MIXTURE OF LAYERS: Dynamic Layer Routing for Visual Reasoning

Jeonghwan Kim[1]*, **Sofia Stoica**[1]*, **Jiwan Chung**[2], **Ansel Blume**[1],
Hyeonjeong Ha[1], **Zhenhailong Wang**[1], **Xin Luna Dong**[3], **Heng Ji**[1]
[1]University of Illinois Urbana-Champaign, [2]Yonsei University, [3]Meta Reality Labs
{jk100, sstoica2, hengji}@illinois.edu

Abstract

Pre-trained vision encoders contain layer-wise visual representations that differ in spatial granularity, semantic abstraction, and sensitivity to local details. However, most Multimodal Large Language Models (MLLMs) rely on only the final or penultimate vision encoder representations or fixed aggregation rules, making visual abstraction largely query-agnostic and limiting access to fine-grained cues such as small objects, spatial details, text, and subtle visual attributes. In this work, we propose **MIXTURE OF LAYERS** (**MoL**), an instruction-conditioned layer routing approach at visual patch level that dynamically aggregates query-relevant latent representations from intermediate vision encoder layers. Given a text query, **MoL** predicts routing probabilities over vision encoder layers and performs a top-k sparse aggregation over selected hidden states at either the image level, patch level, or through a hybrid routing mechanism. In doing so, **MoL** enables query-adaptive access to layer-specific visual features for *fine-grained* visual reasoning. Our experiments across 7 fine-grained visual reasoning tasks demonstrate substantial performance improvements, especially across fine-grained visual grounding and understanding tasks such as +18.9% improvement on *V** in overall accuracy, +4.5% on HRBench4K, and +16.3% on CharXiv compared to the baseline MLLMs, *without* resorting to multi-resolution inputs, simple interleaving of multiple vision encoders, or increasing the number of patch tokens. We study vision encoders' receptive field scales across different layers and their sampling behaviors to provide an in-depth analysis of why layer-wise sampling is helpful, demonstrating that conditional visual representations are a key step towards better visual perception and reasoning in MLLMs. Our project page is available at https://wjdghks950.github.io/mol.github.io/.

1 Introduction

Multimodal Large Language Models (MLLMs) process text and images as input, with designs that typically contain three components: one or more pre-trained vision encoders [40, 55, 38, 17, 20], a vision-language connector [36, 25, 42, 10, 28, 29, 5] that maps the visual features extracted from the vision encoder to the pre-trained Large Language Model (LLM) space [32, 47, 16], and the backbone LLM that takes as input the visual patches and textual tokens to perform complex visual understanding and reasoning. Despite their strong image-text understanding capabilities, many existing MLLMs typically assume that the *later layer* (e.g., final or penultimate) vision encoder representations provide a sufficiently universal abstraction for all downstream tasks as they are strongly aligned with semantic and language-level supervision during training.

Such assumptions, however, become especially limiting for *fine-grained multimodal perception and reasoning*, where answers often depend on small objects, localized text, subtle attributes, count-

* Equal contribution.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

ing, or spatial relations that may not be preserved in a single final-layer representation. We argue that such fixed set of later layers provide limited, query-agnostic access to such granular visual information, even though prior analyses show that spatial and semantic information is distributed differently across vision-encoder layers [14, 15, 7, 23, 3]. Even query-conditioned visual extraction approaches such as Q-Former-based architectures (e.g., BLIP-2 [28], InstructBLIP [10]) or Perceiver Resampler-based designs (e.g., Flamingo [1]) primarily operate on fixed final-layer vision representations rather than dynamically selecting among intermediate-layer features. Recent analyses show that pretrained vision encoders such as CLIP [40] and DINOv2 [38] encode diverse visual features across layers [14, 7, 23], with the most useful layer varying by downstream task [3]. While recent methods exploit intermediate-layer representations [52, 4, 34, 6, 30, 37, 33], they typically rely on manually selected fixed layers [18, 34, 4, 6], increase the number of injected visual tokens [52, 34], or conduct weighted averaging across the layers [33]. More importantly, these choices are largely *text-instruction agnostic*: the same set of representations are used regardless of visual reasoning task (i.e., prompts). Finally, such static treatment also contrasts with biological vision, where the brain appears to selectively emphasize task-relevant levels of this hierarchy rather than use all representations uniformly [44] . This suggests that effective visual processing in MLLMs should dynamically select the representations most relevant to the current instruction, rather than simply exposing the LLM to more visual features or fixed end-layer representations.

Inspired by this principle, we propose **MIXTURE OF LAYERS** (**MoL**), an instruction-guided layer routing mechanism that dynamically selects intermediate-layer visual patch representations from MLLM vision encoders. Rather than relying on a fixed set of visual features, **MoL** predicts instruction-conditioned routing distributions over vision-encoder layers, enabling different image regions and tasks to access different levels of visual abstraction. Conceptually, the router treats intermediate vision layers as a set of candidate representations and selectively aggregates the most *useful* ones for the current query. We instantiate **MoL** with three routing variants: layer-level routing ($\textbf{MoL}_{layer}$), patch-level routing ($\textbf{MoL}_{patch}$), and hybrid routing ($\textbf{MoL}_{hybrid}$), which combines the preceding two variants. Specifically, $\textbf{MoL}_{layer}$ performs image-level routing across the layers of the pre-trained vision encoder, while $\textbf{MoL}_{patch}$ predicts a separate layer distribution for each visual patch, allowing different image regions to draw from different abstraction levels within the vision hierarchy. $\textbf{MoL}_{hybrid}$ combines the two routing schemes through a weighted product-of-experts [21] and uses their disagreement to gate a reserve layer (§3.1), balancing local fine-grained evidence with broader semantic context. **MoL** is particularly effective in fine-grained visual reasoning, showing significant improvements on downstream tasks such as *V** and CharXiv with gains of 18.9% and 16.3%, respectively. In practice, the added cost is small relative to the overall model: the router contributes negligible parameter overhead (<0.02% over the 13B base model) while leaving token-by-token language decoding unchanged.

Our work makes three key contributions:

- We introduce a novel routing formulation that adaptively selects intermediate vision-encoder layer representations conditioned on the input instruction, supporting both image-level $\textbf{MoL}_{layer}$ and patch-level $\textbf{MoL}_{patch}$ decisions.
- We develop a hybrid routing mechanism $\textbf{MoL}_{hybrid}$ with a disagreement-gated reserve layer (§3.1) that balances local fine-grained evidence and global semantic context while routing across vision encoder layers.
- Across seven different fine-grained visual reasoning benchmarks, **MoL** substantially improves fine-grained visual reasoning without increasing the number of visual tokens passed into the LLM or relying on multiple vision encoders, making **MoL** complementary to existing MLLM scaling approaches.

2 Related Work

Multimodal Large Language Models Multimodal Large Language Models (MLLMs) ingest visual patch tokens and textual prompts to generate textual responses [36, 56, 27, 2, 10, 28, 11]. Most MLLMs rely on a pre-trained vision encoder, a vision-language connector that projects visual features into the LLM input space, and a pre-trained LLM backbone [36, 56, 27, 2, 11]. While architectures such as BLIP-2 [28], InstructBLIP [10], and Flamingo [1] introduce text-conditioned querying modules such as Q-Former or Perceiver Resampler, they still operate on fixed final or penultimate-layer vision representations. Similarly, visual abstractors and resamplers in models such as mPLUG-

OWL [53] and Honeybee [5] compress or transform already-extracted visual embeddings rather than dynamically selecting intermediate-layer features.

Consequently, most MLLMs continue to rely on the final or penultimate layers of pre-trained vision encoders [40, 55, 38] under the assumption that these representations sufficiently capture the visual information required for downstream reasoning tasks. In contrast, **MoL** performs instruction-conditioned routing *across* intermediate vision-encoder layers, enabling different image regions and tasks to access different levels of visual abstraction. We leave aside vision encoder-agnostic frameworks such as Emu-3.5 [9] and SOLO [8], since they do not integrate pre-trained vision encoders.

Intermediate Layer Sampling from Pre-trained Vision Encoders Prior work evaluating the beneficial effect of intermediate visual features on MLLMs' visual reasoning [4, 52, 34, 6, 30] shares a similar motivation to our proposed method: look into the intermediate layers' hidden states to retrieve useful visual information. However, they focus on either rule-based selection or visual feature-only sampling strategies, limiting their applicability to novel, fine-grained downstream tasks. **MoL** overcomes prior static approaches to intermediate representations through instruction-guided sampling at multiple levels of representational granularity (i.e., patch- and layer-level). Concurrent work such as TGIF [33] predicts layer weights from the query through an MLP and weighted averages all vision layers. In contrast, $\text{MoL}_{\text{layer}}$ directly measures query-layer representation compatibility and performs sparse top-k routing, allowing the selected abstraction depth to depend on the visual evidence present in the input rather than only on the semantics of the prompt.

Fine-grained Visual Perception and Reasoning Most widely used visual reasoning benchmarks (e.g., MMMU [54], TextVQA [43], POPE [31], GQA [22]) primarily evaluate coarse-grained understanding and do not stress fine-grained perception in dense, high-resolution scenes critical in real-world tasks such as industrial inspection or document/diagram understanding. Recent benchmarks such as *V** [50], HRBench [48], NaturalBench [26], RealWorldQA [51], and CharXiv [49] address this gap by requiring models to reason over localized text, small objects, and subtle attributes.

Prior approaches to improving fine-grained reasoning largely focus on increasing the visual token budget via multi-resolution or native-resolution inputs [27, 37, 2], or concatenating patch embeddings from multiple pre-trained vision encoders [42, 46, 23]. By contrast, **MoL** improves fine-grained reasoning by refining the representation of each patch through instruction-conditioned selection of intermediate-layer features, without introducing additional visual tokens or simply concatenating features from multiple vision encoders [42, 46, 23]. This positions **MoL** as a simple, complementary approach to enhance visual representation quality without scaling input size, allowing it to be integrated with existing multi-patch, high-resolution approaches.

