in Figure 1. Table 1 shows a comparison. We used the weighted combination of focal scores as the fitness function, denoted by $r_{\text{fit}}(\mathcal{E}_i) = w_1\lambda_{CKA} + w_2\lambda_{\text{div}}$, where $w \in [0, 1]$. For a pool of $N = 20$ base models, we complete the focal diversity-based ensemble pruning in under a minute, achieving a 5-order-of-magnitude speedup and 99.66% reduction in latency. Overall, as shown in Figure 2b, the analysis stage scales sup-linearly.

6 Action: Combining VLMs Outputs

As we described in Section 3, there are two types of questions that we aim to ensemble, MCQs and OEQs. We adopt two different ensemble strategies for each one, as shown at the Stage-4 in Figure 1.

For an MC question, following (Gao et al., 2023; Beeching et al., 2023), we aggregate the probabilities of the tokens creating the whole choice to compute the probability of an answer. After repeating the procedure for all the choices, we obtain the probability distribution over the choices, denoted by $\mathbf{q} = [q_1, \ldots, q_m]$, where q represents the probability of a choice and m is the number of choices. Following (Tekin et al., 2024a), we adopt Multi-layer Perceptron (MLP) containing multiple layers of fully connected weights with sigmoid activation functions for fusing outputs of VLMs, which we call *V3Fusion-MLP*. At the final layer, the ensemble performs softmax to produce the output probability: $\tilde{y} = \text{softmax}(\mathbf{W}_H(\ldots\sigma(\mathbf{W}_1[\mathbf{q}_1, \ldots, \mathbf{q}_N])\ldots))$, where H is the number of layers. The first layer takes the concatenation of the probabilities as the input, i.e., $\mathbf{W}_1 \in \mathbb{R}^{(mN)\times d}$ where d is the input dimension of the second layer. We train the model by decreasing the cross-entropy loss $\mathcal{L}_{\text{vote}}(y, \tilde{y}) = -\sum_{i=1}^{m} y_i \log(\tilde{y}_i)$ using SGD and dataset $\mathcal{D}$.

For an OE question, we design the V3Fusion learn-to-ensemble by implementing Longformer Encoder-Decoder (LED) (Beltagy et al., 2020) training method (V3Fusion-LED for short) with two motivations. First, V3Fusion-MLP is not applicable without relaxing the definition of equality between different solutions and it is not generative. Second, our goal with V3Fusion-LED is to create an ensemble learner that applies to all types of tasks and generates its own output (Tekin et al., 2024a). V3Fusion-LED is a sequence-to-sequence (seq2seq) model with encoder-decoder architecture (Jiang et al., 2023b) by concatenating the outputs of the component models of a chosen ensemble of S base models with the input query and generating the final output of V3Fusion. It exploits sparse attention and global attention such that we can increase the context length up to 16396 tokens with 149 million parameters, and utilize a small training dataset. We give the details of the model in Appendix C.

7 Verification & Rectification

There are two types of uncertainties one can model, *aleatoric* (data) uncertainty and *epistemic* (model) uncertainty (Kendall & Gal, 2017). In this section, we leverage the representativeness of multiple VLMs to calculate the system's epistemic uncertainty. Bayesian Neural Networks (BNNs) (Blundell et al., 2015; Graves, 2011) and DEEP ENSEMBLES (Lakshminarayanan et al., 2017) have shown the standard approaches to uncertainty decomposition. Following (Malinin & Gales, 2018), the model uncertainty can be described by posterior distribution over the model parameters $p(\theta|\mathcal{D})$.

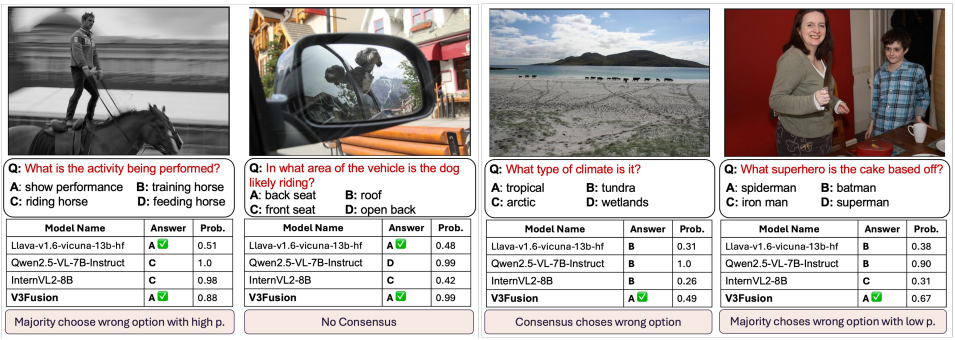


Figure 3: Four examples are used to illustrate the superior performance of V3Fusion compared to existing popular methods. It demonstrates cases where V3Fusion can achieve the correct output, even when the base models fail to reach a consensus or agree on an incorrect option.

Then, Mutual Information (MI) $I[y, \theta|\mathbf{x}]$ between the categorical label y and parameters of the model θ measures the dependency between the predicted label and the model parameters, reflecting the variability among ensemble members $\{P(y|\mathbf{x}, \theta^{(i)})\}_{i=1}^{N}$ which assess uncertainty in predictions due to model uncertainty. Formally one can calculate the MI by the difference between total uncertainty data uncertainty:

$$\underbrace{I[y, \theta|\mathbf{x}]}_{\text{Epistemic}} = \underbrace{\mathcal{H}[\mathbb{E}_{p(\theta|\mathcal{D})}[P(y|\mathbf{x}, \theta)]]}_{\text{Total Uncertainty}} - \underbrace{\mathbb{E}_{p(\theta|\mathcal{D})}[\mathcal{H}[P(y|\mathbf{x}, \theta)]]}_{\text{Aleatoric Uncertainty}} \tag{5}$$

where $\mathcal{H}$ represents the entropy among the logits of the models and here we model the total uncertainty by calculating the entropy of the ensemble model $\mathcal{H}_{ens}[P(y|\mathbf{x}, \theta)]$ and aleatoric uncertainty by taking the average of base model entropy scores. The epistemic uncertainty indicates that when the model is unsure, it hasn't learn the data distribution well, and eventually will help us to prevent hallucination. Unlike the literature, we model epistemic uncertainty using heterogeneous models with diverse architectures and vocabularies. This variety reflects the true disagreement and better approximates a posterior over models to capture the knowledge gaps among base models more effectively.

