IP-MoE: Studying Spatial Token–Patch Interactions and Dense Expert Weighting

Anonymous authors
Paper under double-blind review

Abstract

Mixture-of-Experts (MoE) architectures provide a mechanism for combining expert computation through input-dependent weighting. In vision–language models, however, it remains unclear whether explicit interactions between visual and textual features provide useful information for determining these weights. We investigate this question with **IP-MoE**, a soft dense mixture of experts conditioned on token–patch interactions. IP-MoE constructs a spatial interaction map from question-token and image-patch representations while preserving the image layout, and uses it to assign soft weights to global, local, and hybrid experts. Their weighted outputs, together with image and interaction features, are projected into prefix tokens for a causal language model. We evaluate IP-MoE on seven benchmarks spanning visual question answering, spatial perception, object hallucination, general vision–language abilities, and vision-indispensable reasoning. We further compare bilinear and sum-based interactions, as well as learned and uniform expert weighting. In the evaluated configuration, the model trained with learned expert weighting obtains higher scores on 2D spatial perception and object hallucination, but lower scores on 3D spatial perception, than the model trained with uniform weights. Bilinear interactions likewise yield task-dependent differences relative to sum-based interactions. Training diagnostics indicate nearly uniform gates late in training, and restricting inference to individual expert families produces modest changes in aggregate scores. These findings characterize the behavior and limitations of interaction-conditioned expert weighting, without establishing consistent performance gains or functional expert specialization.

1 Introduction

Leading vision–language models have demonstrated strong performance across a broad range of perception and reasoning tasks. Scaling model size and training data has been an important part of this progress, with corresponding demands on training and deployment resources (Nezhurina et al., 2025; Cai et al., 2025).

Mixture-of-Experts (MoE) architectures distribute computation across multiple networks. Sparse MoE activates a subset of feed-forward or attention networks for each input (Shazeer et al., 2017; Zhang et al., 2022). V-MoE extends routing to vision models, while VLMo explores modality-specific experts for vision–language representation learning (Riquelme et al., 2021; Bao et al., 2022). Routing stability and expert utilization remain important considerations (Chi et al., 2022; Puigcerver et al., 2024). Soft MoE forms expert inputs through continuous weighted combinations of tokens (Puigcerver et al., 2024).

Bilinear models offer a direct formulation of such feature relationships through multiplicative interactions (Tenenbaum & Freeman, 2000). Their applications include visual recognition and vision–language fusion, with compact formulations based on projections, Hadamard products, and tensor decompositions (Lin et al., 2015; Fukui et al., 2016; Kim et al., 2017; Ben-younes et al., 2017; 2019). Applied at the token–patch level, bilinear interactions combine question information with the visual features of each image patch. We hypothesize that the resulting representation can help the router assign expert weights based on the visual information needed to answer the question.

We study IP-MoE, a soft dense MoE system built around a question-conditioned spatial interaction map. Projected question-token and image-patch features are combined and aggregated over question

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

tokens separately at each patch, preserving the image layout. Global, local, and hybrid experts transform this map, while a router assigns one soft weight per expert for each example. Image features, interaction features, and weighted expert outputs form three prefix blocks for a causal language model. This design makes the interaction representation available to expert processing, expert weighting, and direct language-model conditioning.

Our empirical objective is to characterize how interaction and weighting choices affect the observed behavior of this system. We compare three complete training configurations: bilinear interactions with learned weights, bilinear interactions with uniform weights, and sum-based interactions with learned weights. Each configuration is trained once with seed 42 and evaluated across seven benchmarks. We examine their aggregate scores, task-level differences, answer patterns, and recorded training-gate behavior.

The evaluated configurations show no consistent performance ordering across benchmarks. More detailed analyses qualify the aggregate comparisons: higher CV-Bench-2D accuracy coexists with lower spatial-relation accuracy, and higher POPE F1 accompanies higher recall but lower precision. CV-Bench-3D predictions exhibit substantial answer-option imbalance, while both learned configurations approach nearly uniform weights in late-training measurements. Together, these observations characterize the behavior of the fitted systems and clarify the interpretation of their benchmark differences. Because the comparisons change complete computational or optimization configurations, they leave the separate contribution of adaptive routing unresolved.

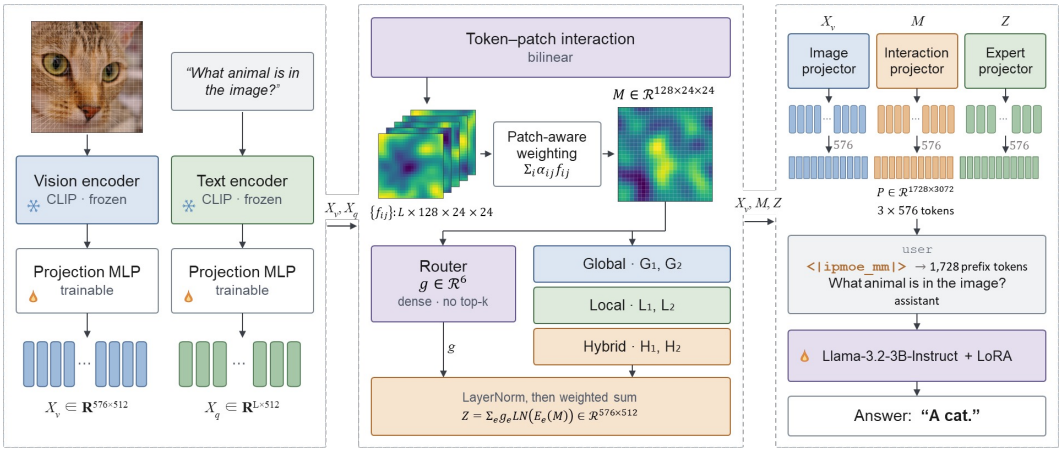
Our contributions are summarized as follows:

- We present IP-MoE, a soft dense vision–language method that supplies a question-conditioned spatial interaction map to global, local, and hybrid experts, an example-level router, and the language model.
- We report comparisons across seven benchmarks, showing task-dependent configuration differences and opposing changes within aggregate metrics. Paired bootstrap intervals quantify conditional evaluation-sample uncertainty where benchmark grouping is recoverable.
- We characterize the recorded gate trajectories and answer patterns, documenting near-uniform late-training weights under both interaction operators and substantial CV-Bench-3D answer-option imbalance. These diagnostics qualify interpretations of aggregate performance without establishing functional expert specialization or a routing-specific effect.

2 Related Work

Vision–Language Models and Mixture of Experts Transformer-based language models and pretrained vision encoders such as CLIP underpin modern vision–language systems (Vaswani et al., 2017; Radford et al., 2021). BLIP-2 connects a frozen image encoder and a frozen language model through a querying transformer, while InstructBLIP conditions visual feature extraction on instructions (Li et al., 2023a; Dai et al., 2023). Sparse MoE increases model capacity through conditional activation of feed-forward or attention experts (Shazeer et al., 2017; Zhang et al., 2022). V-MoE applies expert routing to vision models, while VLMo introduces modality-specific experts for vision–language representation learning (Riquelme et al., 2021; Bao et al., 2022). MolmoE uses a sparse MoE language backbone, while Mono-InternVL incorporates visual experts into a pretrained language model (Deitke et al., 2025; Luo et al., 2025). MoVA selects and combines vision experts according to the image and instruction, and MoME combines adaptive visual-feature modulation with sparse language experts (Zong et al., 2024; Shen et al., 2024). Other studies examine adaptive expert activation, expert merging and compression, clustering-based routing, and the effects of expert count on memorization and reasoning (Guo et al., 2025; Li et al., 2024c; Nielsen et al., 2025; Jelassi et al., 2025). Soft MoE constructs expert inputs from weighted combinations of tokens and recombines their outputs (Puigcerver et al., 2024). IP-MoE uses a soft dense mixture in which all global, local, and hybrid experts process the same question-conditioned spatial map. Their outputs are combined using one weight per expert for each example.

Bilinear Models for Vision–Language Fusion Bilinear models capture multiplicative interactions between two factors while remaining linear in either factor when the other is fixed. Early



(a) Inputs and frozen encoders **(b)** Token–patch interaction and IP-MoE **(c)** Multimodal prefix and LLM

Figure 1: Overview of IP-MoE with early aggregation. (a) A frozen CLIP backbone encodes the image as 24 × 24 = 576 patch tokens and the question as L text tokens, followed by two trainable projections. (b) Token–patch interaction gives one map per question token; patch-aware weighting aggregates them into a single map M that the router and all six experts share. (c) Three projectors form a 1,728-token prefix inserted at the multimodal position in the prompt.

work used them to separate style and content (Tenenbaum & Freeman, 2000). Bilinear CNNs pool outer products of local features for recognition, while Compact Bilinear Pooling provides lower-dimensional approximations (Lin et al., 2015; Gao et al., 2016). For vision–language fusion, MCB approximates visual–textual outer products for question-conditioned attention and answer prediction (Fukui et al., 2016). MLB combines learned projections through elementwise multiplication, while MFB uses factorized bilinear pooling with question and image attention (Kim et al., 2017; Yu et al., 2017). MUTAN and BLOCK parameterize bilinear interactions through Tucker and block-term decompositions, respectively (Ben-younes et al., 2017; 2019). BAN models attention over question-word and visual-region pairs using low-rank bilinear interactions (Kim et al., 2018). Building on projected Hadamard-product fusion, IP-MoE constructs token–patch interactions and aggregates them over question tokens separately at each image patch. The resulting spatial map supports expert processing, expert weighting, and language-model conditioning through a dedicated prefix block.

3 IP-MoE: Interaction Patchified Vision–Language Mixture of Experts

3.1 Framework Overview

As illustrated in Figure 1, IP-MoE is a soft dense MoE method that connects frozen CLIP image and text encoders to a causal language model (Radford et al., 2021). Separate trainable projection MLPs map the question-token and image-patch features to 512-dimensional representations. The interaction encoder combines every valid question token with every image patch, then aggregates question-token information separately at each patch. This produces a question-conditioned spatial map with one feature vector per image patch, preserving the image layout. Six experts process this map: two global experts use multi-head self-attention, two local experts use depthwise 3×3 convolution, and two hybrid experts apply self-attention followed by depthwise convolution within each block. Here, *dense* means that all six experts process every example, and *soft* means that their outputs are combined using normalized continuous weights. Image features, interaction features, and weighted expert outputs are separately projected into the language model's embedding space and concatenated as three prefix blocks. During training, we optimize the connector components and LoRA adapters while keeping the CLIP encoders and pretrained language-model weights frozen (Hu et al., 2022).