3 MIXTURE OF LAYERS: Selective Visual Representations for Fine-Grained Visual Reasoning

3.1 Preliminary

We propose **MIXTURE OF LAYERS** (**MoL**), an instruction-aligned layer routing mechanism that dynamically selects the most useful vision layers for each input image and text instruction. **MoL** can be seamlessly attached to different MLLM backbones without changing the language model architecture. Suppose we have aligned vision and text encoders as in dual-encoder architectures like CLIP [40], SigLIP [55], or DINOv2 w/ Txt [24]. Define the vision encoder's hidden states for its L layers as:

$$\mathbf{H}^1, \mathbf{H}^2, \ldots, \mathbf{H}^L, \qquad \mathbf{H}^\ell \in \mathbb{R}^{N \times d_v}, \tag{1}$$

where N is the number of visual tokens and d_v is the vision encoder hidden dimension. Let the vision encoder's corresponding text encoder[2] path produce the prompt embedding sequence

$$\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_M] \in \mathbb{R}^{M \times d_t}, \quad \bar{\mathbf{t}} = \text{Pool}(\mathbf{T}) \in \mathbb{R}^{d_t} \tag{2}$$

where $\mathbf{T}$ is a sequence of M embedded text tokens of dimension d_t and Pool is an arbitrary pooling operator. We use the pooled text representation $\bar{\mathbf{t}}$ as a query to score vision-layer features for routing:

$$\mathbf{q} = W_Q \bar{\mathbf{t}}, \qquad \mathbf{k}_n^\ell = W_K \mathbf{h}_n^\ell, \qquad s_{n,\ell}^{(p)} = \frac{\mathbf{q}^\top \mathbf{k}_n^\ell}{\sqrt{d_v}}. \tag{3}$$

[2]We refer to vision tower-aligned text encoders as in CLIP [40], SigLIP [55], or DINOv2 w/ Txt [24]

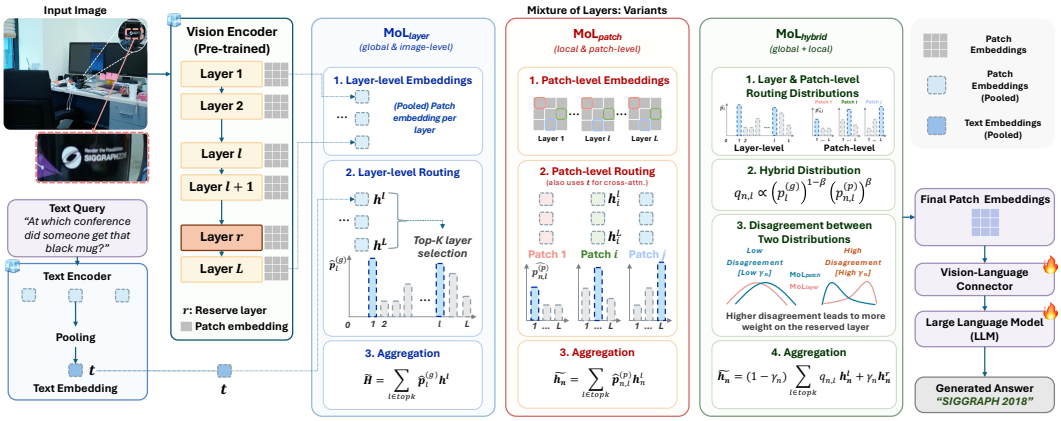


Figure 1: **Overview of MIXTURE OF LAYERS.** This figure illustrates a fine-grained visual reasoning scenario where the MLLM should ground its reasoning based on visual representations that accurately capture the query-relevant visual features. **MoL** can be divided into three complementary variants: (i) $\textbf{MoL}_{layer}$ that generates a query-guided ($\bar{t}$) layer-level probability distribution over the input image; (ii) $\textbf{MoL}_{patch}$, which generates a query-guided distribution over layers for each visual patch; (iii) $\textbf{MoL}_{hybrid}$ that measures the disagreement between the $\textbf{MoL}_{layer}$ and $\textbf{MoL}_{patch}$ and assigns gating weights to the reserve layer (§3.1) representations, which subsequently are pushed through weighted linear combination to form the final patch embedding sequence. We leave out some arrows (e.g., text query to LLM backbone) for brevity.

where $\mathbf{h}_n^\ell$ is the hidden state for each layer ℓ and patch token n. We pass these hidden states into our routing mechanism, which treats each vision layer as an *expert*, similar to MoE [41]. Among the L layers of the pre-trained vision encoder, we optionally designate a "reserve layer" $r \in \{1, \ldots, L\}$ as a stable fallback when routing is uncertain or patch-level evidence is noisy; these layers are the last or penultimate layers originally selected by the backbone MLLMs. Reserve layers, as we show in Section 4.4, provide an essential anchor representation to enhance visual reasoning in **MoL**. For variants with a reserve mechanism, the router selects the top-k non-reserve layers $\mathcal{S}_n$ and combines them with a reserve layer r through a learned gate:

$$\tilde{\mathbf{h}}_n = (1 - \gamma_n) \sum_{\ell \in \mathcal{S}_n} \hat{q}_{n,\ell} \mathbf{h}_n^\ell + \gamma_n \mathbf{h}_n^r, \qquad \hat{q}_{n,\ell} = \frac{q_{n,\ell}}{\sum_{j \in \mathcal{S}_n} q_{n,j}}. \tag{4}$$

Here, $\gamma_n \in [0, 1]$ is the learned reserve-layer gate for patch n, and $q_{n,\ell}$ denotes the router distribution over non-reserve layers. This prevents the model from collapsing to potentially erroneous layers while still allowing it to exploit specialized intermediate representations when appropriate. We denote the set of non-reserve layers by $\mathcal{L}_{\text{nonres}} = \{1, \ldots, L\} \setminus \{r\}$. The final routing distribution, therefore, is decomposed into reserve and non-reserve mass: $p_{n,r} = \gamma_n, \quad p_{n,\ell} = (1 - \gamma_n) q_{n,\ell}, \quad \ell \in \mathcal{L}_{\text{nonres}}$, We subsequently train the model with the standard autoregressive loss, $\mathcal{L}_{\text{LM}}$, and an auxiliary load-balancing loss, $\mathcal{L}_{\text{lb}}$, adopted from Switch transformers [13], to prevent router collapse by encouraging balanced usage of vision encoder layers; λ_{lb} controls its strength.: $\mathcal{L} = \mathcal{L}_{\text{LM}} + \lambda_{\text{lb}} \mathcal{L}_{\text{lb}}$ We further elaborate on the training details in the Appendix.

3.2 MIXTURE OF LAYERS

There are three different ways to calculate the image features using our approach:

Layer-level Sampling ($\textbf{MoL}_{layer}$) The simplest variant performs one shared routing decision for the whole image. We first summarize each layer by average-pooling its visual tokens $\bar{\mathbf{h}}^\ell = \frac{1}{N} \sum_{n=1}^{N} \mathbf{h}_n^\ell$. Using the text query $\bar{\mathbf{t}}$, the router computes a layer distribution p_ℓ over $\ell \in \{1, \ldots, L\}$, selects the top-$k$ layers $\mathcal{S} = \text{TopK}(\mathbf{p}, k)$, and renormalizes their probabilities as $\hat{p}_\ell = p_\ell / \sum_{j \in \mathcal{S}} p_j$. The routed image representation is then written as: $\tilde{\mathbf{H}} = \sum_{\ell \in \mathcal{S}} \hat{p}_\ell \mathbf{H}^\ell$ Thus, $\textbf{MoL}_{layer}$ applies the same selected layers and weights to all patches in the image.

Patch-level Sampling ($\textbf{MoL}_{patch}$) Patch-level routing instead makes a separate layer selection decision for each visual token. For patch n, the router computes a text-conditioned distribution $p_{n,\ell}$ over layers, selects $\mathcal{S}_n = \text{TopK}(\mathbf{p}_{n,:}, k)$, and renormalizes the selected weights as $\hat{p}_{n,\ell} = p_{n,\ell} / \sum_{j \in \mathcal{S}_n} p_{n,j}$. The routed patch representation is $\tilde{\mathbf{h}}_n = \sum_{\ell \in \mathcal{S}_n} \hat{p}_{n,\ell} \mathbf{h}_n^\ell$

This allows different visual patch tokens (i.e., image regions) to draw from different vision-layer abstractions while keeping the number of output patch tokens unchanged.

Hybrid Sampling (MoL$_{hybrid}$) MoL$_{hybrid}$ unifies layer-level and patch-level sampling to combine the stability of global routing with the flexibility of local routing, improving adaptability across downstream tasks. We first compute a global routing distribution $p_\ell^{(g)}$ and patch-level, local routing distribution $p_{n,\ell}^{(p)}$ as

$$p_\ell^{(g)} = \mathrm{softmax}\left(\frac{s_\ell^{(g)}}{\tau}\right), \quad p_{n,\ell}^{(p)} = \mathrm{softmax}_\ell\left(\frac{s_{n,\ell}^{(p)}}{\tau}\right). \tag{5}$$

Here, $p_\ell^{(g)}$ is the image-level routing probability for layer ℓ, and $p_{n,\ell}^{(p)}$ is the patch-level routing probability for patch n and layer ℓ. We then predict a text prompt–dependent mixing weight: $\beta = \sigma(\mathbf{w}_\beta^\top \bar{\mathbf{t}} + b_\beta)$. $\beta \in [0,1]$ controls the interpolation between global and local routing, $\mathbf{w}_\beta$ and b_β are learned parameters, and $\sigma(\cdot)$ is the sigmoid function. The hybrid non-reserve routing distribution is computed via a weighted product-of-experts [21]: $q_{n,\ell} \propto \left(p_\ell^{(g)}\right)^{1-\beta}\left(p_{n,\ell}^{(p)}\right)^{\beta}$. Here, $q_{n,\ell}$ is the fused non-reserve routing probability for patch n and layer ℓ. Equivalently, in log-space, this would be written as follows:

$$\log q_{n,\ell} = (1-\beta)\log p_\ell^{(g)} + \beta \log p_{n,\ell}^{(p)} - \log Z_n, \tag{6}$$

where Z_n is the normalization constant for patch n.

As discussed above (§3.1), we also use a *reserve layer*, r, a penultimate or last layer of the vision tower, as a fallback layer. To determine *when* to fallback or not, we measure the disagreement between local and global routing with

$$d_n = \mathrm{KL}\left(p_{n,:}^{(p)} \,\|\, p^{(g)}\right) = \sum_{\ell \neq r} p_{n,\ell}^{(p)}\left(\log p_{n,\ell}^{(p)} - \log p_\ell^{(g)}\right). \tag{7}$$

where d_n is the disagreement score for patch n, $\mathrm{KL}(\cdot\|\cdot)$ denotes KL divergence, and r is the reserve layer index. Then, we map disagreement to a reserve layer probability as $\gamma_n = \sigma(w_d d_n + b_d)$ wherein $\gamma_n \in [0,1]$ is the reserve weight for patch n, and w_d, b_d are learned scalar parameters. Intuitively, the higher the d_n, i.e., the disagreement between the layer- and patch-level distributions, the more we rely on the reserve layer and vice versa. Letting $\mathcal{S}_n = \mathrm{TopK}(\mathbf{q}_{n,:}, k)$ denote the selected non-reserve layers, the final patch representation is

$$\tilde{\mathbf{h}}_n = (1-\gamma_n)\sum_{\ell \in \mathcal{S}_n} \hat{q}_{n,\ell}\mathbf{h}_n^\ell + \gamma_n \mathbf{h}_n^r, \quad \text{where} \quad \hat{q}_{n,\ell} = \frac{q_{n,\ell}}{\sum_{j\in\mathcal{S}_n} q_{n,j}}. \tag{8}$$

Here, $\hat{q}_{n,\ell}$ denotes the renormalized hybrid routing weight, and $\mathbf{h}_n^r$ is the representation of patch n from the reserve layer r.