An unanswered question is how to select the uncertainty threshold τ dynamically. The incoming data to the system can comprise both uncertain and certain samples, resulting in two distributions that we observe are centered at different positions. Therefore, we apply the likelihood ratio test to two probability density models and select the one that best represents the observed epistemic uncertainty data (Bishop & Nasrabadi, 2006; Burnham & Anderson, 2004; Anonymous, 2025). The algorithm 1 illustrates adaptive threshold selection, where we either select a two-component Gaussian Mixture Model or a single Gaussian, depending on the significance threshold α, which can be 10 corresponding to 0.01 p-value. We reject the samples that have higher Epistemic Uncertainty than the threshold value τ and and accept the ones that have lower. For the rejected samples, we predict by averaging all the base model logits and return the maximum choice as the final output. For the open-ended questions, we select the majority answer if it is satisfied, else we select randomly. This way our methodology can decide on the final answer without the need of external system or supervision making it self-sustained closed system.

8 Experiments

We validate the effectiveness of V3Fusion through extensive evaluations on benchmarks with two question types, multiple-choice and open-ended. We demonstrate that V3Fusion outperforms both the best individual models and fine-tuned VLMs for the ensemble task. Surprisingly, our V3Fusion works even in cases where the majority of the candidate models make errors. Lastly, we provide ablation studies to show the gain from each stage and diversity metric. We run experiments on four popular VQA benchmark datasets: MMMU(Yue et al., 2024a), MMMU-Pro (Yue et al., 2024b), A-OKVQA (Schwenk et al., 2022), and OCR-VQA (Mishra et al., 2019). All experimental setup details are in Appendix-D.

Performance of V3Fusion: Table 2 presents experiments on datasets, comparing the scores of each base model in the pool with those of the ensemble learners V3Fusion-MLP and V3Fusion-LED, plus their rectified version with V3Fusion-Rectify. The Model IDs of V3Fusion denote the models in the

Table 2: Accuracy of the base models and V3Fusion on A-OKVQA, MMMU, MMMU-Pro, and OCR-VQA. Ensemble sets are selected by the combined focal-diversity + Focal-CKA pruning objective. **Abs. Gain** measured for V3Fusion-Rectify against the best base model.

Model Name	Model ID	A-OKVQA†	MMMU (Val)*	MMMU-Pro‡	OCR-VQA**		
		Acc (%)↑	Acc (%)↑	Acc (%)↑	BLEU-1↑	EM (%)↑	F1(%)↑
LlaVA-v1.6-Vicuna-13b	0	83.4	37.02	33.69	78.07	67.71	79.60
LlaVA-v1.6-Vicuna-7b	1	82.62	35.9	32.81	78.93	69.17	80.44
DeepSeek-VL2-Tiny	2	76.68	39.01	34.61	38.30	29.32	41.71
DeepSeek-VL2-Small	3	80.08	39.75	38.00	54.60	41.71	57.25
Qwen2.5-VL-7b-Ins.	4	87.24	51.55	46.98	83.34	72.00	84.80
Intern-VL2-8b	5	85.32	51.30	43.09	52.36	45.55	57.09
V3Fusion-MLP	$123^{\dagger} \mid 235^{*} \mid 234^{\ddagger}$	88.31	$55.07^{\S}$	47.34	-	-	-
V3Fusion-LED	$123^{\dagger} \mid 235^{*} \mid 234^{\ddagger} \mid 12^{**}$	87.86	51.30	46.26	**86.24**	71.91	**86.82**
V3Fusion-Rectify	$123^{\dagger} \mid 235^{*} \mid 234^{\ddagger} \mid 12^{**}$	**89.17**	**56.09**	**49.27**	85.71	**72.08**	86.57
Abs. Gain (%)	-	+1.93	+4.54	+2.29	+2.37	+0.08	+1.77

Table 3: V3Fusion against training-free aggregation, LLM-based multi-agent collaboration, and routing/selection baselines on the multiple-choice benchmarks

Method	MMMU	MMMU-Pro	A-OKVQA
	Acc (%)↑	Acc (%)↑	Acc (%)↑
Reference points			
Strongest base member	51.43	46.26	88.30
Oracle (upper bound)	*86.21*	*82.02*	*96.51*
Training-free aggregation			
Majority Voting	47.58	44.12	87.86
Plurality Voting	54.95	47.53	85.76
Self-consistency (k=6)	46.66	40.25	84.12
LLM-based multi-agent collaboration			
LLM-Aggregator (gpt-4o-mini)	46.21	40.48	86.72
MoA (gpt-4o-mini, 7 calls/q)	47.58	41.48	87.95
MAgICoRe (verify+refine)	46.09	40.79	87.77
Routing and selection			
RouteLLM (Random)	26.45	44.50	68.90
RouteLLM (SWRank)	30.68	46.64	81.83
PairRanker (LLM-Blender)	53.29	48.02	85.76
Symbolic-MoE (routing + LLM agg.)	54.53	45.00	88.47
Ours			
V3Fusion-MLP	55.07	47.34	88.31
V3Fusion-Rectify	**56.09**	**49.27**	**89.17**

Table 4: Gains obtained by each phase for A-OKVQA, MMMU, and MMMU-Pro datasets.

Phases	A-OKVQA	MMMU	MMMU-Pro
Pruning Only	+1.07	+1.36	−1.79
Fusion Only	−1.48	+3.4	+0.55
Pruning+Fusion	+1.07	+3.52	+0.36
Prun.+Fusi.+Rect.	**+1.93**	**+4.54**	**+2.29**

Table 5: V3Fusion against the baselines that produce free-form answers, on OCR-VQA

Method	BLEU-1↑	EM (%)↑	F1 (%)↑
LLM-Blender (Pair R., off-the-shelf)	49.93	30.58	43.95
LLM-Aggregator (gpt-4o-mini)	76.92	69.88	79.82
RouteLLM (Random)	80.73	69.81	82.23
RouteLLM (SWRank)	81.69	70.94	83.27
Self-consistency (k=6)	83.75	**72.17**	85.21
V3Fusion-LED	**86.24**	71.91	**86.82**
V3Fusion-Rectify	85.71	72.08	86.57

ensemble set, which is selected by ensemble pruning. In the MCQ tasks (MMMU, MMMU-Pro, and A-OKVQA), V3Fusion-MLP reaches the best performance by surpassing the best-performing model Qwen2.5-7B by 1.93%, 2.29%, and 4.54%. Whereas, in MCQA tasks, V3Fusion-LED shows the same performance as the best-performing model. The reason is that when V3Fusion-MLP is making a decision, it considers the probabilities assigned to the candidate outputs and observes the pattern for each candidate model. As shown in Figure 3, V3Fusion-MLP can reach the correct answer when the majority or even all the models are giving an incorrect decision. For example, in the third sample in 3, models show consensus at choice B, but the V3Fusion selected the correct decision. When we analyze the probabilities of choices assigned by the models, we observe that Llava-1.6v assigned 0.29 probability to option A. This indicates that, despite Llava-1.6 being the third-best performing model, it contributes valuable information that the V3Fusion model leverages to achieve the correct final decision. On the other hand, the V3Fusion-LED model performs learnable consensus through summarization, which requires reasoning and thinking to generate high-quality output. Hence, it is more suitable for open-ended VQA tasks such as OCR-VQA, where it improves the best model's F1 score by 1.77%. V3Fusion-LED successfully detects the information gathered by each base model and generates the correct output. Each model has its own expertise due to its training dataset's coverage and its learning capabilities, which the visual and language encoding models facilitate. V3Fusion-LED can combine answers from multiple VLMs via learn-to-combine summarization by exploiting the complementary wisdom of individual base models of an ensemble.