3.2 TOKEN–PATCH INTERACTION MAP

Let $t_i \in \mathbb{R}^d$ denote the projected feature of question token i, and $v_j \in \mathbb{R}^d$ the projected feature of image patch j, where d is their feature dimension. The question has L token positions, and the image has $N = G^2$ patches arranged on a $G \times G$ grid. We use $d = 512$, $G = 24$, interaction rank $r = 64$, and map width $C = 128$. Appendix F specifies token masking and the independent encoding of text chunks.

Pairwise interaction. We first construct a raw interaction vector, p_{ij}, for question token i and image patch j. We apply layer normalization (LN) to each feature, then use learned matrices, $W_T, W_V \in \mathbb{R}^{r \times d}$, to project the normalized features from d to r dimensions. We combine the projected features through elementwise multiplication ($\odot$):

$$p_{ij} = \frac{(W_T \,\mathrm{LN}(t_i)) \odot (W_V \,\mathrm{LN}(v_j))}{\sqrt{r}}. \tag{1}$$

We then normalize p_{ij} and apply the GELU activation. Because layer normalization rescales p_{ij}, we append a scalar q_{ij} that summarizes its magnitude before normalization. The exact definition of q_{ij} is provided in Appendix F. We define this augmented vector h_{ij}:

$$h_{ij} = [\mathrm{GELU}(\mathrm{LN}(p_{ij})); q_{ij}] \in \mathbb{R}^{r+1}. \tag{2}$$

A shared MLP, denoted by ϕ, transforms h_{ij} into the final token–patch feature f_{ij}:

$$f_{ij} = \phi(h_{ij}) \in \mathbb{R}^C, \tag{3}$$

where C is the number of feature channels in the resulting interaction map.

Patchwise token aggregation. Different image patches may require different information from the question. We therefore learn token weights separately for each patch, allowing each patch to emphasize different question tokens when forming its question-conditioned feature. We compute these features in three steps. First, a shared learned scoring vector, $w_s \in \mathbb{R}^{r+1}$, assigns a scalar score, s_{ij}, to each token–patch pair:

$$s_{ij} = w_s^\top h_{ij}. \tag{4}$$

Second, we convert these scores into token weights using a softmax. Let $\mathcal{I}$ denote the valid question-token positions. The weight, α_{ij}, specifies how much token i contributes to patch j:

$$\alpha_{ij} = \frac{\exp(s_{ij})}{\sum_{k \in \mathcal{I}} \exp(s_{kj})}, \qquad i \in \mathcal{I}. \tag{5}$$

These weights sum to one separately for each patch. Third, we compute the question-conditioned patch feature, m_j, as a weighted sum of the token–patch features:

$$m_j = \sum_{i \in \mathcal{I}} \alpha_{ij} f_{ij} \in \mathbb{R}^C. \tag{6}$$

Finally, we arrange the patch features, $m_1, \ldots, m_N$, according to their original image positions. This produces the spatial interaction map $M \in \mathbb{R}^{C \times G \times G}$, with C feature channels at each grid position. Question-token aggregation therefore occurs before routing and expert processing while preserving every image-patch position.

The sum-based variant replaces elementwise multiplication in Equation 1 with addition. The scaling and subsequent operations remain unchanged, and feature magnitudes are not explicitly matched between variants.

3.3 INTERACTION-CONDITIONED ROUTING

As shown in Figure 2, the router embeds the interaction map M as N spatial tokens with positional encodings, then applies self-attention across the image grid. Let $R_j \in \mathbb{R}^d$ denote the resulting feature at patch position j. The router normalizes these features, averages them across positions,

and applies a learned MLP, denoted by ϕ_{r}. This produces a score vector, $\ell \in \mathbb{R}^E$, where ℓ_e is the score for expert e and $E = 6$ is the number of experts:

$$\ell = \phi_{\mathrm{r}}\left(\frac{1}{N}\sum_{j=1}^{N}\mathrm{LN}(R_j)\right). \tag{7}$$

A softmax converts these scores into expert weights. Let g_e denote the weight assigned to expert e, and let $\tau > 0$ control how concentrated the weights are. During training, independent Gaussian noise η_e is added to each score; at inference, $\eta_e = 0$:

$$g_e = \frac{\exp((\ell_e + \eta_e)/\tau)}{\sum_{k=1}^{E}\exp((\ell_k + \eta_k)/\tau)}. \tag{8}$$

The weights sum to one. Each example receives one expert-weight vector, which is shared across all patch positions. Thus, g_e weights expert outputs, whereas α_{ij} weights question tokens separately for each patch.

3.4 Spatial Experts and Feature Integration

As shown in Figure 3, the expert pool contains two independently parameterized experts of each type: global, local, and hybrid. Each expert embeds the same interaction map M into spatial tokens and returns one feature vector per image patch. We denote the output of expert e by $E_e(M) \in \mathbb{R}^{N\times d}$. All attention, convolution, and channel-MLP sublayers use pre-normalization and residual connections, and each expert ends with its own layer normalization. Detailed block configurations are provided in Appendix F.

Global experts. Global experts use multi-head self-attention to exchange information across all patch positions. Each attention operation is followed by a channel MLP.

Local experts. Local experts arrange the spatial tokens on the image grid and apply depthwise 3×3 convolution to exchange information between neighboring patches independently within each channel. Each convolution operation is followed by a channel MLP.

Hybrid experts. Each hybrid block applies a global block followed by a local block. The global block exchanges information across all patch positions, while the local block processes neighboring patches.

Expert combination and language-model prefix. Before combining the expert outputs, we apply the same additional layer normalization to each output. Let $z_j \in \mathbb{R}^d$ denote the combined expert feature at patch position j, and let $Z \in \mathbb{R}^{N\times d}$ collect these features:

$$z_j = \sum_{e=1}^{E} g_e\,\mathrm{LN}(E_e(M)_j), \qquad Z = [z_1, \ldots, z_N]^\top. \tag{9}$$

The language model receives three feature sequences. Let $V = [v_1, \ldots, v_N]^\top \in \mathbb{R}^{N\times d}$ collect the image-patch features, and let $\bar{M} = [m_1, \ldots, m_N]^\top \in \mathbb{R}^{N\times C}$ collect the interaction-map features. The third sequence is the combined expert output Z. Separate learned projectors Π_V, Π_M, and Π_Z map these sequences to the language-model embedding dimension d_{LM}. We concatenate them along the token dimension to form the prefix P:

$$P = \mathrm{Concat}_{\mathrm{seq}}\big(\Pi_V(V), \Pi_M(\bar{M}), \Pi_Z(Z)\big) \in \mathbb{R}^{3N\times d_{\mathrm{LM}}}. \tag{10}$$

Each projector preserves all N patch positions, giving $3N$ prefix tokens. These tokens precede the textual question in the language model's user message.

Table 1: Scope of the main configuration comparisons. B-L, B-U, and S-L denote bilinear/learned, bilinear/uniform, and sum-based/learned configurations. Each configuration is trained with seed 42.

Comparison	Changes	Interpretation Limit
B-L vs. S-L	Elementwise multiplication is replaced with addition.	Changes inputs to the router, experts, and interaction prefix together; feature magnitudes are not matched.
B-L vs. B-U	Learned weights are replaced with equal weights; router training and its auxiliary loss are removed.	Compares separately trained models, so differences cannot be attributed to expert weights alone.

Table 2: Configuration results (%). Each model is trained once with seed 42.

Model	NaturalBench	CV-Bench-2D	CV-Bench-3D	MMBench	MMStar	POPE	SEEDBench
B-L	67.61	57.61	51.33	62.71	38.20	84.86	65.06
B-U	67.20	55.89	54.17	62.46	39.00	83.21	65.09
S-L	67.09	54.42	54.58	63.06	38.47	85.18	64.41

4 Experiments

Experiment Design We compare bilinear interactions with learned weighting (B-L), bilinear interactions with uniform weighting (B-U), and sum-based interactions with learned weighting (S-L). All configurations use frozen CLIP image and text encoders, Llama-3.2-3B-Instruct with LoRA, six spatial experts, and 1,728 prefix tokens (Radford et al., 2021; Meta, 2024; Hu et al., 2022). The image encoder is CLIP ViT-L/14 with 336×336-pixel inputs. Each configuration is trained once with seed 42. Table 1 specifies the changes and interpretation limits of the two main comparisons.

Data and Evaluation We train IP-MoE on LLaVA-NeXT-Data (Liu et al., 2024a), with a recorded split of 450,330 training and 4,549 validation examples. We compare the fitted configurations on seven benchmarks using VLMEvalKit (Duan et al., 2026). We report individual-answer accuracy for NaturalBench; average accuracy across ADE20K and COCO for CV-Bench-2D and overall accuracy for CV-Bench-3D; F1 for POPE; and overall accuracy for SEEDBench-IMG and MMStar (Li et al., 2024a; Tong et al., 2024; Li et al., 2023b; 2024b; Chen et al., 2024). For the MMBench v1.1 English development split, we report accuracy under circular evaluation, which tests consistency across answer-option permutations (Liu et al., 2024b). All scores are percentages, and differences are reported in percentage points. IP-MoE uses greedy decoding with a maximum of 32 generated tokens per answer.

Training We first train the connector for one epoch with a learning rate of 3×10^{-4}. We then jointly train the connector and LoRA adapters for two epochs with a learning rate of 10^{-4}. Both stages use a nominal effective batch size of 360. Appendix F provides the objective, regularization coefficients, and implementation settings.

4.1 Quantitative Analysis

Table 2 compares the three configurations. Relative to B-U, B-L scores 1.72 percentage points higher on CV-Bench-2D and 1.65 points higher on POPE F1, but 2.83 points lower on CV-Bench-3D and 0.80 points lower on MMStar. Relative to S-L, B-L scores 3.19 points higher on CV-Bench-2D and 3.25 points lower on CV-Bench-3D. The configurations have no consistent ordering across the seven benchmarks.