Sampling from Layers of Multiple Vision Encoders Unlike prior works that concatenate or interleave last-layer features from multiple vision encoders [46, 45], **MoL** can first route within each vision encoder and then fuse the routed representations across vision encoders. For each vision encoder m, we apply one of the proposed routing variants, $\alpha_m \in \{\mathrm{layer}, \mathrm{patch}, \mathrm{hybrid}\}$, to obtain

$$\tilde{\mathbf{H}}^{(m)} = \mathrm{MoL}_{\alpha_m}^{(m)}\left(\{\mathbf{H}^{(m,\ell)}\}_{\ell=1}^{L_m}, \bar{\mathbf{t}}^{(m)}\right), \tag{9}$$

where $\bar{\mathbf{t}}^{(m)}$ is the vision encoder-aligned text representation. For instance, in the CLIP + DINOv2 w/ Txt setting in Table 3, we apply MoL$_{hybrid}$ independently to both vision encoders and fuse their routed representations through concatenation and upsampling. After routing, we project both branches into the language model's hidden space: $\mathbf{Z}^c = \Pi_c(\tilde{\mathbf{H}}^c)$, $\mathbf{Z}^d = \Pi_d(\tilde{\mathbf{H}}^d)$, where c and d denote CLIP and DINOv2, respectively, and Π_c and Π_d are the corresponding vision-language projectors. Here, $\mathbf{Z}^c \in \mathbb{R}^{N_c \times d_{\mathrm{LLM}}}$ and $\mathbf{Z}^d \in \mathbb{R}^{N_d \times d_{\mathrm{LLM}}}$, where N_c and N_d are the numbers of patch tokens from each vision encoder, and d_{LLM} is the language model's hidden dimension. Since DINOv2 and CLIP branches produce 256 and 576 patch tokens, respectively, we first upsample the projected DINOv2 features to the CLIP token-grid size: $\hat{\mathbf{Z}}^d = \mathrm{Upsample}_{N_d \to N_c}(\mathbf{Z}^d) \in \mathbb{R}^{N_c \times d_{\mathrm{LLM}}}$,

Table 1: **Performance across backbones and router types on fine-grained visual reasoning benchmarks.** Metrics reported are accuracy. V^* benchmark is broken into its constituent sub-categories for detailed analysis. **Best** Δ denotes the maximum improvement (in absolute accuracy points) achieved by any **MoL** variant over the corresponding backbone baseline for each metric. We include $\textbf{MoL}_{\text{layer}}$ for the Vicuna-13B settings with CLIP and DINOv2 w/ Txt as global-routing comparisons. The multi-encoder results are reported separately in Table 3.

Method	V^* *all*	*GPT4V-hard*	*OCR*	*direct attr.*	*rel. pos.*	MMStar	HRBench8K	HRBench4K	RealWorldQA	NaturalBench	CharXiv
LLaVA-v1.5-13B (Vicuna-13B + CLIP) [35, 40]											
LLaVA-v1.5-13B	71.43	52.94	50.00	65.22	93.42	30.00	36.88	43.75	51.64	68.93	23.84
$\text{MoL}_{\text{layer}}$	76.89	58.82	53.33	75.65	92.11	29.53	38.00	40.62	52.35	69.71	26.25
$\text{MoL}_{\text{patch}}$	**84.87**	58.82	**60.00**	**86.09**	**98.68**	30.13	38.50	42.25	51.82	69.41	**29.17**
$\text{MoL}_{\text{hybrid}}$	84.03	**64.71**	56.67	85.22	97.37	**31.07**	**38.62**	**43.88**	**53.14**	**70.36**	25.56
Best Δ	**+13.44**	**+11.77**	**+10.00**	**+20.87**	**+5.26**	**+1.07**	**+1.74**	**+0.13**	**+1.50**	**+1.43**	**+5.33**
Vicuna-13B + DINOv2 w/ Txt [35, 24]											
Vicuna-13B + DINOv2	65.55	**76.47**	40.00	67.83	69.74	28.87	32.75	35.38	50.46	**68.29**	22.92
$\text{MoL}_{\text{layer}}$	67.65	52.94	36.67	60.87	93.42	27.80	34.12	35.38	50.46	67.92	25.94
$\text{MoL}_{\text{patch}}$	76.05	64.71	40.00	74.78	94.74	28.47	33.37	34.88	51.02	66.03	17.71
$\text{MoL}_{\text{hybrid}}$	**84.45**	70.59	**50.00**	**86.09**	**98.68**	**30.13**	**36.50**	**39.87**	**52.29**	67.89	**26.92**
Best Δ	**+18.90**	-5.88	**+10.00**	**+18.26**	**+28.94**	**+1.26**	**+3.75**	**+4.49**	**+1.83**	-0.40	**+4.00**
Llama-3-8B-V (LLama-3-8B-Instruct + SigLIP) [16, 55]											
Llama-3-8B-V	54.62	47.06	66.67	56.52	48.68	**37.73**	**44.00**	**51.12**	**59.87**	74.75	25.05
$\text{MoL}_{\text{patch}}$	52.52	41.18	43.33	60.00	47.37	37.07	40.00	48.88	58.92	**75.02**	26.16
$\text{MoL}_{\text{hybrid}}$	**63.03**	**47.06**	**76.67**	**70.43**	**50.00**	33.40	42.13	45.00	45.62	72.05	**26.82**
Best Δ	**+8.42**	+0.00	**+10.00**	**+13.91**	**+1.32**	-0.66	-1.87	-2.24	-0.95	**+0.27**	**+1.77**
Phi-1.5-1.3B + SigLIP [32, 55]											
Phi-1.5-1.3B	42.01	23.52	26.66	35.65	61.84	34.93	32.00	36.50	34.93	67.31	4.07
$\text{MoL}_{\text{patch}}$	39.92	29.41	50.00	28.70	55.26	33.60	30.63	33.12	38.82	60.86	19.65
$\text{MoL}_{\text{hybrid}}$	**50.84**	**23.52**	**43.33**	**42.60**	**72.36**	34.93	32.00	**37.25**	**39.74**	**68.00**	**20.46**
Best Δ	**+8.83**	**+0.00**	**+16.67**	**+6.95**	**+10.52**	+0.00	+0.00	**+0.75**	**+4.81**	**+0.69**	**+16.39**

Table 2: **Performance comparison against existing state-of-the-art methods on multi-layer visual feature integration.** With all the existing baselines providing LLaVA-baseline checkpoints, we use LLaVA-v1.5-13B [35] as our baseline in this experiment.

Method	V^* *all*	*GPT4V-hard*	*OCR*	*direct attr.*	*rel. pos.*	MMStar	HRBench8K	HRBench4K	RealWorldQA	NaturalBench	CharXiv
LLaVA-v1.5-13B	71.43	52.94	50.00	65.22	93.42	30.00	36.88	43.75	51.64	68.93	23.84
TokenPacker [29]	73.53	47.06	**63.33**	66.09	93.74	29.07	38.00	43.38	54.12	67.63	22.76
MLVF [34]	78.92	45.88	54.00	79.13	96.05	29.84	36.88	42.50	51.90	69.80	24.72
DeepStack-L [37]	65.13	64.71	46.67	56.52	85.53	29.26	37.63	43.75	**57.77**	69.10	21.31
Dense Connector [52]	81.35	47.05	56.67	83.47	**98.68**	30.06	37.12	43.75	52.41	**71.41**	25.23
IGVA [34]	70.38	45.12	52.20	66.79	88.60	29.26	36.17	42.10	51.85	67.63	23.89
TGIF [33]	63.87	52.94	50.00	56.52	82.89	29.07	33.75	40.63	54.38	69.20	20.00
$\text{MoL}_{\text{layer}}$	76.89	58.82	53.33	75.65	92.11	29.53	38.00	40.62	52.35	69.71	27.85
$\text{MoL}_{\text{patch}}$	**84.87**	58.82	60.00	**86.09**	**98.68**	30.13	38.50	42.25	51.82	69.41	**28.65**
$\text{MoL}_{\text{hybrid}}$	84.03	**64.71**	56.67	85.22	97.37	**30.47**	**38.62**	**43.88**	53.14	70.36	26.97

Table 3: **Comparison of single- and multi-encoder routing with a Vicuna-13B backbone.** Interleaved-MoF interleaves CLIP and DINOv2 patch inputs. **MoL** routes within one or both encoders; the dual-encoder variant uses concatenation/upsampling followed by a learned projector. **Best** Δ reports the largest improvement among the three **MoL** configurations over Interleaved-MoF, computed from the displayed scores in percentage points. † denotes DINOv2 with an aligned text encoder.

Method	Visual Input	V^* *all*	*GPT4V-hard*	*OCR*	*direct attr.*	*rel. pos.*	MMStar	HRBench8K	HRBench4K	RealWorldQA	NaturalBench	CharXiv
Interleaved-MoF [46]	CLIP + DINOv2	82.35	58.82	53.33	82.61	**98.68**	31.87	35.75	**44.12**	**57.77**	67.26	26.73
Dual Enc No Router	CLIP + DINOv2	84.45	70.59	66.67	85.22	93.42	30.47	36.13	43.50	53.86	69.12	28.97
$\textbf{MoL}_{\text{hybrid}}$	CLIP	84.03	64.71	56.67	85.22	97.37	31.07	**38.62**	43.88	53.14	**70.36**	25.56
$\textbf{MoL}_{\text{hybrid}}$	DINOv2†	84.45	70.59	50.00	86.09	**98.68**	30.13	36.50	39.87	52.29	67.89	26.92
$\textbf{MoL}_{\text{hybrid}}$ - Dual Enc.	CLIP + DINOv2†	**90.34**	**82.35**	**66.67**	**92.17**	**98.68**	**32.00**	37.62	43.50	52.55	69.72	**31.13**
Best Δ		**+7.99**	**+23.53**	**+13.34**	**+9.56**	**+0.00**	**+0.13**	**+2.87**	**-0.24**	**-4.63**	**+3.10**	**+4.40**

where Upsample uses bilinear interpolation. We subsequently concatenate the two representations along the feature dimension and pass them through a learned linear projector:

$$\mathbf{F} = \left[\mathbf{Z}^c \,\|\, \hat{\mathbf{Z}}^d\right] \mathbf{W}_f + \mathbf{1}_{N_c} \mathbf{b}_f^\top, \qquad \mathbf{F} \in \mathbb{R}^{N_c \times d_{\text{LLM}}}, \tag{10}$$

where $\|$ denotes feature concatenation, $\mathbf{W}_f \in \mathbb{R}^{2d_{\text{LLM}} \times d_{\text{LLM}}}$ and $\mathbf{b}_f \in \mathbb{R}^{d_{\text{LLM}}}$ are learned parameters, and $\mathbf{1}_{N_c}$ broadcasts the bias across patch tokens. This fusion allows the model to combine query-relevant representations from both vision encoders while preserving the CLIP branch's output token length. The resulting patch embedding sequence $\mathbf{F}$ is then passed to the language model.