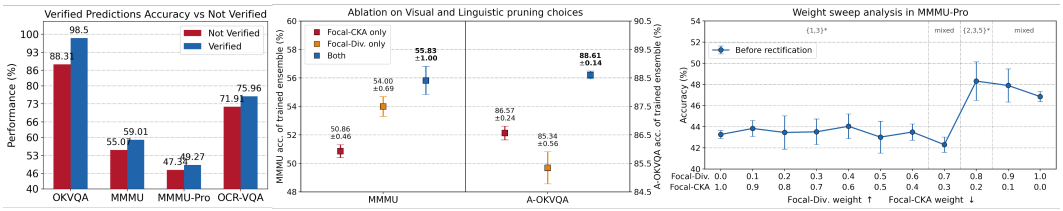


Figure 4: The first figure on the left shows the average accuracy of the verified predictions for each dataset compared to the non-verified predictions. The middle figure shows the performance of trained ensemble models when the pruning stage uses only Focal-CKA, Focal-Div, and both. The rightmost figure shows the weight-sweep across weights of Focal-Div and Focal-CKA and its performance on MMMU-Pro before rectification. The bars show the standard errors.

Performance Against Prior Work: We further evaluate V3Fusion by comparing it with nine different lines of work in Table 3 categorized under training-free aggregation, multi-agent collaboration, and routing and selection. Details of each work are in Appendix-D.1. In all the datasets, V3Fusion reaches the best performance, and compared to the theoretical upper bound, Oracle, there is a 7.34% gap in A-OKVQA. While the other methods consider only the language output to decide on the best model selection, V3Fusion utilizes the visual modality in model selection and learns to combine the wisdom of each model.

Ablation Studies: We observe the gains from stages 3, 4, and 5 by conducting an ablation study in Table 4. The gains from pruning-only (with plurality voting as the final decision maker) and fusion-only (without pruning, i.e., all 6 models combined with the MLP) are inconsistent. Contrarily, V3Fusion with Pruning and Fusion stages provides a gain on all three datasets, and Rectification further improves the performance. The middle plot in Figure 4 shows the gain by each focal metric, and the performance of the ensemble model maximizes when both metrics are enabled. This supports the idea that both visual and linguistic diversity create a more accurate VLM set and is the reason why V3Fusion is better than baseline approaches. However, mixing weights in the fitness function requires a sweep on the validation dataset, as shown by the rightmost plot in Figure 4. Therefore, we run an optimality analysis in Appendix G.1, showing that even with a high model pool $N = 12$, the GA converges to the optimum point less than a second. We perform additional ablation studies in Appendix-F, where we compare diversity metrics with other error correlation metrics. Also, we provide an efficiency analysis regarding the latency and data sensitivity of V3Fusion in Table 9 and 10 in the Appendix-G. A further analysis for comparison between fine-tuned base models is given in Appendix-F.1, and we test the robustness of V3Fusion in different hardware with a one-sample t-test in Appendix-G.2.

Boost by the Rectification Stage: V3Fusion-Rectify boosts the performance even further in each dataset and obtains the highest score. The first plot on the left of Figure 4 shows the accuracy comparison between the verified samples by the Adaptive Epistemic Uncertainty algorithm. Almost all the samples that are verified in the OK-VQA dataset are correct, and we observe 4-5% boost in other datasets. We observe minor improvements in the results of the MLP and LED models with rectification, indicating that the rejected samples are significantly more challenging and require larger models to handle. The illustration of threshold selection process is provided in Appendix-H.

9 Conclusion

We have introduced V3Fusion, a novel framework for diversity fusion of vision-language models (VLMs). This paper makes three original contributions. First, our V3Fusion framework develops the error diversity metrics to capture the complimentary wisdoms of individual component VLMs with respect to erroneous inference results. By leveraging our focal error diversity measures, we construct a candidate ensemble surface for a pool of candidate VLMs, and apply a Genetic Algorithm to effectively prune out those candidate VLMs that do not add value to the performance improvement of our V3Fusion framework. Second, we develop a holistic learning approach to combining and resolving output inconsistencies among the VLMs of a VLM fusion team by training a V3Fusion model. Finally, we calculated the epistemic uncertainty of the system to reject or accept the generated samples with a dynamic adaptive threshold. We conduct extensive experiments to validate V3Fusion on four popular vision-language benchmarks and observe that V3Fusion outperforms the best-performing VLMs.

AI USE STATEMENT

In this work, we did not use generative AI tools for any of the tasks with required disclosure. Specifically, we have not used generative AI tools to generate synthetic datasets, develop theoretical models or conceptual frameworks, formulate mathematical claims or assist with proofs, 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, limited to checking grammar, spelling, and phrasing in text 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 content, 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. 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.

REPRODUCIBILITY STATEMENT

We have made several efforts to ensure the reproducibility of our results. The complete source code for this work is available through an anonymous repository, linked in the abstract. The repository includes step-by-step instructions for setting up the environment and running our experiments, along with instructions for downloading all datasets used in this paper, each of which is publicly available and open source. Details of our experimental setup, including model configurations and hyperparameters, are described in Section 8 and Appendix D.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Lorenzo Agnolucci, Alberto Baldrati, Francesco Todino, Federico Becattini, Marco Bertini, and Alberto Del Bimbo. Eco: Ensembling context optimization for vision-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2811–2815, 2023.

Lisa Alazraki, Lluis Castrejon, Mostafa Dehghani, Fantine Huot, Jasper Uijlings, and Thomas Mensink. How (not) to ensemble lvlms for vqa. In *Proceedings on*, pp. 1–20. PMLR, 2023.

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

Anonymous. Gradshield: Alignment preserving finetuning. In *Submitted to The Fourteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=YYUNm7IibC`. under review.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Edward Beeching, Clémentine Fourrier, Nathan Habib, Sheon Han, Nathan Lambert, Nazneen Rajani, Omar Sanseviero, Lewis Tunstall, and Thomas Wolf. Open llm leaderboard. `https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard`, 2023.

Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pp. 1613–1622. PMLR, 2015.

Leo Breiman. Heuristics of instability and stabilization in model selection. *The annals of statistics*, 24(6):2350–2383, 1996.

Kenneth P Burnham and David R Anderson. Multimodel inference: understanding aic and bic in model selection. *Sociological methods & research*, 33(2):261–304, 2004.

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.

Boyu Chen, Zhengrong Yue, Siran Chen, Zikang Wang, Yang Liu, Peng Li, and Yali Wang. Lvagent: Long video understanding by multi-round dynamical collaboration of mllm agents. *arXiv preprint arXiv:2503.10200*, 2025a.

Justin Chen, Archiki Prasad, Swarnadeep Saha, Elias Stengel-Eskin, and Mohit Bansal. Magicore: Multi-agent, iterative, coarse-to-fine refinement for reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 32651–32674, 2025b.

Justin Chih-Yao Chen, Sukwon Yun, Elias Stengel-Eskin, Tianlong Chen, and Mohit Bansal. Skill-based mixture-of-experts: Adaptive routing for heterogeneous reasoning via inferred skills. *arXiv preprint arXiv:2503.05641*, 2025c.

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.

Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, Ji Ma, Jiaqi Wang, Xiaoyi Dong, Hang Yan, Hewei Guo, Conghui He, Botian Shi, Zhenjiang Jin, Chao Xu, Bin Wang, Xingjian Wei, Wei Li, Wenjian Zhang, Bo Zhang, Pinlong Cai, Licheng Wen, Xiangchao Yan, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites, 2024a. URL https://arxiv.org/abs/2404.16821.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks, 2024b. URL https://arxiv.org/abs/2312.14238.

Zhi Chen, Xin Yu, Xiaohui Tao, Yan Li, and Zi Huang. Cluster-aware prompt ensemble learning for few-shot vision–language model adaptation. *Pattern Recognition*, pp. 112596, 2025d.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Algorithms for learning kernels based on centered alignment. *The Journal of Machine Learning Research*, 13(1):795–828, 2012.

Thomas G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pp. 1–15. Springer, 2000.

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*, 2023.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 12 2023. URL https://zenodo.org/records/10256836.

Alex Graves. Practical variational inference for neural networks. *Advances in neural information processing systems*, 24, 2011.

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.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

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.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision, 2021. URL https://arxiv.org/abs/2102.05918.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023a.

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*, 2023b.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.

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.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation, 2022. URL https://arxiv.org/abs/2201.12086.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023. URL https://arxiv.org/abs/2301.12597.

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

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36, 2024b.

Wenzhe Li, Yong Lin, Mengzhou Xia, and Chi Jin. Rethinking mixture-of-agents: Is mixing different large language models beneficial? *arXiv preprint arXiv:2502.00674*, 2025.

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.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

Zhihe Lu, Jiawang Bai, Xin Li, Zeyu Xiao, and Xinchao Wang. Beyond sole strength: Customized ensembles for generalized vision-language models. *arXiv preprint arXiv:2311.17091*, 2023.

Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks. *Advances in neural information processing systems*, 31, 2018.

Potsawee Manakul, Adian Liusie, and Mark Gales. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 9004–9017, 2023.

Seyedali Mirjalili and Seyedali Mirjalili. Genetic algorithm. *Evolutionary Algorithms and Neural Networks: Theory and Applications*, pp. 43–55, 2019.

Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. Ocr-vqa: Visual question answering by reading text in images. In *2019 international conference on document analysis and recognition (ICDAR)*, pp. 947–952. IEEE, 2019.

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.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021a. URL `https://arxiv.org/abs/2103.00020`.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021b.

Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *Advances in neural information processing systems*, 34:12116–12128, 2021.

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.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

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.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

Selim Furkan Tekin, Fatih Ilhan, Tiansheng Huang, Sihao Hu, and Ling Liu. Llm-topla: Efficient llm ensemble by maximising diversity. *arXiv preprint arXiv:2410.03953*, 2024a.

Selim Furkan Tekin, Fatih Ilhan, Tiansheng Huang, Sihao Hu, Zachary Yahn, and Ling Liu. H3fusion: Helpful, harmless, honest fusion of aligned llms. *arXiv preprint arXiv:2411.17792*, 2024b.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Khanh-Tung Tran, Dung Dao, Minh-Duong Nguyen, Quoc-Viet Pham, Barry O'Sullivan, and Hoang D Nguyen. Multi-agent collaboration mechanisms: A survey of llms. *arXiv preprint arXiv:2501.06322*, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

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.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Y Zou. Mixture-of-agents enhances large language model capabilities. In *International Conference on Learning Representations*, volume 2025, pp. 33944–33963, 2025.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Zhiyu Wu, Xiaokang Chen, Zizheng Pan, Xingchao Liu, Wen Liu, Damai Dai, Huazuo Gao, Yiyang Ma, Chengyue Wu, Bingxuan Wang, et al. Deepseek-vl2: Mixture-of-experts vision-language models for advanced multimodal understanding. *arXiv preprint arXiv:2412.10302*, 2024.

Zhiqiu Xia, Jinxuan Xu, Yuqian Zhang, and Hang Liu. A survey of uncertainty estimation methods on large language models. *arXiv preprint arXiv:2503.00172*, 2025.

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, 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.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training, 2023. URL `https://arxiv.org/abs/2303.15343`.

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.

Shaolei Zhang, Tian Yu, and Yang Feng. Truthx: Alleviating hallucinations by editing large language models in truthful space. *arXiv preprint arXiv:2402.17811*, 2024.

Ruochen Zhao, Wenxuan Zhang, Yew Ken Chia, Weiwen Xu, Deli Zhao, and Lidong Bing. Autoarena: Automating llm evaluations with agent peer battles and committee discussions. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4440–4463, 2025.

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 Diverse Capabilities of VLMs

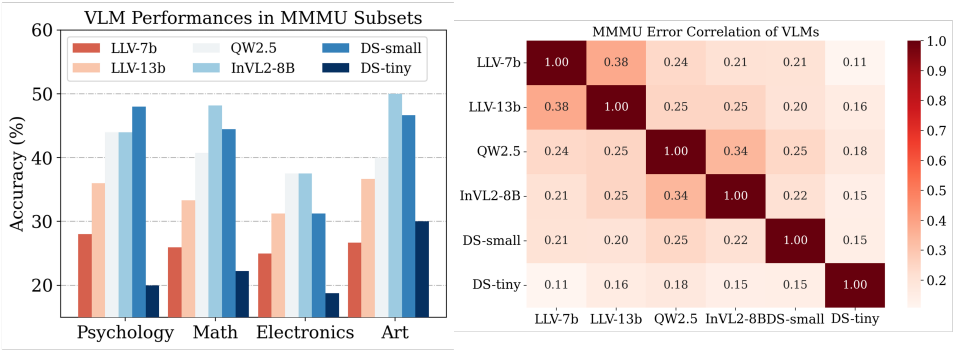


Figure 5: Performance of open-sourced and most popular VLMs in various tasks and their error correlation in MMMU.