Table 3 reports paired 95% bootstrap intervals for the four benchmarks whose resampling groups can be reconstructed. All four B-L versus B-U intervals include zero. For B-L versus S-L, the CV-Bench-2D interval is positive and the CV-Bench-3D interval is negative; the NaturalBench and MMBench intervals include zero. These unadjusted intervals quantify evaluation-sample uncertainty conditional on the fitted checkpoints and saved answer-extraction decisions. They do not

Table 3: Score differences in percentage points, with paired 95% bootstrap intervals in brackets (20,000 resamples; seed 42). Positive values favor B-L. All four benchmarks use accuracy, with benchmark-specific grouping and weighting. Intervals are conditional on the saved checkpoints and answer-extraction decisions, and are not adjusted for multiple comparisons. Intervals for POPE, SEEDBench, and MMStar are omitted because source-image grouping information is unavailable.

Benchmark	B-L − B-U	B-L − S-L
NaturalBench	$+0.41$ $[-0.30, +1.13]$	$+0.51$ $[-0.18, +1.21]$
CV-Bench-2D	$+1.72$ $[-0.05, +3.52]$	$+3.19$ $[+1.17, +5.25]$
CV-Bench-3D	-2.83 $[-6.16, +0.52]$	-3.25 $[-6.09, -0.42]$
MMBench	$+0.26$ $[-1.46, +1.98]$	-0.34 $[-1.98, +1.29]$

Table 4: Contextual reference-system results (%). NaturalBench reports individual-answer accuracy, POPE reports F1, and the remaining benchmarks report accuracy. The IP-MoE row reports the bilinear/learned configuration (B-L). Reference models differ in backbones, training data, prompts, and evaluation implementations; these comparisons do not isolate connector effects. Values are rounded to two decimals; reference systems use their native evaluation wrappers.

Model	NaturalBench	CV-Bench-2D	CV-Bench-3D	MMBench	MMStar	POPE	SEEDBench
BLIP-2	60.30	49.47	49.75	65.21	33.07	74.16	57.87
InstructBLIP	65.95	48.75	52.67	66.75	35.33	81.54	63.69
Mono-InternVL	67.87	54.30	47.58	63.40	46.07	85.00	66.51
SpatialLadder	69.03	66.57	69.58	73.97	48.47	82.15	69.03
Spatial-SSRL	75.08	69.03	72.33	80.33	56.73	87.22	74.37
IP-MoE (B-L)	67.61	57.61	51.33	62.71	38.20	84.86	65.06

measure training-seed variation, and intervals containing zero do not establish equivalence. POPE, SEEDBench-IMG, and MMStar retain point estimates because source-image grouping is unavailable in the saved exports. Appendix C specifies the procedure. The CV-Bench-3D differences also accompany substantial answer-option imbalance, examined in Section 4.2.

The CV-Bench-2D breakdown reveals opposing subtask differences. Relative to S-L, B-L has higher counting accuracy (57.11% vs. 49.24%) but lower spatial-relation accuracy (59.85% vs. 62.15%; Table 20). These directions hold within both ADE20K and COCO. The primary metric averages the two source accuracies equally, whereas the pooled task scores combine individual examples across sources. B-L's higher overall CV-Bench-2D score therefore does not indicate improvement across all evaluated spatial tasks.

The POPE comparison likewise depends on the metric. Relative to B-U, B-L has higher recall (77.93% vs. 74.33%) and lower precision (93.15% vs. 94.49%; Table 22). Its higher F1 reflects this precision–recall trade-off and does not establish fewer false-positive object-presence predictions.

Table 4 provides contextual results for BLIP-2, InstructBLIP, Mono-InternVL, SpatialLadder, and Spatial-SSRL (Li et al., 2023a; Dai et al., 2023; Luo et al., 2025; Li et al., 2026; Liu et al., 2026). Spatial-SSRL scores higher than B-L on all seven primary metrics; comparisons with the other systems vary. Differences in backbones, training data, prompts, and evaluation implementations prevent attributing these rankings to the connector. NaturalBench also distinguishes individual and grouped performance: B-L has 67.61% individual-answer accuracy but 14.95% group accuracy when all four related answers must be correct. Appendix A reports the benchmark breakdowns.

Under the fixed-input, one-token inference protocol in Appendix G, B-L records 11.865 counted TFLOPs and 181.75 ms latency. These measurements describe the complete system under its native wrapper. No matched simpler connector or prefix-length control establishes whether the additional expert computation and three prefix blocks provide a favorable accuracy–cost trade-off.

Table 5: Clean-gate stage endpoints from the final logged training microbatch. Entropy H is in nats (maximum $\ln 6 \simeq 1.7918$). The statistic s_g averages each expert's standard deviation across examples; it does not measure deviation from $1/6$.

	Stage 1		Stage 2	
Configuration	H	s_g	H	s_g
B-L	1.2182	0.0122	1.7838	0.0113
S-L	1.5514	0.0306	1.7897	0.0094

Table 6: CV-Bench-3D option selections from existing outputs (1,200 questions). References contain 603 A and 597 B answers; uniform random choice has 50% expected accuracy, while always answering A scores 50.25%. Option imbalance is descriptive and does not identify its cause.

Model	A count	B count	A rate (%)	Accuracy (%)
B-L	1,163	37	96.92	51.33
B-U	935	265	77.92	54.17
S-L	946	254	78.83	54.58

4.2 Routing and Prediction Analysis

The training records distinguish weight concentration from variation across examples. At the final recorded stage-1 microbatch, B-L has nonuniform clean weights with limited across-example variation. At the stage-2 endpoint, the mean clean-gate entropies of B-L and S-L reach 99.56% and 99.89% of the maximum, respectively (Table 5). These measurements remove explicit routing noise but retain training-mode dropout. They describe sampled, length-grouped training microbatches rather than held-out routing behavior. Appendix H defines the statistics and reports their trajectories.

CV-Bench-3D predictions exhibit substantial answer-option imbalance. B-L selects A on 1,163 of 1,200 questions (96.92%), compared with 77.92% for B-U and 78.83% for S-L (Table 6). On distance questions, B-L selects A on 596 of 600 examples. Its overall accuracy of 51.33% is close to the 50.25% obtained by always answering A. These counts characterize responses under the recorded prompts and option order; they do not identify the cause of the concentration or isolate a difference in geometric reasoning.

Expert-family results and correctness transitions are reported in Appendices B and D. Differences in evaluation-code versions limit attribution of the observed score changes to family restriction alone.

5 Discussion

Aggregate differences depend on the task and comparator. Relative to S-L, B-L's higher CV-Bench-2D score combines higher counting accuracy with lower spatial-relation accuracy. Relative to B-U, its higher POPE F1 combines higher recall with lower precision. These decompositions make the observed differences more specific than the benchmark labels alone: they do not establish uniform improvements in spatial understanding or object-presence judgments. B-L's higher spatial-relation score than S-L on SEEDBench, opposite to their CV-Bench-2D ordering, also limits a general conclusion about relational reasoning.

The shared gate trajectory does not identify a bilinear-specific failure. Both interaction operators approach nearly uniform clean weights late in training. This shared observation cannot, by itself, establish a failure specific to multiplication. At the stage-1 endpoint, B-L instead combines nonuniform weights with limited across-example variation in the sampled microbatch, illustrating that concentration and input variation are distinct properties. During stage 2, validation answer loss decreases while clean-gate entropy approaches its maximum. These records do not distinguish the contributions of expert representations, direct prefixes, or regularization to the observed trajectory.

Nearly uniform final gates do not imply equivalent training. B-L trains with learned, noisy weights and router regularization, whereas B-U fixes equal weights and bypasses the router objective. B-L also has nonuniform weights at its recorded stage-1 endpoint. Similar final clean weights therefore do not imply equivalent fitted experts, connector features, or language-model adapters. The balance objective encourages uniform mean usage across a microbatch while permitting selective weights for individual examples. Neither the objective nor the observed entropy establishes why near-uniformity emerged.

Weight entropy does not measure predictive influence. The answer-loss gradient identity in Appendix I depends on downstream sensitivity and its alignment with each expert's deviation from the mixture, summed across patches. Entropy alone determines none of these quantities. The direct image and interaction prefixes offer other conditioning paths, but their presence does not establish that the decoder bypasses the experts. Likewise, global weighting cannot assign separate mixtures to separate patches, but it can still depend on the question-conditioned map. These architectural properties identify possible restrictions without demonstrating ineffective routing or absent expert specialization.

The CV-Bench-3D deficit is accompanied by answer-option imbalance. B-L's accuracy is close to the always-A reference and its responses are strongly concentrated on A. Answer-position preference and prompt sensitivity remain alternatives to an explanation based on impaired 3D representations. The observed 2D and 3D score differences consequently do not establish that bilinear interactions improve one form of spatial reasoning at the expense of another.

Limitations. Each configuration is trained once. The evaluation intervals do not measure stability across training runs, and the comparisons change complete computational or optimization configurations. A matched simpler connector, a sum-based uniform configuration, and controlled gate and prefix interventions are absent. The additional computation of dense experts and the three prefix blocks therefore has no established benefit over a simpler matched design. Training–benchmark overlap has not been audited. The analysis archive lacks evaluated weights and complete matching historical evaluation source, limiting independent regeneration of predictions. Source-version differences also restrict the interpretation of the family runs. The findings apply to the recorded configurations; the separate contributions of adaptive weighting, interaction representations, and expert computation remain unresolved.

6 Conclusion

We studied three IP-MoE configurations across seven benchmarks. Their mixed rankings conceal opposing subtask changes. Both learned configurations approach nearly uniform weights in recorded late-training microbatches, and CV-Bench-3D responses exhibit substantial answer-option imbalance. These findings characterize the evaluated systems; establishing the incremental value of interaction-conditioned routing and expert computation requires matched controls and repeated training.