4 Experiments

In this section, we elaborate on the experimental setup of the backbone multimodal LLMs (MLLMs) we apply **MIXTURE OF LAYERS** (**MoL**) to, discuss the downstream fine-grained visual understanding and reasoning benchmarks, detail the analytical frameworks employed to study layer-wise

sampling behaviors, and demonstrate how **MoL** sampled layer representations enhance fine-grained visual understanding in MLLMs.

Models Our work builds **MoL** in a fully reproducible MLLM training framework built on top of [35, 19]. This framework provides both the vision-language connector pre-training and multimodal instruction-tuning stages, unlike open-weight-only models [2, 16] where the underlying training pipeline data is not fully available. Namely, in this work we use LLaVA-1.5-13B [35], Llama-3-8B-V [16], and Phi-1.5-1.3B[32] + SigLIP [55] as experimental baselines, all of which are open-sourced by the Bunny framework [19]. The pre-trained vision encoders span CLIP [40], DINOv2 w/ Txt[38], and SigLIP [55]. For DINOv2 w/ Txt we pair it with Vicuna-13B, which is equivalent of a LLaVA-v1.5-13B with DINOv2 w/ Txt as the vision encoder. For additional training details, please refer to Appendix.

Dataset We adopt the exact pre-training and instruction-tuning datasets provided by the individual backbone MLLMs [35, 11, 19] to ensure a fair comparison. For fine-grained visual understanding benchmarks, we employ *V** [50], HRBench4K, HRBench8K [48], NaturalBench [26], RealWorldQA [51], and CharXiv [49] to evaluate the baseline and **MoL**-enhanced capabilities of different MLLMs.

4.1 Effect of Mixture of Layers on Model Perception

Fine-grained Visual Understanding Table 1 shows that **MoL** improves fine-grained visual reasoning across multiple backbone architectures and vision encoders, across different parameter sizes (13B, 8B, 1.5B). On the primary LLaVA-v1.5-13B setting, **MoL** yields consistent gains over the baseline, with especially large improvements on *V** overall (+13.44), direct attribute recognition (+20.87), OCR (+10.00), and CharXiv (+5.33). Similar gains appear with DINOv2 w/ Txt, where $\textbf{MoL}_{\text{hybrid}}$ improves *V** by +18.90 and relative position by +28.94, suggesting that adaptive layer routing is not specific to CLIP-based vision encoders. The gains are also visible in smaller or different backbones: for Phi-1.5-1.3B, **MoL** substantially improves OCR (+16.67), relative position (+10.52), and CharXiv (+16.39). Across settings, $\textbf{MoL}_{\text{patch}}$ is often strongest on localized perceptual categories, while $\textbf{MoL}_{\text{hybrid}}$ tends to perform better on benchmarks requiring both local grounding and broader reasoning. These results suggest that selecting task-relevant intermediate layers is a general mechanism for improving fine-grained perception, rather than a backbone-specific optimization.

Comparison against State-of-the-Art In Table 2, we compare **MoL** with prior approaches that aggregate intermediate visual features via multi-resolution stacking (DeepStack [37]), token compression and fusion (TokenPacker [29]), or dense cross-layer connections (MLVF [34], Dense Connector [52]). Unlike these fixed aggregation schemes, **MoL** adaptively routes vision-layer representations conditioned on the input instruction. **MoL** consistently improves fine-grained visual reasoning. $\textbf{MoL}_{\text{patch}}$ achieves the best overall *V** performance, outperforming Dense Connector by 3.52 points, with particularly strong gains on localized categories such as `OCR` and `direct attribute`. $\textbf{MoL}_{\text{hybrid}}$ further improves performance on more challenging benchmarks (e.g., GPT4V-hard, MM-Star, HRBench), indicating the benefit of combining global and patch-level routing. Importantly, **MoL** operates with a single vision encoder yet matches or outperforms prior multi-encoder approaches (e.g., Interleaved-MoF [46]) as shown in Table 3, demonstrating that adaptive layer routing can be more effective than increasing visual diversity through simple interleaving of additional encoder features or more patch tokens.

Coarse-grained Visual Understanding Although **MoL** targets fine-grained perception, we also evaluate standard VQA benchmarks in Table 4. **MoL** largely preserves coarse-grained reasoning performance, with small gains on POPE, GQA, and MMMU, and minimal change on TextVQA. We include the full results here and defer additional analysis to the Appendix.

Table 4: **Coarse-grained VQA performance.**

Model	POPE	GQA	TextVQA	MMMU
LLaVA-v1.5-13B (Vicuna-13B + CLIP) [35, 40]				
LLaVA-v1.5-13B	85.60	63.26	**61.25**	36.89
$\text{MoL}_{\text{patch}}$	86.23	63.73	61.09	**38.22**
$\text{MoL}_{\text{hybrid}}$	**86.47**	**63.75**	61.11	37.00
Best Δ	**+0.87**	**+0.49**	-0.14	**+1.33**
Vicuna-13B + DINOv2 w/ Txt [35, 24]				
Vicuna-13B + DINOv2	84.05	61.46	46.87	34.67
$\text{MoL}_{\text{patch}}$	**84.12**	59.96	**47.12**	34.11
$\text{MoL}_{\text{hybrid}}$	84.08	60.57	46.80	34.67
Best Δ	**+0.07**	-0.89	+0.25	+0.00

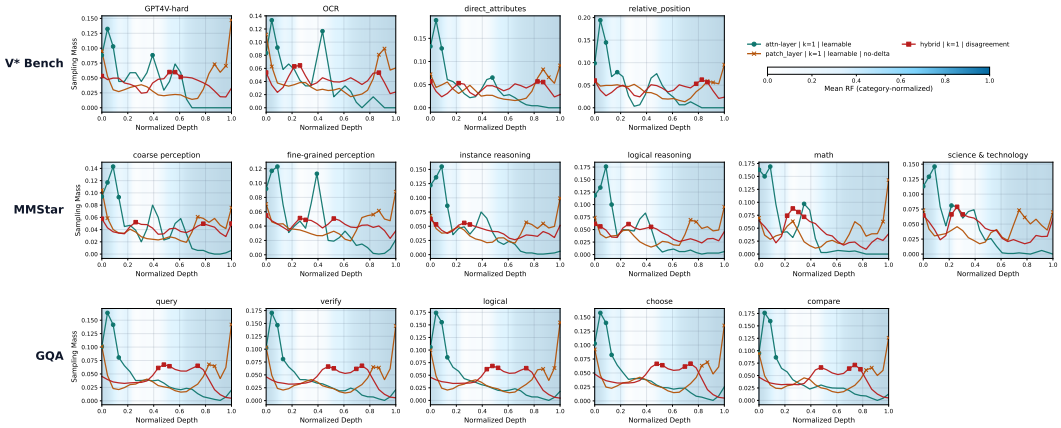


Figure 2: **Layer selection behaviors of MIXTURE OF LAYERS.** Average normalized routing probability across vision-encoder layers for LLaVA-v1.5-13B on *V**Bench, MMStar, and GQA categories. Background intensity shows category-normalized receptive field (RF) [39]; lighter regions indicate more localized layers. Markers denote selected top-k layers ($k = 4$). DINOv2 variants are shown in Appendix Figure 4.

4.2 MIXTURE OF LAYERS with Multiple Vision Encoders

Setup We extend hybrid routing to a Vicuna-13B backbone with CLIP and DINOv2 w/ Txt, applying a separate text-conditioned router to each vision encoder. The evaluated configuration selects one non-reserve layer per patch ($k = 1$) and combines it with the encoder's reserve representation using the disagreement gate. Each routed stream is projected to the language model's hidden dimension. The concatenation-upsampling fusion resizes the DINOv2 stream to the CLIP token-grid size, concatenates the two streams along the feature dimension, and applies a learned linear projection to produce 576 visual tokens. Table 3 compares this configuration with Interleaved-MoF [46], and an additional baseline for dual encoder representation concatenation-upsampling without the router (Dual Enc. No Router), where both build on Vicuna-13B backbone. This is a comparison between complete systems; the pretrained encoders and fusion mechanisms are not a controlled routing-only ablation.

Results The multi-encoder configuration achieves 90.34 on *V**, compared with 82.35 for Interleaved-MoF. The largest gain is on GPT4V-hard (82.35 versus 58.82), followed by OCR (66.67 versus 53.33) and direct attributes (92.17 versus 82.61); relative-position accuracy is tied at 98.68. These changes correspond to 19 additional correct answers out of 238: four on GPT4V-hard, four on OCR, and eleven on direct attributes. The same checkpoint achieves 31.13 on CharXiv. The remaining reported comparisons are mixed, including lower HRBench4K and RealWorldQA scores. We therefore do not infer a uniform advantage across benchmarks from this comparison.

Interpretation and Insights A plausible explanation is that routing and encoder diversity address different sources of lost visual information. Routing exposes query-relevant intermediate representations within each encoder, while the two encoders supply different learned representations of the image. The learned channel projection gives the language model access to both routed streams without extending the sequence beyond the CLIP token count. Combining an image-level routing preference with patch-specific evidence also allows the selected abstraction to vary across regions, while the reserve representation provides an anchor. The concentration of gains in OCR, attributes, and challenging visual questions is consistent with this explanation; the unchanged relative-position result suggests that extra visual diversity is less useful when the baseline is already near ceiling.

4.3 Analysis of Sampling Behavior

Layer-wise Sampling Behavior Figure 2 shows that **MoL** learns non-uniform, task-dependent routing patterns across benchmarks and question categories. The layer-level router (**MoL**$_{layer}$), which assigns one distribution to the entire image consistently prioritizes early layers, especially for categories requiring localized evidence such as OCR, direct attributes, and relative position. This is because global routing must rely on features that are consistently informative across the whole image, and early layers preserve fine-grained spatial details before they are aggregated. By

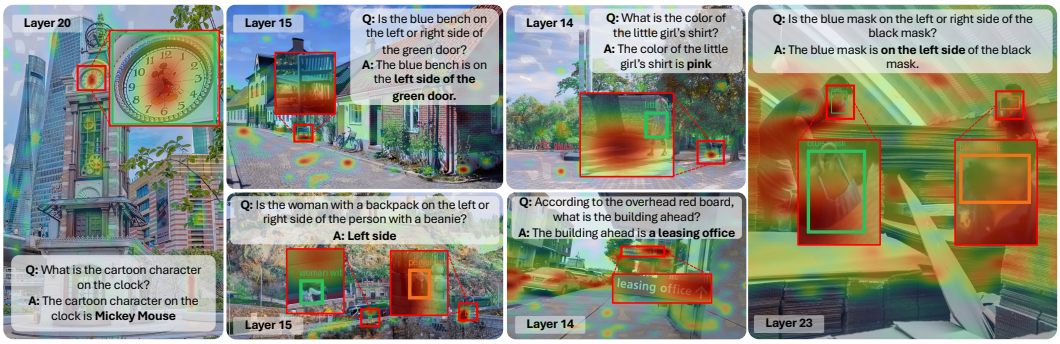


Figure 3: **Self-attention heatmap visualization of the selected layers aggregated across all the queries of the layers on the selected layers of CLIP-ViT-L14.** For additional qualitative analysis on the selected layers of the DINOv2 encoder, please refer to Figure 5.