Figure 5 shows an experimental proof on the diverse capabilities of VLMs, which caused by their structural differences such as, architectures, vocabularies, hidden sizes, training corpora and their instruction tuning strategies. As shown in the bar plot on the left of Figure 5, the performances of different VLMs vary in different categories of college subjects. For example, while Deepseek-small is the best performing model in Psychology, it performs third-best in Electronics. A similar phenomenon can be observed for the other models, such as InternVL2 and Qwen. As our second insight, we observe that overall error correlation among these models is less. The ones with the high error correlations are within the models that have the same architectures, as shown in the correlation plot in Figure 5, indicating redundancy. On contrary, the correlations between models that have different architectures are low, highlighting their importance in an ensemble pool. Based on this motivation, we aim for a mechanism that will select candidate models derived from their error correlations, while motivating their individual expertise.

C More Details for Fusing MCQs and OEQs

For an MC question, as proposed in Hendrycks et al. (2020), the probabilities assigned to choices are obtained by calculating the probability of the choice's token using the next-token probability softmax equation. For instance, $p(w_t = \mathrm{A}|x, w_{<t})$ is calculated for choice A. However, a more popular methodology proposed by Gao et al. (2023) is used by the HuggingFace Leader Board Beeching et al. (2023) and also in our paper. We aggregate the probabilities of the tokens creating the whole choice to compute the probability of an answer. After repeating the procedure for all the choices, we obtain the probability distribution over the choices, denoted by $\mathbf{q} = [q_1, \ldots, q_m]$, where q represents the probability of a choice and m is the number of choices.

As objective at Fusion Model, we want to find the best parameters $\theta = (\mathbf{W}_1, \ldots, \mathbf{W}_H)$ to maximize the likelihood, which can be reduced to minimizing the cross-entropy loss on a dataset, which is the collection of probabilities for each component model. Thus, we split the dataset into training, validation, and test sets, use the training set to train the ensemble model, the validation set to stop training, and finally use the test set to calculate performance. In each iteration of training, the

parameters are updated by minimizing the cross-entropy loss function:

$$\begin{aligned} \theta_{\text{best}} &= \underset{\theta}{\operatorname{argmin}} \sum_{x,y \in \mathcal{D}^{\text{train}}} \mathcal{L}_{\text{vote}}(y, \tilde{y}), \\ \tilde{y} &= f(\mathcal{M}_1(x), \ldots, \mathcal{M}_N(x); \theta), \\ \mathcal{L}_{\text{vote}}(y, \tilde{y}) &= -\sum_{i=1}^{m} y_i \log(\tilde{y}_i). \end{aligned} \tag{6}$$

We use SGD to update the parameters at every iteration. The ensemble learner analyzes the probabilities assigned by each model and their confidence level. Thus, we train the learn-to-ensemble model to recognize patterns across the predictions of each component model efficiently. This allows the ensemble learner to learn to make the correct choice even in the absence of consensus, rather than blindly relying on consensus voting algorithms such as majority or plurality voting.

For a OE question, recall, $y^{(2)} = \{\omega_1, \ldots, \omega_L\}$, where L is the sequence length of the desired output. Each model in the pool generates the predicted sequence denoted by $\mathcal{M}_i(\mathbf{x}, \mathbf{I}) = \{\hat{\omega}_1, \ldots, \hat{w}_{T_i}\} = z_i$ and L_i is the sequence length of the i^{th} model output which can be different than L. Let $\mathcal{Z} = \{z_1, \ldots, z_N\}$ be the collection of candidates, VL3Fusion-LED models the conditional probability of $p(y|\mathbf{x}, \mathcal{Z})$. We give the input sequence, $\mathbf{x}_s$, to the seq2seq model in the format of $\mathbf{x}_s = \text{concat}(x, z_1, \ldots, z_N)$ and use special tokens as separators to indicate the beginning and end of the question or an answer. Consider an ensemble from 3 base models, the input below is sent to the V3Fusion-LED model:

$$\begin{aligned} x_s = &< \text{boq} > x < \text{eoq} > < \text{boc1} > z_1 < \text{eoc1} > \\ &< \text{boc2} > z_2 < \text{eoc2} > < \text{boc3} > z_3 < \text{eoc3} >. \end{aligned} \tag{7}$$

We use distinct tokens to indicate which model each candidate belongs to. To reduce complexity and increase context length, we utilize the *sliding window attention* pattern proposed by Beltagy et al. (2020). To stress the relation between the question and each candidate's answer, we employ *selective global attention* on the tokens of x of the input question. The global attention is the standard self-attention by scoring each token against every other token. With the sliding and global attention mechanism, we increase the context window length, reduce the computational complexity, and improve the performance.

Overall, V3Fusion-LED is optimized by finding the best model parameter ϕ that will maximize the joint distribution over the target tokens $p(y|x, z_1, \ldots, z_N; \phi)$. It performs auto-regressive generation using the following cross-entropy loss $\mathcal{L}_{sum} = -\sum_{t=1}^{T} \log p(w_t|w_{<t-1}, x, \mathcal{Z}; \phi)$ We use SGD to perform updates on the parameters in each iteration. As the V3Fusion-LED model is trained, it learns to generate the correct token sequence by utilizing the information provided by each candidate answer and the V3Fusion-LED evaluation results.

D Experimental Setup

Datasets. The experiments are performed on four popular VQA benchmark datasets: MMMUYue et al. (2024a), MMMU-Pro Yue et al. (2024b), A-OKVQA Schwenk et al. (2022), and OCR-VQA Mishra et al. (2019). The first three are VQA datasets 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. To perform training on fusion models, we used MMMU-Pro to train the model and evaluate it on MMMU validation data. For the MMMU-Pro, we used MMMU validation data and tested it on MMMU-Pro test data. There are 577 common samples between the MMMU-Pro test data and the MMMU-val data. We remove the intersections from the train data. 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. For the A-OKVQA dataset, we use 3,000 samples from its training set to train our V3Fusion models and 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.

The OCR-VQA dataset Mishra et al. (2019) consists of book cover images, each paired with four open-ended questions for OCR tasks. For example, the question *What is the title of the book?* requires the model to recognize the text in the image and provide the answer in text form. Therefore, the answer is open-ended. There are 166,202 training pairs, 20,731 validation samples, and 20,796 test samples. V3Fusion-LED can perform reasoning effectively for such open-ended VQA tasks.