AI USE STATEMENT

AI was used to review the manuscript against ICLR guidance, give methodological and experimental feedback, interpret supplied results and training diagnostics, implement and check paired bootstrap and prediction-level analyses of saved outputs, regenerate plots from recorded training metrics, revise the manuscript, format tables, inspect supplied code, derive and check an architectural gradient analysis, check selected references, and construct and correct technical schematics. Generative image tools were used to produce illustrative schematics during manuscript development, including imagery incorporated into a diagram in the submitted paper. Responsibility for the final content and verification of AI-assisted work remains with the authors.

ETHICS STATEMENT

This study adapts pretrained vision–language components using an existing data collection and evaluates existing benchmarks. Generated answers can be incorrect, including object-presence judgments, and the evaluations do not establish suitability for safety-critical use. Bias, privacy, and data-licensing questions inherited from the pretrained components and training collection are not resolved by these benchmark results. The study does not measure these risks or establish an absence of training–benchmark overlap.

REPRODUCIBILITY STATEMENT

Section 3 describes the architecture, and Appendix F provides implementation and evaluation details. The supplementary code includes the IP-MoE implementation, data preprocessing utilities, training configurations and scripts, and evaluation scripts with a bundled VLMEvalKit integration. Trained checkpoints, historical logs, saved predictions, and post-hoc analysis scripts are not included.

REFERENCES

Hangbo Bao, Wenhui Wang, Li Dong, Qiang Liu, Owais Khan Mohammed, Kriti Aggarwal, Subhojit Som, Songhao Piao, and Furu Wei. VLMo: Unified vision-language pre-training with mixture-of-modality-experts. In *Advances in Neural Information Processing Systems*, volume 35, pp. 32897–32912, 2022. doi: 10.52202/068431-2384.

Hedi Ben-younes, Remi Cadene, Matthieu Cord, and Nicolas Thome. MUTAN: Multimodal Tucker fusion for visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2631–2639, 2017. doi: 10.1109/ICCV.2017.285.

Hedi Ben-younes, Remi Cadene, Nicolas Thome, and Matthieu Cord. BLOCK: Bilinear superdiagonal fusion for visual question answering and visual relationship detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 8102–8109, 2019. doi: 10.1609/aaai.v33i01.33018102.

Weilin Cai, Juyong Jiang, Fan Wang, Jing Tang, Sunghun Kim, and Jiayi Huang. A survey on mixture of experts in large language models. *IEEE Transactions on Knowledge and Data Engineering*, 37(7):3896–3915, 2025. doi: 10.1109/TKDE.2025.3554028.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. Are we on the right way for evaluating large vision-language models? In *Advances in Neural Information Processing Systems*, volume 37, pp. 27056–27087, 2024. doi: 10.52202/079017-0850.

Zewen Chi, Li Dong, Shaohan Huang, Damai Dai, Shuming Ma, Barun Patra, Saksham Singhal, Payal Bajaj, Xia Song, Xian-Ling Mao, Heyan Huang, and Furu Wei. On the representation collapse of sparse mixture of experts. In *Advances in Neural Information Processing Systems*, volume 35, pp. 34600–34613, 2022. doi: 10.52202/068431-2507.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. InstructBLIP: Towards general-purpose vision-language models with instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 49250–49267, 2023. doi: 10.52202/075280-2142.

Matt Deitke, Christopher Clark, Sangho Lee, et al. Molmo and PixMo: Open weights and open data for state-of-the-art vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 91–104, 2025. doi: 10.1109/CVPR52734.2025.00018.

Haodong Duan, Xinyu Fang, Junming Yang, Xiangyu Zhao, Zerun Ma, Yuxuan Qiao, et al. VLMEvalKit: An open-source toolkit for evaluating large multi-modality models. *arXiv preprint arXiv:2407.11691v5*, 2026.

Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 457–468. Association for Computational Linguistics, 2016. doi: 10.18653/v1/D16-1044.

Yang Gao, Oscar Beijbom, Ning Zhang, and Trevor Darrell. Compact bilinear pooling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 317–326, 2016. doi: 10.1109/CVPR.2016.41.

Yongxin Guo, Zhenglin Cheng, Xiaoying Tang, Zhaopeng Tu, and Tao Lin. Dynamic mixture of experts: An auto-tuning approach for efficient Transformer models. In *International Conference on Learning Representations*, 2025.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Samy Jelassi, Clara Mohri, David Brandfonbrener, Alex Gu, Nikhil Vyas, Nikhil Anand, David Alvarez-Melis, Yuanzhi Li, Sham M. Kakade, and Eran Malach. Mixture of parrots: Experts improve memorization more than reasoning. In *International Conference on Learning Representations*, 2025.

Jin-Hwa Kim, Kyoung-Woon On, Woosang Lim, Jeonghee Kim, Jung-Woo Ha, and Byoung-Tak Zhang. Hadamard product for low-rank bilinear pooling. In *International Conference on Learning Representations*, 2017.

Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. Bilinear attention networks. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

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. In *Advances in Neural Information Processing Systems*, volume 37, pp. 17044–17068, 2024a. doi: 10.52202/079017-0542.

Bohao Li, Yuying Ge, Yixiao Ge, Guangzhi Wang, Rui Wang, Ruimao Zhang, and Ying Shan. SEED-Bench: Benchmarking multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13299–13308, 2024b. doi: 10.1109/CVPR52733.2024.01263.

Hongxing Li, Dingming Li, Zixuan Wang, Yuchen Yan, Hang Wu, Wenqi Zhang, Yongliang Shen, Weiming Lu, Jun Xiao, and Yueting Zhuang. SpatialLadder: Progressive training for spatial reasoning in vision-language models. In *International Conference on Learning Representations*, 2026.

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 *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19730–19742. PMLR, 2023a.

Pingzhi Li, Zhenyu Zhang, Prateek Yadav, Yi-Lin Sung, Yu Cheng, Mohit Bansal, and Tianlong Chen. Merge, then compress: Demystify efficient SMoE with hints from its routing policy. In *International Conference on Learning Representations*, 2024c.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, 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*, pp. 292–305. Association for Computational Linguistics, 2023b. doi: 10.18653/v1/2023.emnlp-main.20.

Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear CNN models for fine-grained visual recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1449–1457, 2015. doi: 10.1109/ICCV.2015.170.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. LLaVA-NeXT: Improved reasoning, OCR, and world knowledge. LLaVA blog, January 2024a. URL https://llava-vl.github.io/blog/2024-01-30-llava-next/.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. MMBench: Is your multi-modal model an all-around player? In *Computer Vision – ECCV 2024*, pp. 216–233. Springer, 2024b. doi: 10.1007/978-3-031-72658-3_13.

Yuhong Liu, Beichen Zhang, Yuhang Zang, Yuhang Cao, Long Xing, Xiaoyi Dong, Haodong Duan, Dahua Lin, and Jiaqi Wang. Spatial-SSRL: Enhancing spatial understanding via self-supervised reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Gen Luo, Xue Yang, Wenhan Dou, Zhaokai Wang, Jiawen Liu, Jifeng Dai, Yu Qiao, and Xizhou Zhu. Mono-InternVL: Pushing the boundaries of monolithic multimodal large language models with endogenous visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24960–24971, 2025. doi: 10.1109/CVPR52734.2025.02324.

Meta. Llama-3.2-3B-Instruct, September 2024. URL https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct. Model card.

Marianna Nezhurina, Tomer Porian, Giovanni Puccetti, Tommie Kerssies, Romain Beaumont, Mehdi Cherti, and Jenia Jitsev. Scaling laws for robust comparison of open foundation language-vision models and datasets. In *Advances in Neural Information Processing Systems*, volume 38, pp. 170907–170946, 2025. doi: 10.52202/085713-5148.

Stefan K. Nielsen, Rachel S. Y. Teo, Laziz U. Abdullaev, and Tan M. Nguyen. Tight clusters make specialized experts. In *International Conference on Learning Representations*, 2025.

Joan Puigcerver, Carlos Riquelme, Basil Mustafa, and Neil Houlsby. From sparse to soft mixtures of experts. In *International Conference on Learning Representations*, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021.

Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling vision with sparse mixture of experts. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

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.

Leyang Shen, Gongwei Chen, Rui Shao, Weili Guan, and Liqiang Nie. MoME: Mixture of multimodal experts for generalist multimodal large language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 42048–42070, 2024. doi: 10.52202/079017-1330.

Joshua B. Tenenbaum and William T. Freeman. Separating style and content with bilinear models. *Neural Computation*, 12(6):1247–1283, 2000. doi: 10.1162/089976600300015349.

Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, Austin Wang, Rob Fergus, Yann LeCun, and Saining Xie. Cambrian-1: A fully open, vision-centric exploration of multimodal LLMs. In *Advances in Neural Information Processing Systems*, volume 37, pp. 87310–87356, 2024. doi: 10.52202/079017-2771.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Zhou Yu, Jun Yu, Jianping Fan, and Dacheng Tao. Multi-modal factorized bilinear pooling with co-attention learning for visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1839–1848, 2017. doi: 10.1109/ICCV.2017.202.

Xiaofeng Zhang, Yikang Shen, Zeyu Huang, Jie Zhou, Wenge Rong, and Zhang Xiong. Mixture of attention heads: Selecting attention heads per token. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 4150–4162. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.emnlp-main.278.

Zhuofan Zong, Bingqi Ma, Dazhong Shen, Guanglu Song, Hao Shao, Dongzhi Jiang, Hongsheng Li, and Yu Liu. MoVA: Adapting mixture of vision experts to multimodal context. In *Advances in Neural Information Processing Systems*, volume 37, pp. 103305–103333, 2024. doi: 10.52202/079017-3282.

A Detailed Benchmark Results

The following tables report the source, task, and category breakdowns for the models in Table 4. Rows and benchmark categories follow the corresponding performance tables. All scores are percentages; benchmark scores are displayed to two decimal places. NaturalBench's Acc is individual-answer accuracy, and G Acc requires all four answers in a group to be correct. POPE's Overall denotes F1. CV-Bench-2D averages the ADE20K and COCO accuracies. MMBench uses circular evaluation on the English v1.1 development split. Category labels follow the evaluator; CP denotes coarse perception in the MMBench comparison table. Higher values are better, and no cross-benchmark average is reported. Wide category tables are presented in consecutive panels.