Table 5: **Ablation on vision encoder-aligned text encoder.** $\textbf{MoL}_{image}$ uses image-only routing, where the pooled embeddings of each vision encoder layer is converted to logits to form a distribution over the layers, while $\textbf{MoL}_{hybrid}$ uses text-conditioned routing. We denote the improvement / degradation compared to the baseline model in green / red. We use a Vicuna-13B + CLIP backbone.

Model	*V**	GQA	TextVQA	MMMU
$\textbf{MoL}_{\text{image}}$	77.31 (+5.88)	62.94 (-0.32)	58.17 (-3.08)	36.00 (-0.89)
$\textbf{MoL}_{\text{hybrid}}$	**84.03** (+12.60)	**63.75** (+0.49)	**61.11** (-0.14)	**37.00** (+0.11)

Table 6: **Ablations on routing design.** (a) Removing the reserve layer degrades performance, indicating its importance as a fallback when routing is uncertain. In tandem, having an anchor (*reserve*) layer with complementary k layers helps MLLMs perform grounded fine-grained visual reasoning. (b) Varying k shows that selecting a small set of layers suffices for strong fine-grained reasoning.

Benchmark	$\textbf{MoL}_{\text{hybrid}}$	$\textbf{MoL}_{\text{hybrid}}$ No Reserve
*V**	82.35	78.57 (-3.78%)
HRBench8K	38.62	35.50 (-3.12%)
HRBench4K	43.88	38.75 (-5.13%)
POPE	86.50	82.10 (-4.40%)
MMMU	37.00	34.00 (-3.00%)

(a) Effect of removing the reserve layer.

Benchmark	$\textbf{MoL}_{\text{patch}}$				$\textbf{MoL}_{\text{hybrid}}$			
	$k=1$	$k=2$	$k=4$	Best	$k=1$	$k=2$	$k=4$	Best
*V**	76.89	73.95	79.41	79.41	84.03	76.89	67.65	84.03
HRBench8K	39.38	36.50	37.25	39.38	38.62	38.25	34.88	38.62
HRBench4K	43.12	42.75	43.75	43.75	43.88	44.00	43.50	44.00
MMStar	30.13	31.93	29.93	31.93	30.47	30.00	29.73	30.47
NaturalBench	69.41	68.07	68.61	69.41	69.96	68.91	70.45	70.45
CharXiv	28.65	28.47	27.94	28.65	26.13	26.25	26.97	26.97
RealWorldQA	55.42	54.90	55.16	55.42	55.03	56.60	56.34	56.60

(b) Effect of k on fine-grained visual reasoning.

contrast, the patch-level router ($\textbf{MoL}_{patch}$) places more mass on later layers. Since it computes routing independently for each patch, it can leverage semantically enriched representations that encode broader context, allowing individual regions to be interpreted relative to the overall scene and query. As a result, later-layer selections often reflect patches that benefit from contextualized, higher-level features. The hybrid router ($\textbf{MoL}_{hybrid}$) balances these two behaviors by combining global and patch-level routing, yielding more consistent mid-to-late layer usage. Overall, these patterns suggest that global routing favors detail-preserving features, while patch-level routing can exploit contextualized representations, motivating the need for both.

Receptive Field Analysis across Different Layers To gain a deeper understanding of the routing behavior in light of the above layer-wise sampling behiavor analysis, we analyze the receptive field (RF) of each layer using the average self-attention distance [12]: $\text{RF}^{\ell} = \mathbb{E}_i \left[\sum_j A^{\ell}_{i,j} d(i,j) \right]$. RF increases with depth: early layers capture local interactions, while later layers aggregate broader spatial context. This provides a mechanistic explanation for the routing patterns. $\textbf{MoL}_{\text{layer}}$ favors early, small-RF layers because global routing must rely on detail-preserving features that are consistently informative across the image. In contrast, $\textbf{MoL}_{\text{patch}}$ often selects later, large-RF layers, since patch-wise routing can leverage contextualized representations to interpret each region relative to the scene and query. $\textbf{MoL}_{\text{hybrid}}$ combines both regimes, enabling access to both local evidence and global context. Thus, RF analysis complements the routing observations by showing that **MoL** adapts to the inherent locality-context trade-off encoded across vision layers.

4.4 Ablations

Role of Vision Encoder-Aligned Text Encoders Table 5 studies whether routing should be conditioned on the text encoder aligned with the vision encoder. We compare $\textbf{MoL}_{image}$, which predicts layer-routing probabilities from vision hidden states alone, against the text-conditioned $\textbf{MoL}_{hybrid}$. Vision-only routing substantially underperforms the baseline, especially on *V**, where it drops by 9.16 points. We hypothesize that image-only routing lacks a task-specific signal for determining which layer representations are relevant, making the router more prone to selecting inconsistent or semantically mismatched layers during joint optimization. In contrast, $\textbf{MoL}_{hybrid}$ improves *V** by 13.44 points and yields small gains on GQA and MMMU while largely preserving TextVQA performance. These results suggest that layer selection should not be treated as an image-only decision: the input instruction provides an essential signal for selecting which visual abstractions are relevant for the current task.

Role of Reserve Layers and the Number of Layers Table 6 highlights two key properties of **MoL**. First, removing the reserve layer consistently degrades performance (3-5 points across benchmarks), indicating that it acts as a stable *anchor* that preserves strong, text-aligned representations when routing over intermediate layers is uncertain or noisy. Second, varying k shows that performance does not benefit from aggregating many layers; instead, small k values (e.g., $k = 1$ or $k = 4$) are sufficient and often optimal. Together, these results suggest that effective fine-grained visual reasoning arises from a balance between stability and selectivity: the reserve layer provides robustness, while sparse routing over a few task-relevant layers avoids introducing noise from irrelevant representations.

Qualitative Analysis on the Self-Attention Layers of Vision Encoders Figure 3 shows that **MoL** selects layers whose self-attention maps align with query-relevant regions. For localized tasks such as OCR and attribute recognition, mid-to-early layers (< layers 14-15) attend to fine-grained details (e.g., text on signs, small objects), while later layers highlight broader semantic regions (e.g., people, objects). For spatial reasoning queries, attention is distributed across multiple relevant regions, enabling the model to resolve relationships such as left/right positioning. We defer **Limitations and Broader Impacts** to Appendix A due to lack of space.

5 Conclusion

We introduced **Mixture of Layers** (**MoL**), an instruction-guided layer routing framework for MLLMs that dynamically selects intermediate vision-encoder representations for fine-grained visual reasoning conditioned on the input textual instruction. Instead of relying on fixed final-layer features or increasing the visual token budget, **MoL** selects task-relevant patch representations through layer-level, patch-level, and hybrid routing for both a single and dual vision encoder schemes. Across fine-grained visual understanding benchmarks, **MoL** consistently improves over backbone MLLMs and prior static feature-integration methods, while remaining competitive on conventional coarse-grained VQA tasks. Our analysis further shows that different routing strategies exploit different regions of the vision-encoder hierarchy: layer-level routing favors localized early layers, patch-level routing leverages contextualized later layers, and hybrid routing effectively combines both regimes to reap the benefit of both worlds. Overall, **MoL** demonstrates that dynamically selecting the appropriate visual abstraction can be a more effective alternative to scaling visual inputs or adding multiple vision encoders for improving MLLM perception and reasoning.

6 Acknowledgements

This research is based upon work supported by DARPA ITM Program No. FA8650-23-C-7316, the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via 560000C260018, CapitalOne-Illinois Center for Generative AI Safety, Knowledge Systems, and Cybersecurity (ASKS), Amazon-Illinois Center on AI for Interactive Conversational Experiences (AICE). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of DARPA, IARPA, ODNI, DARPA, or the U.S. Government. The U.S. Government is authorized

to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein. This research used the Delta and DeltaAI advanced computing and data resources, which are supported by the National Science Foundation (award OAC 2320345 and award OAC 2005572) and the State of Illinois. Delta and DeltaAI are joint efforts of the University of Illinois Urbana-Champaign and its National Center for Supercomputing Applications. The AI Research Institutes program by National Science Foundation and the Institute of Education Sciences, U.S. Department of Education through Award 2229873 - AI Institute for Transforming Education for Children with Speech and Language Processing Challenges.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[2] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[3] Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Rasheed, Junke Wang, Marco Monteiro, Hu Xu, Shiyu Dong, Nikhila Ravi, Daniel Li, Piotr Dollár, and Christoph Feichtenhofer. Perception encoder: The best visual embeddings are not at the output of the network, 2025.

[4] Yue Cao, Yangzhou Liu, Zhe Chen, Guangchen Shi, Wenhai Wang, Danhuai Zhao, and Tong Lu. Mmfuser: Multimodal multi-layer feature fuser for fine-grained vision-language understanding. *arXiv preprint arXiv:2410.11829*, 2024.

[5] Junbum Cha, Wooyoung Kang, Jonghwan Mun, and Byungseok Roh. Honeybee: Locality-enhanced projector for multimodal llm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13817–13827, 2024.

[6] Haoran Chen, Junyan Lin, Xinghao Chen, Yue Fan, Jianfeng Dong, Xin Jin, Hui Su, Jinlan Fu, and Xiaoyu Shen. Multimodal language models see better when they look shallower. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 6688–6706, 2025.

[7] Haozhe Chen, Junfeng Yang, Carl Vondrick, and Chengzhi Mao. INViTE: INterpret and control vision-language models with text explanations. In *The Twelfth International Conference on Learning Representations*, 2024.

[8] Yangyi Chen, Xingyao Wang, Hao Peng, and Heng Ji. Solo: A single transformer for scalable vision-language modeling. *Transactions on Machine Learning Research*, 2024, 2024.

[9] Yufeng Cui, Honghao Chen, Haoge Deng, Xu Huang, Xinghang Li, Jirong Liu, Yang Liu, Zhuoyan Luo, Jinsheng Wang, Wenxuan Wang, et al. Emu3. 5: Native multimodal models are world learners. *arXiv preprint arXiv:2510.26583*, 2025.