Evaluation Metrics. To evaluate the performance of VLMs and V3Fusion methods, we use accuracy on MMMU, MMMU-Pro and A-OKVQA datasets. For the open-ended OCR-VQA dataset, we use BLUE-1, Exact Match (EM), and F1 scores to evaluate and compare the performance of individual VLMs and V3Fusion methods.

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.

V3Fusion Training Models/Algorithms. V3Fusion-MLP model contains two fully connected hidden layers with 100 neurons and ReLU activations between the layers. The model weights start from Xavier initialization and converge in 500 epochs optimized by Adam. To implement V3Fusion-LED, we employ the Longformer-Encoder-Decoder Beltagy et al. (2020) model, which is initialized from BART weights Lewis et al. (2019). The model contains additional positional encoding matrices on top of BART weights, with a total of 161 million parameters. Since not every question contains the same number of choices, we pad the generated probability vectors from each model to match the maximum number of choices.

Ensemble Pruning. We selected $w_1 = 0.5$, and $w_2 = 0.5$ while scoring a candidate ensemble set with the fitness function to give equal importance to Focal-CKA and Focal-Diversity in multiple-choice type datasets and open-ended datasets such as OCR-VQA. The genetic algorithm stops when the fitness function does not change for 100 consecutive generations.

D.1 Baseline approaches

We compared V3fusion with the following frontier multi-agent baselines and standard aggregation baselines:

- **Self-consistency (Self-MoA)**(Wang et al., 2022; Li et al., 2025): Self-MoA revisits mixture-of-agents and finds that aggregating multiple outputs sampled from only the single strongest model often matches or beats mixing different models, making it a strong test of whether model diversity actually helps. Our variant samples the best single non-degenerate base VLM's stored per-question output distribution k=6 times and majority-votes, compute-matched to the ensemble size, isolating whether gains come from model diversity or just extra inference calls.
- **PairRanker** (LLM-Blender-style selection)(Jiang et al., 2023b): PairRanker is the ranking module of the LLM-Blender ensembling framework, which compares candidate outputs from different models pairwise and selects the best one. We use a small ranker trained in-domain directly on our stored candidate outputs (same train/val/test convention as V3Fusion-MLP), addressing the off-the-shelf LLM-Blender checkpoint's out-of-domain failure.
- **LLM-Aggregator**: This is a simple generative fusion baseline in which a single strong LLM acts as the judge and synthesizer for the whole ensemble. One LLM (gpt-4o-mini by Achiam et al. (2023)) reads all candidate answers and produces a final answer, zero-shot.
- **MoA** (Wang et al., 2025): Mixture-of-Agents builds a layered architecture in which each agent receives all outputs of the previous layer as auxiliary context and refines them, with a final aggregator producing the answer. We use 2 proposer layers of 3 agents + a final aggregation layer (7 LLM calls/question).
- **Symbolic-MoE** (Chen et al., 2025c): Symbolic-MoE is a skill-based mixture-of-experts framework that infers the skills each question requires and recruits the experts with the highest skill-specific competence, then aggregates their outputs. In our implementation, a MiniLM question embedding stands in for the paper's LLM-tagged skill labels, routing each

question to its top-3 most competent base models (trained MMMU-Pro→MMMU, same convention as our own fuser) before LLM aggregation.

- **MAgICoRe** (Chen et al., 2025b): MAgICoRe is a multi-agent, coarse-to-fine refinement framework that spends extra compute only on hard problems, applying simple aggregation to easy ones and iterative review-and-refine rounds to difficult ones. Starting from a free plurality vote, a verifier LLM scores confidence; low-confidence ("hard") questions get up to 2 refine-then-reverify rounds, high-confidence ones are accepted after one call, adaptive compute instead of a fixed pipeline.

E Correlation Analysis on Focal Metrics

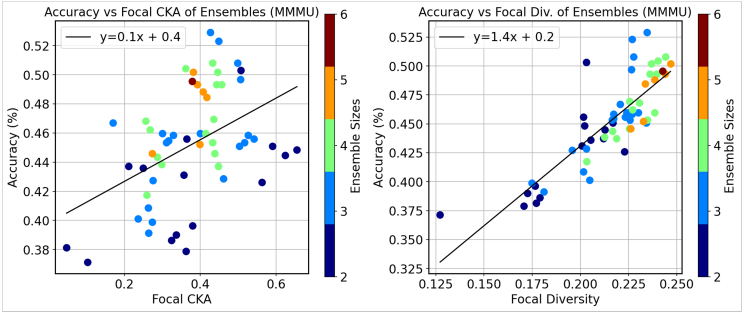


Figure 6: The Correlation between the Accuracy of the ensemble combinations with different sizes and the Focal-CKA on left, and Focal-Diversity on Right.

In section 5, we presented the ensemble pruning metrics to allow us to select the best ensemble set in terms of performance, which is measured by accuracy. A more direct analysis is to compare performance against diversity. The diversity we aim for has two dimensions: visual perception and language semantics. We capture the first with Focal-CKA with relative dissimilarity among perceptual understanding in an ensemble set. The second is captured by the Focal-diversity, focusing on the output generated by the language decoder. Therefore, in Figure 6, we plot all the possible ensemble set combinations with different sizes and measure their Accuracy and Focal metrics when they team up. In both metrics, as shown in Figure 6, the performance is positively correlated with the diversity metrics. The best-fit line with a positive slope shows the degree of correlation. Overall, the Figure 2b is the surface created by both metrics, and the pruning algorithm (boosted by GA) finds the best combination on that surface.

F Additional Ablation Studies

Table 6: Focal diversity score against 5 other pairwise error correlation scores (MMMU, OKVQA).

Dataset	Fleiss	Corr. Coef	Bin. Disag.	Kappa	Bin. Entor.	Focal-Div.
MMMU	37.14	37.14	42.61	37.14	50.31	**54.50**
OKVQA	82.79	82.79	79.04	82.79	76.77	**88.31**

Table 6 shows scores of V3Fusion when we use popular pairwise error correlation metrics at the pruning stage. Compared to Focal-Diversity, these metrics underperform on both datasets. Therefore, Focal-Diversity is more capable of distinguishing the best model combination for given tasks.

To further observe the effect of the pruning and attention mechanisms, we execute two ablation studies in Figure 7. First, we ensemble all the models in the pool and compare their performances with the ensemble model selected by the pruning mechanism. As shown in both plots, the pruning improves the performance of V3Fusion-MLP and -LED in every task. Although there is less improvement in the V3Fusion-LED model in OCR-VQA, the pruned ensemble set is reaching a slightly better performance with fewer models. Given that the models are at a billion-scale parameter size, our approach significantly reduces computational redundancy compared to selecting all models.