Table 7: CV-Bench-2D source accuracies (%). Overall is the arithmetic mean of the ADE20K and COCO accuracies.

Model	ADE20K	COCO	Overall
blip2	46.76	52.17	49.47
instructblip	45.81	51.68	48.75
mono internvl	46.60	61.99	54.30
spatial ladder	61.45	71.68	66.57
spatial ssrl	61.30	76.77	69.03
IP-MoE (B-L)	51.50	63.73	57.61

Table 8: CV-Bench-3D depth, distance, and overall accuracies (%).

Model	Depth	Distance	Overall
blip2	51.17	48.33	49.75
instructblip	53.67	51.67	52.67
mono internvl	67.33	27.83	47.58
spatial ladder	73.33	65.83	69.58
spatial ssrl	81.83	62.83	72.33
IP-MoE (B-L)	52.50	50.17	51.33

Table 9: POPE results (%). Acc, Precision, and Recall are reported separately; Overall denotes F1 and is the primary score.

Model	Acc	Precision	Recall	Overall
blip2	78.53	93.15	61.60	74.16
instructblip	83.46	92.23	73.07	81.54
mono internvl	86.01	91.63	79.27	85.00
spatial ladder	83.91	92.22	74.07	82.15
spatial ssrl	88.12	94.39	81.07	87.22
IP-MoE (B-L)	86.10	93.15	77.93	84.86

Table 10: SEEDBench category and overall accuracies (%). Panels preserve the category order.

(a) Instance categories

Model	Instance Attributes	Instance Identity	Instance Interaction	Instance Location	Instances Counting
blip2	55.26	63.74	64.95	51.12	42.95
instructblip	66.44	67.29	68.04	53.89	50.84
mono internvl	69.71	69.31	67.01	60.53	56.80
spatial ladder	69.00	73.18	69.07	66.36	60.48
spatial ssrl	76.86	78.54	75.26	68.71	66.73
IP-MoE (B-L)	67.18	69.09	65.98	57.16	54.35

(b) Remaining categories and Overall

Model	Scene Understanding	Spatial Relation	Text Understanding	Visual Reasoning	Overall
blip2	72.29	44.90	61.90	77.34	57.87
instructblip	72.70	45.66	58.33	79.15	63.69
mono internvl	73.62	46.88	55.95	69.18	66.51
spatial ladder	75.78	58.14	65.48	75.83	69.03
spatial ssrl	78.34	61.64	71.43	77.34	74.37
IP-MoE (B-L)	74.22	42.77	45.24	77.34	65.06

Table 11: MMBench results (%), panel 1 of 4: AR through RR. Category labels and their order are retained.

Model	AR	CP	FP-C	FP-S	LR	RR
blip2	70.35	74.32	55.94	63.14	50.00	65.22
instructblip	71.36	76.69	51.05	67.24	51.69	66.96
mono internvl	71.36	79.39	48.25	63.14	39.83	52.17
spatial ladder	78.39	83.11	62.24	79.52	50.85	66.96
spatial ssrl	82.91	88.51	67.83	82.25	66.10	80.00
IP-MoE (B-L)	68.84	75.68	50.35	59.04	43.22	63.48

Table 12: MMBench results (%), panel 2 of 4: Action Recognition through Identity Reasoning. Category labels and their order are retained.

Model	Action Recognition	Attribute Comparison	Attribute Recognition	Celebrity Recognition	Function Reasoning	Future Prediction	Identity Reasoning
blip2	87.04	54.55	71.62	82.83	88.61	57.50	95.56
instructblip	88.89	38.64	72.97	83.84	88.61	65.00	97.78
mono internvl	85.19	29.55	81.08	54.55	68.35	27.50	88.89
spatial ladder	90.74	47.73	90.54	93.94	88.61	27.50	97.78
spatial ssrl	88.89	77.27	93.24	93.94	88.61	55.00	100.00
IP-MoE (B-L)	87.04	27.27	75.68	60.61	78.48	52.50	93.33

Table 13: MMBench results (%), panel 3 of 4: Image Emotion through Object Localization. Category labels and their order are retained.

Model	Image Emotion	Image Quality	Image Scene	Image Style	Image Topic	Nature Relation	Object Localization
blip2	74.00	24.53	96.15	71.70	88.89	45.83	34.57
instructblip	80.00	28.30	95.19	77.36	88.89	56.25	40.74
mono internvl	80.00	64.15	91.35	69.81	80.56	22.92	55.56
spatial ladder	74.00	60.38	97.12	88.68	80.56	56.25	49.38
spatial ssrl	80.00	71.70	98.08	88.68	97.22	81.25	53.09
IP-MoE (B-L)	82.00	35.85	96.15	66.04	80.56	50.00	35.80

Table 14: MMBench results (%), panel 4 of 4: OCR through Overall. Category labels and their order are retained.

Model	OCR	Physical Property Reasoning	Physical Relation	Social Relation	Spatial Relationship	Structured ImageText Understanding	Overall
blip2	56.41	36.00	54.17	93.02	20.00	46.15	65.21
instructblip	69.23	37.33	45.83	90.70	17.78	44.87	66.75
mono internvl	66.67	64.00	70.83	74.42	22.22	46.15	63.40
spatial ladder	84.62	56.00	37.50	95.35	42.22	62.82	73.97
spatial ssrl	92.31	66.67	50.00	95.35	33.33	71.79	80.33
IP-MoE (B-L)	71.79	44.00	58.33	81.40	28.89	38.46	62.71

Table 15: MMStar category and overall accuracies (%).

Model	Coarse Perception	Fine-Grained Perception	Instance Reasoning	Logical Reasoning	Math	Science & Technology	Overall
blip2	53.60	33.20	43.20	27.20	22.00	19.20	33.07
instructblip	52.40	34.00	46.40	33.20	21.60	24.40	35.33
mono internvl	60.80	36.00	52.80	41.60	44.80	40.40	46.07
spatial ladder	60.80	44.40	55.20	44.00	53.60	32.80	48.47
spatial ssrl	72.80	52.00	61.20	56.40	59.60	38.40	56.73
IP-MoE (B-L)	57.60	27.20	51.20	36.80	34.80	21.60	38.20

Table 16: NaturalBench results (%). Q Acc and I Acc are the paired metrics; G Acc is group accuracy requiring all four answers in a group to be correct. Acc denotes individual-answer accuracy and is the primary score.

Model	Q-Acc	I-Acc	G-Acc	Acc
blip2	22.05	27.29	5.05	60.30
instructblip	34.76	41.45	11.26	65.95
mono internvl	39.84	44.89	14.74	67.87
spatial ladder	42.18	46.76	17.68	69.03
spatial ssrl	52.53	55.84	26.58	75.08
IP-MoE (B-L)	38.29	44.42	14.95	67.61

B Detailed Configuration Comparisons

This appendix reports category scores for the three trained configurations and the B-L expert-family restrictions. The latter retain only the global, local, or hybrid expert pair, renormalize its weights, and preserve both direct prefix blocks. Their source-version differences are documented in Appendix F. All training uses seed 42; these comparisons are descriptive. Scores are percentages, reported to two decimal places.

Table 17: Recorded expert-family evaluations (%). O and X indicate retained and excluded expert families. Restricted rows use the B-L checkpoint and renormalize retained weights; direct prefixes remain. Four recorded source files differ from the unrestricted evaluation, so these descriptive comparisons do not isolate family effects. Metrics follow Table 4.

(a) NaturalBench and CV-Bench

Model	Global	Local	Hybrid	NaturalBench	CV-Bench-2D	CV-Bench-3D
IP-MoE (all)	O	O	O	67.61	57.61	51.33
IP-MoE (Global only)	O	X	X	67.57	57.05	52.33
IP-MoE (Local only)	X	O	X	67.04	57.66	51.25
IP-MoE (Hybrid only)	X	X	O	67.14	56.83	51.42

(b) MMBench, MMStar, POPE, and SEEDBench

Model	Global	Local	Hybrid	MMBench	MMStar	POPE	SEEDBench
IP-MoE (all)	O	O	O	62.71	38.20	84.86	65.06
IP-MoE (Global only)	O	X	X	62.89	38.33	84.77	64.97
IP-MoE (Local only)	X	O	X	62.71	38.07	85.16	64.83
IP-MoE (Hybrid only)	X	X	O	61.94	38.73	84.07	64.90

B.1 NATURALBENCH

Table 18 reports individual, paired, and group metrics. Group accuracy requires all four related answers to be correct, providing a stricter measure of consistent image–question discrimination.

Table 18: NaturalBench results. The paired-metric labels Q_Acc and I_Acc are retained from the evaluator. Group accuracy requires all four answers in an image–question group to be correct.

Metric	B-L	B-U	S-L	Global	Local	Hybrid
Individual-answer accuracy	67.61	67.20	67.09	67.57	67.04	67.14
Q-Acc	38.29	37.79	37.26	38.32	36.95	37.18
I-Acc	44.42	44.11	42.50	44.55	43.18	43.18
Group accuracy	14.95	13.68	13.42	14.32	14.32	14.11

B.2 CV-BENCH-2D

Tables 19 and 20 separate source and task effects. B-L exceeds S-L on counting and scores lower on spatial relations within both sources. The primary metric averages source accuracies equally; pooled task scores use individual examples.

Table 19: CV-Bench-2D source accuracies. Overall is the arithmetic mean of the ADE20K and COCO accuracies, rather than accuracy pooled over all examples.

Metric	B-L	B-U	S-L	Global	Local	Hybrid
ADE20K	51.50	51.66	48.97	50.87	51.34	51.18
COCO	63.73	60.12	59.88	63.23	63.98	62.48
Overall (source average)	57.61	55.89	54.42	57.05	57.66	56.83

Table 20: CV-Bench-2D task accuracies computed from saved evaluator hits. Each row is correct answers divided by the number of examples n. Pooled rows combine both sources within a task and do not use the source-averaging rule of the primary metric. Relation denotes the evaluator's spatial-relation category.