[10] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. InstructBLIP: Towards general-purpose vision-language models with instruction tuning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[11] Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, et al. Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 91–104, 2025.

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[13] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

[14] Yossi Gandelsman, Alexei A Efros, and Jacob Steinhardt. Interpreting clip's image representation via text-based decomposition. In *The Twelfth International Conference on Learning Representations*, 2024.

[15] Amin Ghiasi, Hamid Kazemi, Eitan Borgnia, Steven Reich, Manli Shu, Micah Goldblum, Andrew Gordon Wilson, and Tom Goldstein. What do vision transformers learn? a visual exploration. *arXiv preprint arXiv:2212.06727*, 2022.

[16] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[17] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[18] Liqi He, Zuchao Li, Xiantao Cai, and Ping Wang. Multi-modal latent space learning for chain-of-thought reasoning in language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18180–18187, 2024.

[19] Muyang He, Yexin Liu, Boya Wu, Jianhao Yuan, Yueze Wang, Tiejun Huang, and Bo Zhao. Efficient multimodal learning from data-centric perspective. *arXiv preprint arXiv:2402.11530*, 2024.

[20] Greg Heinrich, Mike Ranzinger, Hongxu Yin, Yao Lu, Jan Kautz, Andrew Tao, Bryan Catanzaro, and Pavlo Molchanov. Radiov2. 5: Improved baselines for agglomerative vision foundation models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22487–22497, 2025.

[21] Geoffrey E Hinton. Products of experts. In *Proceedings of the Ninth International Conference on Artificial Neural Networks (ICANN'99)*, volume 1, pages 1–6. IEE, 1999.

[22] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709, 2019.

[23] Dongsheng Jiang, Yuchen Liu, Songlin Liu, XIAOPENG ZHANG, Jin Li, Hongkai Xiong, and Qi Tian. From CLIP to DINO: Visual encoders shout in multi-modal large language models, 2024.

[24] Cijo Jose, Théo Moutakanni, Dahyun Kang, Federico Baldassarre, Timothée Darcet, Hu Xu, Daniel Li, Marc Szafraniec, Michaël Ramamonjisoa, Maxime Oquab, et al. Dinov2 meets text: A unified framework for image-and pixel-level vision-language alignment. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24905–24916, 2025.

[25] Jing Yu Koh, Ruslan Salakhutdinov, and Daniel Fried. Grounding language models to images for multimodal inputs and outputs. In *International Conference on Machine Learning*, pages 17283–17300. PMLR, 2023.

[26] Baiqi Li, Zhiqiu Lin, Wenxuan Peng, Jean de Dieu Nyandwi, Daniel Jiang, Zixian Ma, Simran Khanuja, Ranjay Krishna, Graham Neubig, and Deva Ramanan. Naturalbench: Evaluating vision-language models on natural adversarial samples. *Advances in Neural Information Processing Systems*, 37:17044–17068, 2024.

[27] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-onevision: Easy visual task transfer. *Transactions on Machine Learning Research*, 2025.

[28] 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*, pages 19730–19742. PMLR, 2023.

[29] Wentong Li, Yuqian Yuan, Jian Liu, Dongqi Tang, Song Wang, Jie Qin, Jianke Zhu, and Lei Zhang. Tokenpacker: Efficient visual projector for multimodal llm. *International Journal of Computer Vision*, 133(10):6794–6812, 2025.

[30] Xu Li, Yi Zheng, Haotian Chen, Xiaolei Chen, Yuxuan Liang, Chenghang Lai, Bin Li, and Xiangyang Xue. Instruction-guided fusion of multi-layer visual features in large vision-language models. *Pattern Recognition*, 170:111932, 2026.

[31] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 292–305, 2023.

[32] Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. Textbooks are all you need ii: phi-1.5 technical report. *arXiv preprint arXiv:2309.05463*, 2023.

[33] Chenchen Lin, Sanbao Su, Rachel Luo, Yuxiao Chen, Yan Wang, Marco Pavone, and Fei Miao. Text-guided layer fusion mitigates hallucination in multimodal llms. *arXiv preprint arXiv:2601.03100*, 2026.

[34] Junyan Lin, Haoran Chen, Yue Fan, Yingqi Fan, Xin Jin, Hui Su, Jinlan Fu, and Xiaoyu Shen. Multi-layer visual feature fusion in multimodal llms: Methods, analysis, and best practices. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 4156–4166, 2025.

[35] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26296–26306, 2024.

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

[37] Lingchen Meng, Jianwei Yang, Rui Tian, Xiyang Dai, Zuxuan Wu, Jianfeng Gao, and Yu-Gang Jiang. Deepstack: Deeply stacking visual tokens is surprisingly simple and effective for lmms. *Advances in Neural Information Processing Systems*, 37:23464–23487, 2024.

[38] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

[39] Namuk Park, Wonjae Kim, Byeongho Heo, Taekyung Kim, and Sangdoo Yun. What do self-supervised vision transformers learn? In *The Eleventh International Conference on Learning Representations*, 2023.

[40] 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*, pages 8748–8763. PMLR, 2021.

[41] 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. In *International Conference on Learning Representations*, 2017.

[42] Min Shi, Fuxiao Liu, Shihao Wang, Shijia Liao, Subhashree Radhakrishnan, Yilin Zhao, De-An Huang, Hongxu Yin, Karan Sapra, Yaser Yacoob, Humphrey Shi, Bryan Catanzaro, Andrew Tao, Jan Kautz, Zhiding Yu, and Guilin Liu. Eagle: Exploring the design space for multimodal LLMs with mixture of encoders. In *The Thirteenth International Conference on Learning Representations*, 2025.

[43] Amanpreet Singh, Vivek Natarjan, Meet Shah, Yu Jiang, Xinlei Chen, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8317–8326, 2019.

[44] Adam Steel, Brenda D. Garcia, Kala Goyal, Anna Mynick, and Caroline E. Robertson. Scene perception and visuospatial memory converge at the anterior edge of visually responsive cortex. *Journal of Neuroscience*, 43(31):5723–5737, 2023.

[45] Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai C Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *Advances in Neural Information Processing Systems*, 37:87310–87356, 2024.

[46] Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9568–9578, June 2024.

[47] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[48] Wenbin Wang, Liang Ding, Minyan Zeng, Xiabin Zhou, Li Shen, Yong Luo, Wei Yu, and Dacheng Tao. Divide, conquer and combine: A training-free framework for high-resolution image perception in multimodal large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 7907–7915, 2025.

[49] Zirui Wang, Mengzhou Xia, Luxi He, Howard Chen, Yitao Liu, Richard Zhu, Kaiqu Liang, Xindi Wu, Haotian Liu, Sadhika Malladi, et al. Charxiv: Charting gaps in realistic chart understanding in multimodal llms. *Advances in Neural Information Processing Systems*, 37:113569–113697, 2024.

[50] Penghao Wu and Saining Xie. V?: Guided visual search as a core mechanism in multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13084–13094, 2024.

[51] xAI. Realworldqa: A real-world reasoning benchmark for multimodal llms, 2024.

[52] Huanjin Yao, Wenhao Wu, Taojiannan Yang, YuXin Song, Mengxi Zhang, Haocheng Feng, Yifan Sun, Zhiheng Li, Wanli Ouyang, and Jingdong Wang. Dense connector for mllms. *Advances in Neural Information Processing Systems*, 37:33108–33140, 2024.

[53] Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*, 2023.

[54] 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*, pages 9556–9567, 2024.

[55] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986, 2023.

[56] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. MiniGPT-4: Enhancing vision-language understanding with advanced large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

Appendix

How to read this appendix. Section A states the scope, limitations, and broader impacts of the work. Sections B–C document how the models and evaluations were run. Sections D–F provide the extended empirical evidence behind the main-paper results. The links in the roadmap jump directly to each section.

Appendix Section	Contents
A. Limitations and Broader Impacts	Scope of claims, methodological limitations, expected benefits, risks, and responsible-use considerations.
B. Training Details	Hardware, two-stage training, optimization settings, objective, and router naming.
C. Evaluation Details	Benchmark scripts, output files, score aggregation, and missing-entry conventions.
D. Additional Benchmark Results	Full benchmark matrix, k sweep, and reserve-layer ablation.
E. Router and RF Analyses	Layer-selection behavior, receptive-field overlays, router usage diagnostics, and DINOv2 placeholders.
F. Qualitative Examples	Selected attention panels plus a LLaVA + DINOv2 heatmap analysis.
G. Reproducibility Notes	Remaining caveats, source artifacts, and checkpoint-config conventions.

A Limitations and Broader Impacts

Scope of the empirical claims. **MIXTURE OF LAYERS** is designed to improve the visual representation interface between a pretrained vision encoder and a multimodal language model by routing over intermediate vision-layer features. The central claim is therefore not that **MoL** universally improves every multimodal task, but that instruction-conditioned access to task-relevant intermediate representations improves fine-grained visual reasoning while largely preserving standard VQA performance. This distinction matters: the strongest gains in our experiments appear on benchmarks that require localized evidence, such as V*Bench, CharXiv, HRBench, OCR-style questions, direct attribute recognition, and relative-position reasoning. On more conventional or coarse-grained VQA benchmarks, the gains are smaller and sometimes close to neutral. We view this as a useful boundary on the contribution rather than a weakness of the evaluation: **MoL** is targeted at perception bottlenecks where the final vision layer can discard details needed by the language model.

Architectural limitations. **MoL** assumes access to the hidden states of a pretrained vision encoder and, for the text-conditioned variants, a text encoder aligned with that vision tower. This makes the method natural for open and reproducible MLLM training pipelines such as LLaVA- and Bunny-style models, but less directly applicable to vision encoder-free MLLM architectures [8, 9] or models that expose only final visual embeddings. The router also introduces additional trainable parameters, routing hyperparameters such as k, and a reserve-layer design choice. Our ablations show that a small number of selected layers and the reserve layer are important, but these choices may need to be retuned for substantially different vision encoders, tokenization schemes, resolutions, or downstream domains.

Evaluation limitations. The experiments cover several backbones, parameter scales, and vision encoders, including CLIP, SigLIP, and DINOv2 w/ Txt, but they do not exhaust the space of modern MLLM architectures. We do not claim that the exact best router variant transfers unchanged to every backbone: $\textbf{MoL}_{\text{patch}}$ tends to help localized perceptual categories, while $\textbf{MoL}_{\text{hybrid}}$ is more robust when both local grounding and broader context are needed. We also report single-run results for most large-scale training settings because training 8B–13B multimodal models is computationally expensive. The breadth of benchmarks and ablations partially mitigates this limitation, but future work should add repeated runs, stronger uncertainty estimates, and evaluation on additional domains such as video, medical imagery, remote sensing, and document-heavy workflows.