F.1 FINETUNING BASE MODELS

Table 7: Compare V3Fusion with the supervised fine-tuned (SFT) VLMs.

Method	ID	OCR-VQA		
		BLEU-1↑	EM (%)↑	F1↑
LlaVA-v1.5-Vicuna-7b (SFT)	123456	69.25	58.98	71.24
Qwen2-7B-VL-Instruct (SFT)	123456	84.99	75.27	86.49
Qwen2.5-7B-VL-Instruct (SFT)	123456	85.65	75.57	87.04
V3Fusion-LED	456	**87.79**	74.91	**88.83**

We also compare the performance of finetuned base models and V3Fusion-LED in the generative tasks in Table 7, using the same training data. We select Qwen2.5-7B, Qwen2-7B, and Llava-v1.5 as visual-language models to ensemble the candidate outputs generated by the base models. We used the same prompt, incorporating candidate model outputs for the Visual-Question pair, and fine-tuned the model using supervised fine-tuning (SFT) with LoRA. Training was stopped after three consecutive steps without a validation performance improvement. As shown in Table 7, even though V3Fusion-LED uses a smaller number of parameters, it outperforms fine-tuned versions of the base models. Particularly, V3Fusion-LED has a shorter training time (35 mins) compared to VLM training (1.47 hours for Qwen2.5-7B). This occurs because these models use an image encoder and text decoder architecture designed to learn image-text relationships. As a result, fine-tuning them for summarization is less effective than using sequence-to-sequence models. We additionally fine-tune the base models individually (without the ensemble objective) and compare their performance and training time on OCR-VQA and MMMU against V3Fusion Table 8. V3Fusion showed significantly high performance in MMMU with half the training time.

Table 8: SFT best performing base models vs V3Fusion in OCR-VQA and MMMU datasets.

Train Data (%)	Train. Time(h)	OCR-VQA			MMMU
		BLUE-1	EM	F1	Acc (%)
llava-v1.6-vicuna-7b-hf	4.40	50.27	30	53.88	29.06
llava-v1.6-vicuna-13b-hf	7.02	60.23	44	63.98	30.55
Qwen2.5-VL-7B-Instruct	2.17	85.12	**74.89**	86.5	27.95
V3Fusion	1.39	**86.24**	71.91	**86.82**	**56.09**

A more direct way to measure the gain by ensemble learning is to compare with the individual model performances when they are finetuned and evaluated with the same datasets. The base models, however, have a significant amount of parameters compared to V3Fusion's summary model (LED) and MCQ model (MLP). The training time, therefore, is 7× longer than V3Fusion as reflected in Table 8. V3Fusion improves in both datasets, OCR-VQA and MMMU, with the highest F1 and Accuracy scores and lower training time. Especially in the MMMU task, V3Fusion shows 25.54% higher accuracy than the best performing finetuned base model.

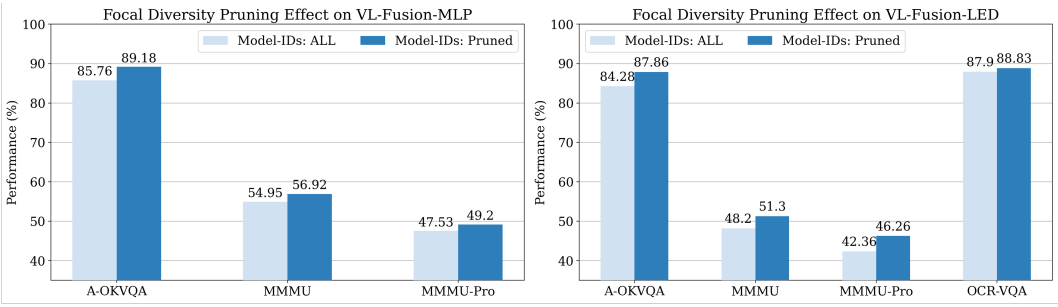


Figure 7: The effect of Focal-diversity Pruning on V3Fusion-MLP and V3Fusion-Weighted is shown for all the datasets.

G EFFICIENCY ANALYSIS

Table 9: Inference time and allocated/consumed memory comparison with single VLMs in the model pool.

Model Name	ID	Inf.	Mem. (MB)	
		Mean(ms)	Alloc.	Cons.
llava-v1.6-Vic.-7b	0	350.71	13472.43	7.43
llava-v1.6-Vic.-13b	1	511.25	25587.72	7.26
Qwen2.5-VL-7B-Ins.	2	551.48	31774.13	4.02
InternVL2-8B	3	558.66	15917.26	1.08
deepseek-vl2-tiny	4	206.14	6548.17	0.08
deepseek-vl2-small	5	727.02	30928.16	0.06
V3Fusion	234	768.02	54239.56	5.18

Table 10: The effect of training data dependency on VL3Fusion for OCR-VQA, MMMU, and A-OKVQA datasets.

Train	OCR-VQA			MMMU	A-OKVQA
Data (%)	BLUE-1	EM	F1	Acc (%)	Acc (%)
1	15.70	0.03	26.84	42.41	87.69
5	82.32	68.58	83.34	52.33	87.78
10	84.53	71.01	85.26	52.17	87.70
30	85.57	71.51	86.29	52.84	87.26
50	86.13	72.34	86.76	52.84	86.99
75	86.12	71.84	86.74	54.22	86.48
100	86.27	71.78	86.86	54.50	88.31

Table 9 reports V3Fusion in total inference cost (time, memory) against individual VLMs in our model pool. The Mean inference time for 1000 samples of OCR-VQA is given. V3Fusion loads the pruned model pool in parallel, and the reflection phase takes 41ms. Compared to the top-performing single VLM (Qwen2.5-VL-7B), V3Fusion achieves the best overall performance at a reasonable cost, with an inference time of 768.02ms (Qwen2.5: 551.48ms) and 5.18 MB of memory consumed (Qwen2.5: 4.02 MB). In addition, our training data dependency is lightweight. Table 10 provides the effectiveness of training data in varying fractions. It shows that using only 5% of data for training V3Fusion to perform fusion is effective, achieving as good performance as using a large fraction (50 100%).

G.1 GA OPTIMALITY ANALYSIS

Table 11: The $N = 12$ model pool used to compare the GA with brute-force search, with each model's accuracy on MMMU. *Added* marks the six models added to the original pool of six.