Source / task	n	B-L	B-U	S-L	Global	Local	Hybrid
ADE20K: counting	342	45.32	46.78	38.30	43.27	45.03	46.20
ADE20K: relation	291	58.76	57.39	61.51	59.79	58.76	57.04
COCO: counting	446	66.14	62.33	57.62	63.45	65.92	65.70
COCO: relation	359	60.72	57.38	62.67	62.95	61.56	58.50
Pooled counting	788	57.11	55.58	49.24	54.70	56.85	57.23
Pooled relation	650	59.85	57.38	62.15	61.54	60.31	57.85

B.3 CV-BENCH-3D

B-L scores below B-U and S-L on both depth and distance (Table 21). The recorded global-family run has higher depth accuracy but slightly lower distance accuracy than B-L, subject to the source-version limitation above. Answer-option distributions are reported in Appendix D.

Table 21: CV-Bench-3D overall and task accuracies.

Metric	B-L	B-U	S-L	Global	Local	Hybrid
Depth	52.50	54.17	55.50	54.83	52.33	52.67
Distance	50.17	54.17	53.67	49.83	50.17	50.17
Overall	51.33	54.17	54.58	52.33	51.25	51.42

B.4 POPE

Table 22 separates F1, accuracy, precision, and recall. B-L's higher F1 than B-U accompanies higher recall and lower precision; it does not imply fewer false-positive object-presence predictions.

Table 22: POPE results. F1, accuracy, precision, and recall are distinct metrics; the overall F1 is the primary reported score.

Metric	B-L	B-U	S-L	Global	Local	Hybrid
Random F1	86.56	84.47	86.92	86.43	86.91	85.72
Popular F1	85.52	83.71	85.69	85.39	85.84	84.81
Adversarial F1	82.61	81.51	83.01	82.58	82.84	81.78
Overall F1	84.86	83.21	85.18	84.77	85.16	84.07
Overall accuracy	86.10	85.00	86.31	86.03	86.23	85.51
Overall precision	93.15	94.49	92.86	93.21	92.36	93.35
Overall recall	77.93	74.33	78.67	77.73	79.00	76.47

B.5 SEEDBENCH-IMG

Table 23 shows opposing category differences behind small overall changes. B-L's higher spatial-relation score than S-L contrasts with their CV-Bench-2D relation scores. Categories differ in size, so the overall score is not their unweighted mean.

Table 23: SEEDBench-IMG overall and category accuracies. Categories retain the evaluator's names.

Metric	B-L	B-U	S-L	Global	Local	Hybrid
Instance Attributes	67.18	67.46	66.75	66.92	66.90	66.83
Instance Identity	69.09	69.96	70.34	69.09	68.92	69.36
Instance Interaction	65.98	61.86	65.98	64.95	65.98	64.95
Instance Location	57.16	56.34	55.83	56.54	57.26	57.36
Instances Counting	54.35	53.58	52.64	54.39	53.94	53.82
Scene Understanding	74.22	74.54	73.88	74.16	74.00	74.10
Spatial Relation	42.77	43.38	40.18	43.38	43.07	43.68
Text Understanding	45.24	45.24	34.52	47.62	40.48	45.24
Visual Reasoning	77.34	74.62	76.13	77.64	77.64	76.74
Overall	65.06	65.09	64.41	64.97	64.83	64.90

B.6 MMBench

Table 24 reports coarse and fine categories under circular evaluation. The small overall differences combine gains and losses across categories.

Table 24: MMBench v1.1 English development results under circular evaluation. AR, CP, FP-C, FP-S, LR, and RR retain the evaluator's category labels; the remaining rows report its finer categories.

Metric	B-L	B-U	S-L	Global	Local	Hybrid
AR	68.84	67.84	70.35	71.36	69.35	68.84
CP	75.68	75.68	76.69	76.01	76.35	74.66
FP-C	50.35	47.55	48.25	50.35	50.35	50.35
FP-S	59.04	60.07	59.73	58.70	58.70	58.36
LR	43.22	43.22	37.29	42.37	42.37	40.68
RR	63.48	63.48	68.70	61.74	62.61	62.61
Action recognition	87.04	87.04	88.89	87.04	87.04	87.04
Attribute comparison	27.27	20.45	22.73	27.27	29.55	27.27
Attribute recognition	75.68	75.68	79.73	74.32	75.68	74.32
Celebrity recognition	60.61	67.68	61.62	59.60	60.61	60.61
Function reasoning	78.48	79.75	78.48	78.48	78.48	75.95
Future prediction	52.50	55.00	52.50	50.00	52.50	50.00
Identity reasoning	93.33	91.11	93.33	93.33	93.33	93.33
Image emotion	82.00	82.00	78.00	82.00	82.00	82.00
Image quality	35.85	35.85	33.96	37.74	39.62	35.85
Image scene	96.15	95.19	95.19	96.15	96.15	94.23
Image style	66.04	66.04	75.47	64.15	66.04	64.15
Image topic	80.56	83.33	86.11	83.33	80.56	80.56
Nature relation	50.00	52.08	54.17	50.00	50.00	50.00
Object localization	35.80	33.33	33.33	37.04	35.80	34.57
OCR	71.79	66.67	71.79	71.79	69.23	71.79
Physical property reasoning	44.00	41.33	48.00	50.67	45.33	46.67
Physical relation	58.33	54.17	58.33	45.83	54.17	58.33
Social relation	81.40	81.40	90.70	83.72	81.40	79.07
Spatial relationship	28.89	26.67	24.44	28.89	26.67	28.89
Structured image–text understanding	38.46	37.18	29.49	38.46	37.18	35.90
Overall	62.71	62.46	63.06	62.89	62.71	61.94

B.7 MMStar

Table 25 reports overall and category accuracies. Bilinear interactions and learned weighting do not improve every category.

Table 25: MMStar overall and category accuracies.

Metric	B-L	B-U	S-L	Global	Local	Hybrid
Coarse perception	57.60	63.20	60.00	60.80	58.40	57.60
Fine-grained perception	27.20	28.00	27.60	25.20	25.20	26.80
Instance reasoning	51.20	51.20	47.20	51.20	51.20	52.40
Logical reasoning	36.80	34.80	38.40	36.00	36.00	38.40
Math	34.80	33.60	34.80	35.60	37.60	36.00
Science & technology	21.60	23.20	22.80	21.20	20.00	21.20
Overall	38.20	39.00	38.47	38.33	38.07	38.73

C Paired Evaluation Uncertainty

We estimate evaluation-sample uncertainty for four benchmarks using paired bootstrap resampling of the existing B-L, B-U, and S-L predictions. We independently reimplement the analysis from these predictions using the procedure below. For each included benchmark, we draw 20,000 resamples using the units below, with the same sampled units across all three configurations within each replicate. Each benchmark starts a separate NumPy PCG64 generator with seed 42. Table 3 reports B-L minus each comparator in percentage points, with 95% intervals defined by the 2.5th and 97.5th percentiles of the bootstrap differences. These intervals condition on the evaluated checkpoints and saved answer-extraction decisions and do not account for variation across training seeds. The bootstrap uses fixed extracted decisions without new model inference or judge calls.

Resampling units and scores. For NaturalBench, we sample 1,900 official four-answer groups with replacement and report individual-answer accuracy, averaging the number of correct answers per group divided by four. Answers are extracted with the archived deterministic word-boundary rule. For CV-Bench-2D, we independently resample 633 ADE20K and 805 COCO source images within their respective strata and average the two source accuracies with equal weight, as in the original evaluator. Each exported row corresponds to a distinct source image. For CV-Bench-3D, the 1,200 questions refer to 674 source images identified jointly by their source dataset and image identifier. We sample these image clusters with replacement and recompute total correct answers divided by total questions in the sampled clusters; thus the score remains question-weighted even when cluster sizes differ. For MMBench, the 4,329 answer-option permutations are already aggregated into 1,164 base-question circular groups. We resample those groups and average their archived all-permutation correctness indicators.

Interpretation and coverage. These are descriptive, unadjusted intervals conditional on the existing checkpoints, benchmark sample, and saved answer-extraction decisions. They quantify evaluation-sample uncertainty under the stated resampling assumptions, not variation across training seeds, training sets, or repeated judge calls. They do not remove potential dependence between distinct images from related scenes or sources. The B-L versus S-L comparisons on CV-Bench-2D and CV-Bench-3D retain opposite directions in these intervals; this describes opposite benchmark-score differences conditional on the fitted systems, without establishing their stability across training runs or a router-specific effect. The other six intervals span zero; these intervals do not establish equivalence. No multiple-comparison correction or confirmatory significance claim is made. We restrict this analysis to the four benchmarks whose relevant grouping is recoverable from the saved exports. POPE paths are indexed per exported record without the original image identifiers, and the SEEDBench-IMG and MMStar result exports lack source-image identifiers; their source-image dependence cannot be reliably reconstructed from these files alone.

D Diagnostics from Saved Predictions

We analyze existing predictions and correctness decisions after aligning questions, reference answers, answer options, and available grouping metadata across configurations. Scores are computed using the benchmark-specific aggregation rules described below and agree with the original evaluation summaries.

Answer-option selections. Every CV-Bench-3D record has two available choices, A and B. The reference distribution is 300 A and 300 B for depth, and 303 A and 297 B for distance. The complete set therefore contains 603 A and 597 B answers. Table 26 reports selections for the three fitted configurations. The saved prediction strings are A or B, except for one B-U response, `B sofa`, which is counted as B. Comparing these extracted options with the reference answers reproduces all 1,200 archived correctness decisions for each configuration. The observed distribution is specific to the saved prompts and option order; it does not measure sensitivity to counterfactual option permutations.

Table 26: CV-Bench-3D option selections, including both subtasks. Counts are based on saved responses. Accuracy and the fraction selecting A are percentages.

Task	Model	N	A count	B count	A rate	Accuracy
Overall	B-L	1,200	1,163	37	96.92	51.33
Depth	B-L	600	567	33	94.50	52.50
Distance	B-L	600	596	4	99.33	50.17
Overall	B-U	1,200	935	265	77.92	54.17
Depth	B-U	600	525	75	87.50	54.17
Distance	B-U	600	410	190	68.33	54.17
Overall	S-L	1,200	946	254	78.83	54.58
Depth	S-L	600	517	83	86.17	55.50
Distance	S-L	600	429	171	71.50	53.67

Correctness transitions under family restriction. For each saved answer record, we compare the full B-L mixture's correctness with the recorded global-, local-, or hybrid-family run. Table 27 reports both directions of change. For MMBench, a record is an all-permutation circular question group, so a transition concerns group correctness. These counts show that improvements and regressions between the recorded outputs can cancel in an aggregate score. The source versions differ in four files (Appendix F), so the transitions cannot be attributed to the intended restriction alone. They do not identify which expert representations differ or establish functional specialization.