Interpretability limitations. Router probabilities, receptive-field overlays, and selected-layer attention maps provide useful diagnostics for understanding how **MoL** uses intermediate representations. They should not, however, be interpreted as complete causal explanations of the model's answer. A selected layer may contribute useful features even when its attention map is diffuse, and visually plausible attention can still coincide with an incorrect answer. We therefore use these analyses as evidence about routing behavior and locality-context trade-offs, not as a guarantee that the final language-model reasoning is faithful or safe.

Broader positive impacts. Improved fine-grained visual reasoning can benefit applications that require careful visual grounding, such as assistive image understanding, chart and figure interpretation, educational tools, accessibility support, and scientific document analysis. A practical advantage of **MoL** is that it can improve access to useful visual abstractions without necessarily increasing the number of visual tokens or requiring multiple vision encoders. This may offer a more compute-conscious path than simply scaling image resolution, concatenating additional towers, or passing substantially longer visual sequences to the language model.

Risks and misuse. The same fine-grained perception improvements can also increase risks when applied to sensitive visual data. Better OCR, attribute recognition, and small-object grounding may make it easier to extract private information from images, analyze people or locations without consent, or strengthen surveillance-style applications. **MoL** also does not remove inherited biases from the underlying language model, vision encoder, or instruction-tuning data. Since the method improves perception rather than safety alignment, it should not be treated as sufficient for high-stakes deployment in domains such as medical diagnosis, legal decision-making, hiring, biometric identification, or safety-critical robotics.

Backbone	Stage	Data	ZeRO	Epochs	Global Batch	Projector LR	Router LR
Vicuna-13B + CLIP	Pretrain	BLIP/LAION-CC-SBU 558K	2	1	256	1×10^{-3}	1×10^{-5}–5×10^{-5}
Vicuna-13B + CLIP	Finetune	LLaVA-v1.5 mix 665K	3	1	128	2×10^{-5}	2×10^{-5}
Llama-3-8B + SigLIP	Pretrain	Bunny pretrain 2M	2	1	128	1×10^{-3}	1×10^{-4}
Llama-3-8B + SigLIP	Finetune	Bunny instruction 695K	3	1	128	2×10^{-5}–2×10^{-4}	2×10^{-5}–2×10^{-4}
Phi-1.5 + SigLIP	Pretrain	Bunny pretrain 2M	2	1	256	5×10^{-4}	5×10^{-4}
Phi-1.5 + SigLIP	Finetune	Bunny instruction 695K	3	1	128	5×10^{-5}–2×10^{-4}	5×10^{-5}–2×10^{-4}

Table 7: **Training setup.** The Vicuna-13B + CLIP setting follows the LLaVA-v1.5 two-stage training recipe. The Llama-3-8B + SigLIP and Phi-1.5 + SigLIP settings follow the Bunny two-stage recipe and use LoRA during instruction tuning.

Variant	Implementation key	Routing granularity	k	Reserve layer	Reserve mode	Text conditioning
$\textbf{MoL}_{\text{layer}}$	`attn-token`	image-level layer set	1–4	-2	local or scaled gate	attention
$\textbf{MoL}_{\text{patch}}$	`patch_layer`	patch-specific layer set	1–4	-2	local gate	attention
$\textbf{MoL}_{\text{hybrid}}$	`hybrid`	global-local fused routing	1–4	-2	disagreement gate	attention

Table 8: **Router configuration summary.** The reserve layer is the penultimate vision-encoder layer used as an anchor representation. Appendix tables use normalized method names rather than implementation keys.

Responsible-use considerations. We frame **MoL** as a representation-routing method for research on multimodal perception, not as a complete deployment system. Systems built on top of this work should include privacy-aware data handling, domain-specific evaluation, bias and robustness audits, and human oversight in high-stakes settings. Releasing training details, normalized router names, checkpoint-configuration conventions, and analysis scripts is intended to make the empirical claims easier to audit and to help future work identify both successes and failure modes rather than treating improved benchmark scores as a blanket guarantee of reliability.

B Training Details

All models were trained on a single server with four NVIDIA H100 NVL GPUs, each with 95830 MiB of device memory. We use DeepSpeed ZeRO-2 for the vision-language connector pretraining stage and DeepSpeed ZeRO-3 for the multimodal instruction-tuning stage. Unless otherwise stated, training uses bf16, tf32, gradient checkpointing, a cosine learning-rate schedule, warmup ratio 0.03, weight decay 0, maximum sequence length 2048, and lazy image preprocessing.

The training objective is the standard autoregressive language-modeling loss with an auxiliary router load-balancing loss:

$$\mathcal{L} = \mathcal{L}_{\text{LM}} + \lambda_{\text{lb}}\mathcal{L}_{\text{lb}}. \tag{11}$$

The default load-balancing coefficient is $\lambda_{\text{lb}} = 10^{-2}$. During training, the router temperature is linearly annealed from 2.0 to 1.0. We save router statistics during training for the analysis in Appendix E.

Router naming. We group the `attn-token` implementation variants as $\textbf{MoL}_{\text{layer}}$, and report the better-performing layer-level variant for each setting. The `patch_layer` implementation corresponds to $\textbf{MoL}_{\text{patch}}$, and the `hybrid` implementation corresponds to $\textbf{MoL}_{\text{hybrid}}$.

Table 7 summarizes the hardware and optimization recipe used for each model family. The main distinction is between the Vicuna/LLaVA training path, which follows the LLaVA-v1.5 data mixture, and the Bunny training path, which uses the Bunny pretraining and instruction-tuning data for the Llama-3 and Phi backbones. We report global batch sizes after accounting for the four GPUs and gradient accumulation.

Table 8 gives the naming bridge between implementation keys in checkpoints and the method names used throughout the paper. This is important because the experiments include multiple layer-level router implementations. We treat them as the same conceptual family, $\textbf{MoL}_{\text{layer}}$, when summarizing broad benchmark results.

C Evaluation Details

We evaluate with benchmark-specific scripts under `scripts/v1_5/eval`. Each run writes predictions, per-example details, and a summary file under

Method	MMBench	MMMU	MMStar	MMVet	MMVP	CV-Bench	NaturalBench	CharXiv	CountBenchQA	GQA	POPE	TextVQA
LLaVA-v1.5-13B	76.49	36.89	30.00	41.11	65.33	62.60	68.93	25.20	50.71	63.26	85.60	61.25
$\textbf{MoL}_{\text{layer}}$	75.81	36.56	29.53	–	64.67	57.88	69.71	27.85	48.68	63.43	86.53	60.64
$\textbf{MoL}_{\text{patch}}$	76.40	38.22	31.93	–	65.00	63.28	69.41	30.89	48.68	63.73	86.23	61.09
$\textbf{MoL}_{\text{hybrid}}$	75.94	37.00	31.07	41.88	65.67	62.46	70.45	26.97	51.93	63.75	86.47	61.11

Table 9: **Full benchmark matrix for the primary LLaVA-v1.5 family, part 1.** $\textbf{MoL}_{\text{layer}}$ is the per-metric best of the `attn-layer` and `attn-token` layer-level implementations. $\textbf{MoL}_{\text{patch}}$ and $\textbf{MoL}_{\text{hybrid}}$ are the best available scores within their normalized method families.

Method	DocVQA	MME-RW	HallusionBench	RealWorldQA	ChartQA	HRBench4K	HRBench8K	V*Bench	DTD
LLaVA-v1.5-13B	23.98	32.33	51.64	55.29	20.28	43.75	36.88	71.43	35.41
$\textbf{MoL}_{\text{layer}}$	24.68	30.39	55.09	55.82	20.56	41.38	38.00	76.89	31.94
$\textbf{MoL}_{\text{patch}}$	25.28	33.18	52.97	55.42	21.08	43.75	39.38	84.87	–
$\textbf{MoL}_{\text{hybrid}}$	25.21	32.34	53.23	56.60	22.44	44.00	38.62	84.03	–

Table 10: **Full benchmark matrix for the primary LLaVA-v1.5 family, part 2.** MME-RW denotes MME-RealWorld. Empty entries indicate unavailable scores in the generated benchmark table.

Benchmark	$\textbf{MoL}_{\text{patch}}$				$\textbf{MoL}_{\text{hybrid}}$			
	$k=1$	$k=2$	$k=4$	Best	$k=1$	$k=2$	$k=4$	Best
V*Bench	76.89	73.95	79.41	79.41	84.03	76.89	67.65	84.03
HRBench8K	39.38	36.50	37.25	39.38	38.62	38.25	34.88	38.62
HRBench4K	43.12	42.75	43.75	43.75	43.88	44.00	43.50	44.00
MMStar	30.13	31.93	29.93	31.93	30.47	30.00	29.73	30.47
NaturalBench	69.41	68.07	68.61	69.41	69.96	68.91	70.45	70.45
CharXiv	28.65	28.47	27.94	28.65	26.13	26.25	26.97	26.97
RealWorldQA	55.42	54.90	55.16	55.42	55.03	56.60	56.34	56.60

Table 11: **Fine-grained visual reasoning under different k values.** A small selected set of layers is usually sufficient; the best k varies by task and routing family.

`playground/data/eval/BENCHMARK/answers`. Unless the benchmark defines a specialized metric, we report accuracy. For V*Bench, HRBench4K/8K, MMStar, NaturalBench, RealWorldQA, CharXiv, and the conventional VQA benchmarks, scores are percentages. For benchmarks with multiple splits or categories, the reported number is the official aggregate used by the corresponding evaluation script.

The broad benchmark matrix in Appendix D is generated from `playground/data/eval/benchmark_overall_table.md`. Empty entries indicate that the run was not available in the generated table. We omit columns that were entirely empty for the selected LLaVA-v1.5 family rows.

D Additional Benchmark Results

Tables 9 and 10 expand the main-paper results to a broader set of benchmarks for the primary LLaVA-v1.5 family. The table is split only for readability. These numbers are intended to show whether **MoL** preserves general multimodal ability while improving fine-grained visual reasoning. The strongest gains concentrate on fine-grained benchmarks such as V*Bench, CharXiv, HRBench, and MMStar, while conventional VQA-style benchmarks usually remain close to the baseline.

Table 11 isolates the effect of the number of selected layers. The main pattern is that selecting more layers is not automatically better. A small k often gives the best result, which supports the interpretation that the router should selectively retrieve a few useful abstraction levels instead of averaging many layers indiscriminately.

Table 12 tests the reserve layer directly. Removing it hurts V*Bench, HRBench, POPE, and MMMU, indicating that the reserve representation acts as a stable fallback when the routed intermediate layers are noisy or uncertain.