Model	Pool	MMMU Acc. (%)
gemma-4-31B-it	Added	78.34
Qwen3-VL-235B-A22B	Added	75.92
InternVL3.5-241B-A28B	Added	70.45
Qwen2-VL-72B	Added	61.15
pixtral-12b-2409	Added	53.63
InternVL2-8B	Original	51.08
Qwen2.5-VL-7B	Original	50.32
granite-vision-4.1-4b	Added	47.52
deepseek-vl2-tiny	Original	37.71
llava-v1.6-vicuna-13b	Original	36.69
llava-v1.6-vicuna-7b	Original	35.54
deepseek-vl2-small	Original	31.72

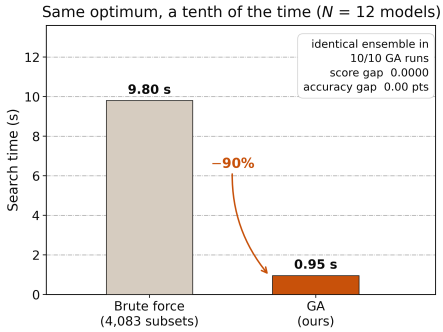


Figure 8: Search time on the $N = 12$ pool: brute force over all 4,083 candidate ensembles versus the GA. All 10 GA runs return the same ensemble as brute force, in about a tenth of the time.

To test whether the GA still finds the highest-scoring ensemble as the search space grows, we double the model pool from six to twelve models (Table 11). The six added models are mostly larger and stronger than the original ones, scoring 47.52–78.34% on MMMU versus 31.72–51.08%. With $N = 12$ models there are 4,083 candidate ensembles, which brute-force search evaluates exhaustively. In all 10 GA runs, the GA returned the same ensemble as brute-force search, with identical pruning score and accuracy, while cutting the search time from 9.80 s to 0.95 s, a 90% reduction (Figure 8). Since the number of candidate ensembles grows exponentially with the pool size, the savings should grow further for larger pools.

G.2 Robustness to Random Seeds and Hardware

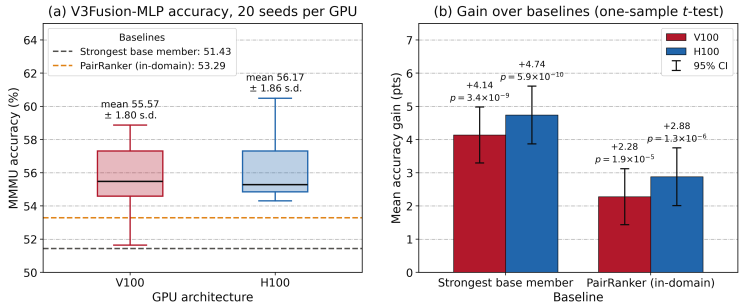


Figure 9: (a) Accuracy over 20 seeds on each GPU; boxes span the interquartile range, the black line marks the median, whiskers span the full range, and dashed lines mark the baselines. (b) Mean accuracy gain over each baseline with 95% confidence intervals; p-values are from one-sample t-tests ($n = 20$ per GPU).

In Figure 9, we train the V3Fusion-MLP fusion head with 20 random seeds on each of two GPU architectures, V100 and H100. Panel (a) shows the spread of MMMU accuracy across seeds: the mean is $55.57 \pm 1.80\%$ on V100 and $56.17 \pm 1.86\%$ on H100, so the two architectures agree to within 0.6 points. All 40 runs outperform the strongest base model (51.43%), and 38 of 40 also outperform the in-domain PairRanker baseline (53.29%). Panel (b) reports the mean gain over each baseline with its 95% confidence interval: $+4.14$ and $+4.74$ points over the strongest base model, and $+2.28$ and $+2.88$ points over PairRanker, on V100 and H100 respectively. A one-sample t-test rejects the hypothesis of zero gain in all four comparisons ($p \leq 1.9 \times 10^{-5}$).

H Rectification Stage Analysis

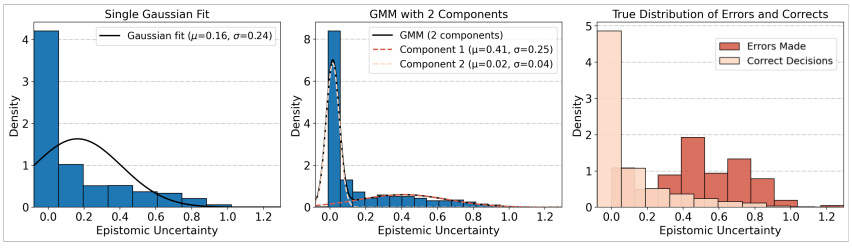


Figure 10: We illustrate the threshold selection process and the resulting boost in Vision Verification. The second and third plots show the single- and two-component Gaussian fits to the Epistemic Uncertainty, the Alg. 1 selects threshold as $\tau = 0.1315$.

Figure 10 shows how Alg. 1 selects the rejection threshold on A-OKVQA. The algorithm uses only the epistemic uncertainty of the ensemble on the 1,145 evaluation questions; correctness labels are never used. The left panel fits a single Gaussian ($\mu = 0.16$, $\sigma = 0.24$), which cannot represent the distribution: 57% of the questions have an uncertainty below 0.05, while the rest form a long, flat tail reaching 1.3. The Gaussian therefore underestimates the spike at zero and places about a quarter of

its mass on negative values, where almost no questions lie. The middle panel fits a two-component GMM. It separates a narrow low-uncertainty component ($\mu = 0.02$, $\sigma = 0.04$), holding about two-thirds of the questions, on which the base models agree, from a broad high-uncertainty component ($\mu = 0.41$, $\sigma = 0.25$), on which they disagree. The log-likelihood ratio between the two fits is 822.4, far above the margin $\alpha = 10$, so the GMM is selected and τ is set to the largest uncertainty in the low-uncertainty component, $\tau = 0.1315$, which is essentially where the two components are equally likely. The 67.2% of questions below τ keep the fused prediction, and the remaining 32.8% are passed to rectification. This choice matters: the single-Gaussian rule $\tau = \mu + 2\sigma \approx 0.65$ would sit in the middle of the error mass, flag only 7.4% of the questions, and catch only about a third of the errors.

The right panel checks τ against the ground truth by splitting the same uncertainty values by whether the fused prediction is correct. Correct predictions pile up near zero, whereas most errors lie between 0.25 and 0.9 (median uncertainty 0.01 vs. 0.49). The selected threshold separates the two well: about 85% of the errors lie above τ, three quarters of the correct predictions lie below it, and the predictions accepted without rectification are about 97% accurate. The errors left below τ are questions on which the base models agree with each other, which no disagreement-based measure can flag. Note that τ is selected without access to the correctness labels, which are used only in the right panel to evaluate it; its separation of errors from correct predictions is therefore not the result of tuning the threshold on the labels.