The POPE transition counts use the 5,127 unexpanded exported records. The primary POPE F1 instead follows the archived evaluator, expanding comma-separated category memberships to 9,000 category records before computing precision and recall. Consequently, the unweighted POPE transition counts cannot be converted directly into the reported F1 difference. CV-Bench-2D likewise uses a source-averaged primary score, whereas its transition percentages pool exported records. For the remaining accuracy metrics, the primary difference equals the net correctness change divided by the corresponding record count.

Correctness versus prediction disagreement. A correctness transition necessarily changes the evaluation outcome. It does not count changes from one incorrect answer to another, and different prediction strings can yield the same extracted answer. The compact machine-readable analysis therefore stores raw string changes separately, after trimming leading and trailing whitespace. For MMBench, those string counts refer only to the saved representative base-question rows; they do not summarize disagreement across every option permutation. Table 27 uses correctness transitions throughout.

Table 27: Recorded family-run correctness transitions relative to B-L, with the source-version limitation described above. $C \to W$ and $W \to C$ are counts; Changed is their sum divided by N, expressed as a percentage; Δ is the primary-score difference in percentage points. MMBench uses circular groups. POPE uses unexpanded records for counts and expanded-category F1 for Δ.

Benchmark	Family	N	$C \to W$	$W \to C$	Changed	Δ
NaturalBench	Global	7,600	141	138	3.67	-0.04
NaturalBench	Local	7,600	146	103	3.28	-0.57
NaturalBench	Hybrid	7,600	135	100	3.09	-0.46
CV-Bench-2D	Global	1,438	42	34	5.29	-0.56
CV-Bench-2D	Local	1,438	28	29	3.96	+0.05
CV-Bench-2D	Hybrid	1,438	42	30	5.01	-0.78
CV-Bench-3D	Global	1,200	10	22	2.67	+1.00
CV-Bench-3D	Local	1,200	4	3	0.58	-0.08
CV-Bench-3D	Hybrid	1,200	2	3	0.42	+0.08
MMBench	Global	1,164	14	16	2.58	+0.17
MMBench	Local	1,164	11	11	1.89	+0.00
MMBench	Hybrid	1,164	17	8	2.15	-0.77
MMStar	Global	1,500	33	35	4.53	+0.13
MMStar	Local	1,500	24	22	3.07	-0.13
MMStar	Hybrid	1,500	19	27	3.07	+0.53
POPE	Global	5,127	17	18	0.68	-0.09
POPE	Local	5,127	33	22	1.07	+0.30
POPE	Hybrid	5,127	39	25	1.25	-0.79
SEEDBench-IMG	Global	14,232	178	165	2.41	-0.09
SEEDBench-IMG	Local	14,232	161	128	2.03	-0.23
SEEDBench-IMG	Hybrid	14,232	182	159	2.40	-0.16

E Prediction-Level Examples

Table 28 lists matched examples selected to illustrate both correct and incorrect answer changes relative to the two trained alternatives. Each displayed prediction and reference answer is verified against the archived per-example records. For multiple-choice examples, the answer option is decoded to its numerical content. The examples are illustrative and do not estimate the frequency of improvements or regressions. The two POPE rows ask the same question about different images. The saved outputs do not contain the corresponding images, interaction maps, or expert weights, so this table supports response-level comparisons only.

Table 28: Matched prediction examples from the archived evaluation records. IDs are dataset indices. The POPE question is shortened only by omitting "Please answer yes or no." B-L, B-U, and S-L follow Appendix B.

Benchmark / ID	Question	Reference	B-L	B-U	S-L
CV-2D / 44	How many blankets are in the image?	1	1	0	0
CV-2D / 5	How many beds are in the image?	1	2	1	1
POPE / 49	Is there a bottle in the image?	No	No	Yes	Yes
POPE / 32	Is there a bottle in the image?	No	Yes	No	No

F Implementation Details and Reproduction Scope

This appendix describes the implementation and training settings documented by the available code and experimental records. Evaluation-version differences relevant to interpreting the results are summarized below.

Training-data records. The available B-L and S-L logs for both stages and the B-U stage-2 log record 450,330 training examples and 4,549 validation examples, with split seed 42 and validation-fraction setting 0.01. They record no sample cap and no additional exclusions by the 1,024-token training-prompt filter. These counts come directly from run metadata. The available preprocessing implementation pins `lmms-lab/LLaVA-NeXT-Data` at revision `dff8edfca0f16f9b4f1407331f6659aa881fc016`, retains the first nonempty user–assistant exchange, removes image placeholders, and excludes missing-image, incomplete-pair, and duplicate-identifier records. It partitions complete groups sharing identical encoded-image bytes between training and validation. These implementation details are distinguished from the logged counts: the archive does not establish the exact historical preprocessing source or provide per-source exclusion tallies.

Feature extraction and text handling. The implementation uses `openai/clip-vit-large-patch14-336` as one CLIP model containing distinct vision and text towers. Both towers remain frozen and in evaluation mode; their trainable projection MLPs are connector components. Images are resized to a 336×336 square without center cropping. Visual features come from the penultimate vision layer, with the class token removed, before CLIP's final visual normalization and projection. Text features use the final text hidden states followed by CLIP's frozen text projection and then the trainable connector MLP. The text encoder removes attention-masked padding and the leading beginning-of-sequence and trailing end-of-sequence tokens. It divides the remaining CLIP subword tokens into consecutive, nonoverlapping chunks of at most 75 tokens, without a linguistic content filter. Each chunk is wrapped with its own beginning/end tokens and encoded independently; CLIP attention does not cross chunk boundaries. The projected content features exclude those boundary tokens and are concatenated in question order, with a sinusoidal chunk-index encoding whose scale is trainable. The valid-token set $\mathcal{I}$ contains these retained positions; batch padding is masked out of token aggregation. An empty input receives one end-token feature as a placeholder, ensuring a nonempty aggregation set.

Training question–answer text is constrained by the LLM token budget. The training and generation configuration sets a maximum of 1,024 text tokens in training, a 2,048-token prompt limit at generation, and a 4,096-token assembled sequence limit. The trainer filters prompts that cannot accommodate supervised answer tokens under its budget. Collation itself does not truncate the CLIP question sequence. The three visual prefix blocks retain all 576 patch positions and are inserted at the multimodal marker inside the user message. Ordinary question text is also present in that message.

In the available implementation, training supplies the retained first user question to CLIP, without the assistant answer or later conversation turns. At evaluation, CLIP receives the benchmark text payload, including options and instructions already present. The additional model-level option-letter instruction is appended only when constructing the language-model chat. These descriptions concern the available source; its correspondence to historical evaluation is qualified below.

Magnitude feature. The scalar channel accompanying the normalized pair feature is

$$q_{ij} = a\left[\log\left(1+\sqrt{r^{-1}\|p_{ij}\|_2^2+\epsilon}\right)-\log\left(1+\sqrt{\epsilon}\right)\right]+b, \qquad \epsilon = 10^{-6}, \tag{11}$$

where a and b are learned and initialized to one and zero. The complete pair feature is $h_{ij} = [\mathrm{GELU}(\mathrm{LN}(p_{ij})); q_{ij}]$. Pair projections, the token-scoring projection, and the two-layer map MLP use no bias. The sum-based arm changes the pair operator before this channel is computed. Consequently, its magnitude distribution need not match that of the bilinear arm even though the downstream architecture is shared.

Training objective. We minimize answer-token cross-entropy with router regularization. For a distributed microbatch of B examples,

$$\begin{aligned} \bar{g}_e &= B^{-1}\sum_{b=1}^{B} g_{be}, & \mathcal{L}_{\mathrm{bal}} &= E\sum_{e=1}^{E}\bar{g}_e^2 - 1, \\ \mathcal{L}_z &= B^{-1}\sum_{b=1}^{B}\left(\log\sum_{e=1}^{E}\exp\ell_{be}\right)^2, & \mathcal{L} &= \mathcal{L}_{\mathrm{ans}} + 0.01\left(0.1\mathcal{L}_{\mathrm{bal}} + 0.001\mathcal{L}_z\right). \end{aligned} \tag{12}$$

Here g includes training noise and ℓ denotes clean logits. The effective coefficients of $\mathcal{L}_{\text{bal}}$ and $\mathcal{L}_z$ are 10^{-3} and 10^{-5}. The balance term encourages uniform mean expert usage across examples, without requiring uniform weights for each example. Losses are aggregated with global target-token weighting over an optimizer window, as specified below. The uniform training variant fixes $g_e = 1/E$ and bypasses the router and its auxiliary loss.

Recorded training and decoding configuration. The connector uses width 512, interaction rank 64, and 128 map channels. The router and experts have depth two, eight attention heads where applicable, and dropout 0.05. Three blocks of 576 embeddings form the 1,728-token prefix. LoRA uses rank 64 and scaling parameter 128. The training configuration specifies one connector epoch at 3×10^{-4} followed by two connector-and-LoRA epochs at 10^{-4}. AdamW uses $\beta = (0.9, 0.95)$ and weight decay 0.05 on eligible connector matrices, with no decay on biases, normalization parameters, or LoRA weights. Cosine decay follows warmup fractions 0.10 and 0.05 for the respective stages. Recorded runs use six distributed workers, microbatches of ten, and six accumulation steps, giving a nominal effective batch of 360, with BF16 autocasting. Training uses length-grouped batches. IP-MoE uses greedy decoding with at most 32 new tokens. The available generation implementation appends a request for exactly one option letter and no explanation to detected multiple-choice questions; other questions receive no such suffix. This source-level behavior has not been verified against every historical evaluation version. No hyperparameter sensitivity study is reported, and stage-1 B-U training measurements are unavailable. Appendix G gives measured costs.