Benchmark	$\textbf{MoL}_{\text{hybrid}}$	$\textbf{MoL}_{\text{hybrid}}$ No Reserve
V*Bench	82.35	78.57 (-3.78)
HRBench8K	38.62	35.50 (-3.12)
HRBench4K	43.88	38.75 (-5.13)
POPE	86.50	82.10 (-4.40)
MMMU	37.00	34.00 (-3.00)

Table 12: **Reserve-layer ablation.** Removing the reserve layer consistently hurts performance, supporting its role as a stable anchor representation.

E Additional Router and Receptive Field Analyses

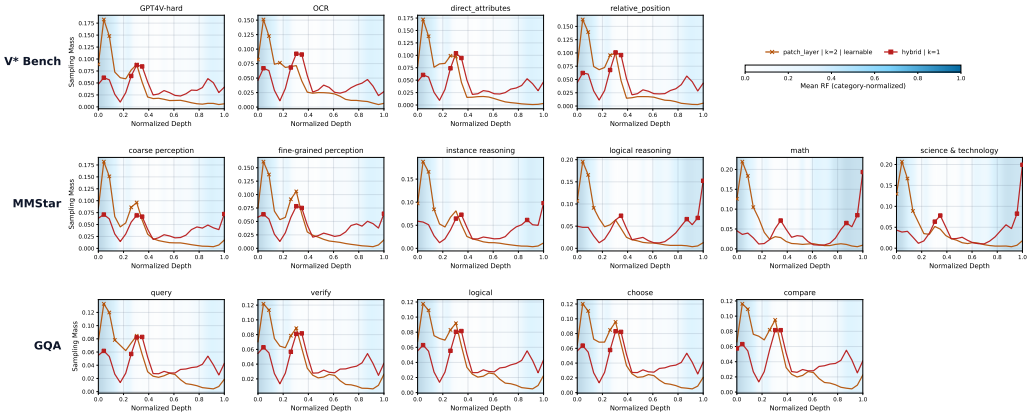


Figure 4: **DINOv2 layer sampling with receptive-field background on V*Bench, MMStar, and GQA.** Curves show the normalized routing probability assigned to each DINOv2 vision-encoder layer for $\textbf{MoL}_{\text{patch}}$ and $\textbf{MoL}_{\text{hybrid}}$ across question categories. Background intensity denotes the category-normalized receptive field (RF), where lighter regions indicate more localized layers and darker regions indicate broader spatial context. Type-wise markers indicate the selected top-k layers.

DINOv2 RF-background Figure 4 shows that DINOv2 exhibits category-dependent routing patterns consistent with the CLIP analysis in the main paper. $\textbf{MoL}_{\text{patch}}$ often assigns high probability to early layers, especially for V*Bench and perceptual MMStar/GQA categories, suggesting that localized DINOv2 features remain useful for fine-grained cues such as OCR, attributes, and relative position. In contrast, $\textbf{MoL}_{\text{hybrid}}$ places more mass on mid-to-late layers for categories requiring broader reasoning, such as logical reasoning, math, and science/technology in MMStar. This supports the role of hybrid routing: it preserves access to local evidence while selectively relying on deeper, larger-RF layers when the query requires more semantic or relational context.

F Additional Qualitative Examples

Qualitative Analysis on the DINOv2 Attention Maps Figure 5 shows that **MoL** with a DINOv2 encoder consistently attends to localized, query-relevant regions for fine-grained visual reasoning. For OCR-heavy examples (e.g., reading the conference name on a mug or identifying text on a banner), the selected layers concentrate attention tightly around the relevant textual regions, despite cluttered backgrounds and small font sizes. In counting and attribute-centric examples, such as identifying the jersey number of a basketball player or recognizing the brand logo on a bag, the router selects layers whose attention maps isolate the discriminative object regions while suppressing surrounding distractions. Interestingly, even in semantically broader scenes (e.g., identifying the instrument held by the ape), the selected layers maintain strong localized grounding on the queried object rather than diffusing attention across the full scene. These patterns suggest that **MoL** with DINOv2 learns to exploit intermediate-layer representations that preserve both fine-grained local structure and semantically meaningful object boundaries, enabling grounded visual reasoning without requiring additional visual tokens or multiple encoders.

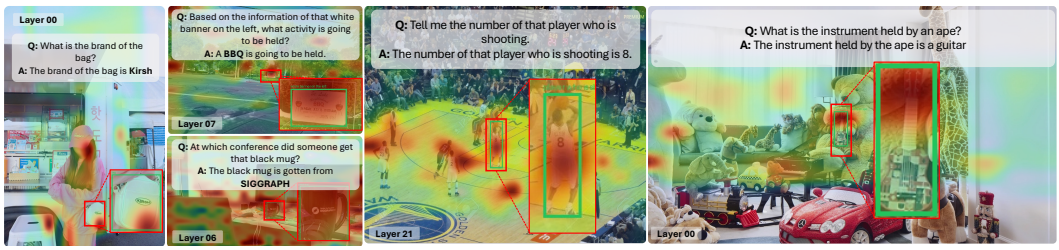


Figure 5: **Self-attention heatmap visualization of the selected layers aggregated across all the queries of the layers for DINOv2 encoder for the selected layers.**

G Reproducibility Notes

The appendix reports the generated result artifacts available under `playground/data/eval`. Because the main experiments train large MLLMs, we do not report multiple random seeds for every model and benchmark. We instead prioritize breadth across backbones, vision encoders, routing variants, and benchmarks. Checkpoint `config.json` files are treated as the source of truth for router mode, k, reserve-layer settings, and learning-rate metadata when run-script names and final checkpoint labels differ.

For reproducibility, final paper figures used by this appendix are copied into `paper_draft_latex/figures/appendix`. The source analysis artifacts remain in `playground/data/eval`, including the benchmark matrix, router/RF plots, reserve comparisons, and resolution analyses.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes].

   Justification: The abstract and introduction state the paper's main contribution, **MIXTURE OF LAYERS**, and scope the claims around fine-grained visual reasoning. The reported claims are supported by the experimental results in Section 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

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

   Answer: [Yes].

   Justification: We discuss limitations in the Appendix, including the empirical scope of the evaluation and remaining directions for extending **MIXTURE OF LAYERS** to broader backbones, modalities, and settings.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions.
   - The authors should reflect on the scope of the claims made.
   - The authors should reflect on the factors that influence the performance of the approach.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

   Answer: [N/A].

   Justification: This paper does not present theoretical results or formal proofs.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.

- The proofs can either appear in the main paper or the supplemental material.
- Any informal proof provided in the core of the paper should be complemented by formal proofs.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper?

   Answer: [Yes].

   Justification: We describe the model architecture, routing variants, training setup, datasets, and evaluation benchmarks in the main paper and Appendix. We will release code and instructions to reproduce the main experimental results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Reproducibility can be provided via code, data, instructions, model checkpoints, hosted models, or other appropriate means.
   - NeurIPS requires submissions to provide some reasonable avenue for reproducibility.

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

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

   Answer: [Yes].

   Justification: All datasets used in the paper are publicly available, and we will release the codebase with instructions for training and evaluation. At submission time, any released materials will be anonymized as required.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines for more details.
   - While code and data release is encouraged, [No] is acceptable with justification.
   - The instructions should contain the exact command and environment needed to reproduce the results.
   - The authors should provide instructions on data access and preparation.
   - The authors should provide scripts to reproduce all experimental results.
   - At submission time, released materials should be anonymized if applicable.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details necessary to understand the results?

   Answer: [Yes].

   Justification: The main paper describes the backbone models, vision encoders, datasets, benchmarks, and routing variants. Additional training details, hyperparameters, and evaluation settings are provided in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail necessary to appreciate the results.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

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

   Answer: [No].

   Justification: We do not report statistical significance tests or error bars. The experiments are computationally expensive, and our comparisons follow standard benchmark evaluation protocols for MLLMs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests.
   - The factors of variability captured by the error bars should be clearly stated.
   - The method for calculating the error bars should be explained.
   - The assumptions made should be given.
   - It should be clear whether the error bar is standard deviation or standard error.
   - For asymmetric distributions, authors should avoid inappropriate symmetric error bars.
   - If error bars are reported, the text should explain how they were calculated.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources needed to reproduce the experiments?

   Answer: [Yes].

   Justification: We provide compute resource details, including GPU type, memory, and training/evaluation cost, in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers, including relevant memory and storage.
   - The paper should provide the amount of compute required for individual runs and estimate total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics?

   Answer: [Yes].

   Justification: The research uses publicly available datasets and models, does not involve human subjects, and does not require deviations from the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation.
   - The authors should make sure to preserve anonymity.

10. **Broader impacts**

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

    Answer: [N/A].

Justification: This work is foundational research on visual representation routing in MLLMs and is not tied to a specific deployment or application domain. We do not identify a direct societal impact beyond the general impacts associated with improving multimodal model capabilities.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why.
- Examples of negative societal impacts include malicious uses, fairness considerations, privacy considerations, and security considerations.
- Foundational research may not need application-specific broader impact discussion unless there is a direct path to harm.
- Authors should consider harms from intended use, incorrect results, and misuse.
- If there are negative societal impacts, authors may discuss mitigation strategies.

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse?

Answer: [N/A].

Justification: The paper does not release a new high-risk dataset or pretrained generative model. The released code is intended to reproduce the proposed routing method and experiments using existing public assets.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models with high misuse risk should be released with necessary safeguards.
- Datasets scraped from the Internet could pose safety risks.
- Authors are encouraged to make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets used in the paper properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes].

Justification: We cite the original papers for all datasets, models, and codebases used in this work. We use publicly available assets and will document their licenses and terms of use in the Appendix or released codebase.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license should be included for each asset.
- For scraped data, copyright and terms of service should be provided.
- If assets are released, license, copyright information, and terms of use should be provided.
- For repackaged datasets, both original and derived licenses should be provided.
- If license information is unavailable, authors are encouraged to contact asset creators.

13. **New assets**

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

Answer: [Yes].

Justification: We introduce and release the **MIXTURE OF LAYERS** codebase. The released asset will include documentation, setup instructions, training and evaluation scripts, and license information.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate details of the dataset/code/model through documentation.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, assets should be anonymized if applicable.

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

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

    Answer: [N/A].

    Justification: This paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in supplemental material is fine.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least minimum wage.

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

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether IRB approvals were obtained?

    Answer: [N/A].

    Justification: This paper does not involve crowdsourcing or research with human subjects, so IRB approval is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - IRB approval or equivalent may be required depending on the country or institution.
    - Authors should adhere to the NeurIPS Code of Ethics and institutional guidelines.
    - For initial submissions, do not include information that would break anonymity.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research?

    Answer: [N/A].

    Justification: The paper studies MLLM architectures, but LLMs are not used as a non-standard research assistant or data-generation component in the core methodology. Any use of LLMs for writing, editing, or formatting does not affect the scientific contribution.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to the LLM policy in the NeurIPS handbook for what should or should not be described.