Reference-model identifiers and evaluation scope. Table 29 lists the model identifiers documented in the evaluation configuration and inference-cost records for the reference rows in Table 4. Their wrappers use model-specific image preprocessing and prompt formatting. The available baseline implementations default to greedy decoding with at most 256 new tokens, whereas IP-MoE records a 32-token budget. Historical baseline generation overrides are not established by the saved answer exports. Consequently, the reference table is not a comparison with matched backbones, training data, prompts, or decoding budgets.

Table 29: Reference-model identifiers documented in the evaluation configuration and cost records.

Reference model	Identifier
BLIP-2 Flan-T5-XL	`Salesforce/blip2-flan-t5-xl`
InstructBLIP Flan-T5-XL	`Salesforce/instructblip-flan-t5-xl`
Mono-InternVL	`OpenGVLab/Mono-InternVL-2B`
SpatialLadder	`hongxingli/SpatialLadder-3B`
Spatial-SSRL	`internlm/Spatial-SSRL-3B`

Optimization and aggregation. The balance and logit regularizers in Equation 12 use statistics synchronized over the distributed microbatch. Their gradients are accumulated with weights proportional to the microbatch's global number of supervised answer tokens. The language loss is likewise normalized over global target tokens in an optimizer window, rather than averaging per-example answer losses. Balancing is applied separately to each microbatch; it is not recomputed from a gate marginal over the entire accumulation window. LoRA is attached to the attention query, key, value, and output projections and to the feed-forward gate, up, and down projections. Stage 1 gives LoRA a zero learning rate; stage 2 updates it. Biases, normalization parameters, and LoRA weights have zero weight decay.

Inference interventions. For a selected expert pair S, let $m_S = \sum_{k \in S} g_k$ and let $I_S(e)$ equal one for $e \in S$ and zero otherwise. Family restriction uses

$$g_e^{(S)} = I_S(e) \begin{cases} g_e/m_S, & m_S > 10^{-6}, \\ 1/|S|, & m_S \le 10^{-6}. \end{cases} \tag{13}$$

Thus, retained probabilities are divided by their sum, with equal weights within the pair as the small-mass fallback. Every expert is still evaluated before the weighted sum, so the restriction is not a latency or sparsity experiment. Both direct image and interaction prefixes remain. A separate generation-time uniform-gate option exists in the implementation, but no result for that fixed-

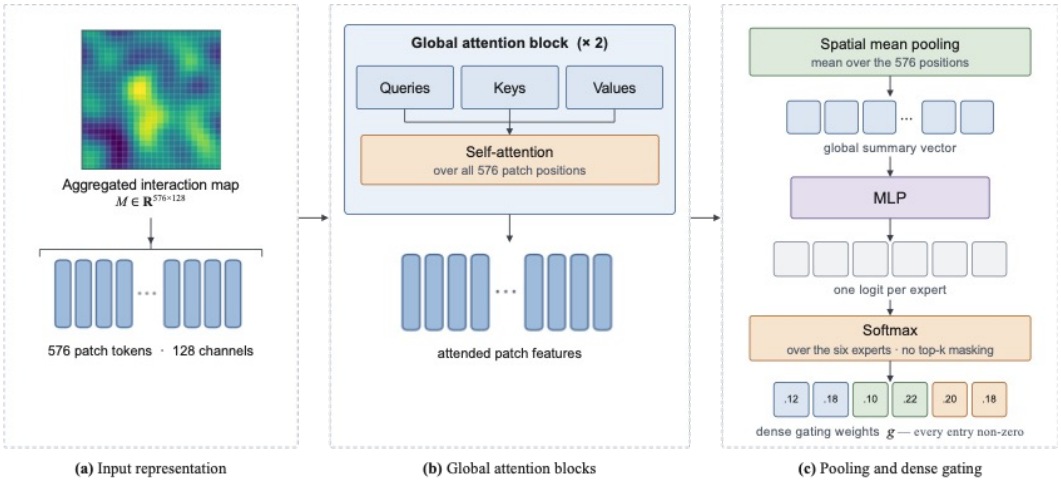
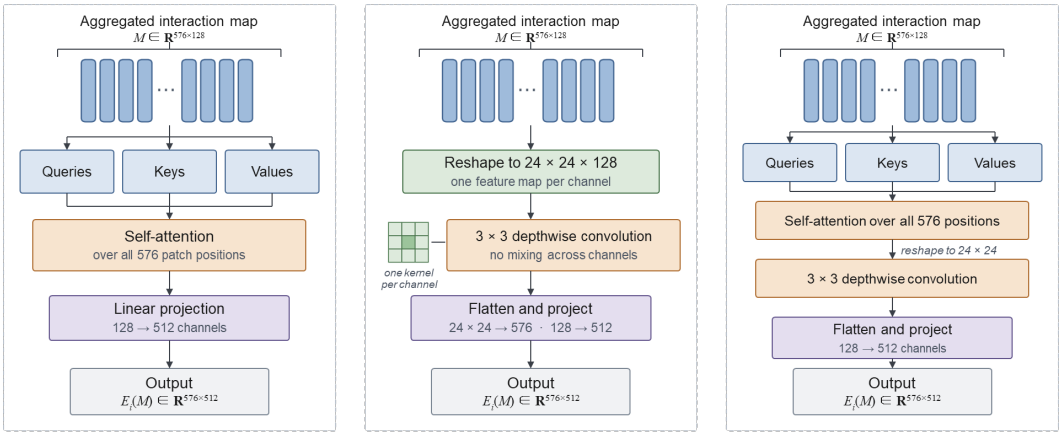


Figure 2: The IP-MoE router broken down into its components. (a) The aggregated interaction map is read as 576 patch tokens of 128 channels. (b) Two global attention blocks project those tokens to queries, keys and values and attend over all 576 positions. (c) Spatial mean pooling reduces the attended features to a single global summary vector, an MLP turns it into one logit per expert, and a softmax over the six experts gives the gating weights. No top-k masking is applied, so every weight is non-zero and all six experts contribute; the values shown are illustrative.

checkpoint control appears in the reported tables; B-U instead denotes a separately trained uniform model that bypasses the router and its auxiliary loss.

Evaluation-version differences. Evaluation metadata records distinct checkpoints for B-L, B-U, and S-L and the same source version for their main evaluations. The three expert-family evaluations record the B-L checkpoint but use a different source version, with differences in the evaluation driver, model configuration, expert connector, and model adapter. The analysis archive does not include the evaluated weights or complete matching historical source, so the behavioral effects of these software differences cannot be verified from these materials. Consequently, the family comparisons describe the recorded predictions and do not isolate the effect of expert-family restriction. The available predictions and training records support the reported analyses, but are insufficient to reproduce model inference exactly.

Additional architecture diagrams. Figures 2 and 3 expand the routing and expert-processing paths of the overview in Figure 1. They depict the architecture described in Section 3.



(a) Global expert (× 2) **(b)** Local expert (× 2) **(c)** Hybrid expert (× 2)

Figure 3: Each expert first projects the 128-channel interaction map to 512-dimensional spatial tokens and adds positional encodings. Two global, local, or hybrid blocks process these tokens, followed by a final layer normalization.

G Computational Costs

Inference protocol. Table 30 reports measurements on an NVIDIA A100-SXM4-40GB GPU using batch size one, seed 42, the same synthetic 336×336 RGB image, and the prompt "Describe the main objects visible in this image." Each run generates exactly one token after three warmup calls, with latency summarized over ten timed calls. Timing includes native image preprocessing, image encoding, prompt prefill, the first prediction, and text decoding. These measurements describe one fixed input, not benchmark-average latency or sustained decoding throughput.

Table 30: One-token inference costs under native evaluation wrappers. Parameters include all expert weights. Counted operations are reported in TFLOPs; latency mean and standard deviation (SD) are in milliseconds; memory is peak allocated GPU memory in GiB, including resident weights.

Model	Params. (B)	Counted TFLOPs	Mean (ms)	SD (ms)	Peak (GiB)
BLIP-2 OPT-2.7B	3.745	0.783	44.92	2.02	7.04
BLIP-2 Flan-T5-XL	3.942	0.665	79.92	1.35	7.36
InstructBLIP Flan-T5-XL	4.023	0.656	80.93	2.19	7.51
Kosmos-2	1.664	0.396	33.82	0.19	3.14
Mono-InternVL	3.111	1.094	59.39	1.52	6.45
MolmoE	7.222	1.473	706.84	7.61	13.69
Spatial-SSRL	3.755	15.117	203.57	8.03	7.26
SpatialLadder	3.755	1.040	101.25	1.00	7.02
IP-MoE: bilinear, learned	3.774	11.865	181.75	0.65	8.54
IP-MoE: bilinear, uniform	3.774	11.858	180.29	0.57	8.54
IP-MoE: sum-based, learned	3.774	11.865	182.76	0.61	8.54

Scope of comparison. The operation counter includes supported matrix multiplication, convolution, and attention operations, with a multiply–add counted as two operations. Normalization, softmax, other elementwise operations, causal-mask savings, and unsupported custom operations are excluded. The values therefore describe counted operations, not exhaustive FLOP totals. Native image preprocessing, prompt templates, and precision are preserved: Kosmos-2 uses FP16 weights, most external baselines use BF16, and IP-MoE uses mixed FP32/BF16 weights. Spatial-SSRL receives a 72×72 image grid with 1,296 merged visual tokens and 1,326 input IDs, whereas SpatialLadder receives an 18×18 grid with 81 merged visual tokens and 111 input IDs. IP-MoE uses 1,728 prefix embeddings. These differences limit direct architectural comparisons; prefix length

alone does not establish a proportional increase in total inference cost. The reports identify final stage-2 checkpoints.

Training measurements. Table 31 summarizes the available epoch-level training costs. IP-MoE was trained on six NVIDIA A100-SXM4-40GB GPUs using one distributed worker per GPU, a per-worker microbatch size of ten, and six gradient-accumulation steps, yielding a nominal effective batch size of 360.

Table 31: Measured IP-MoE training costs by stage and epoch. Time is the maximum elapsed training time across workers; allocated and reserved memory are the maximum per-worker peaks. Throughput is global examples per second. Missing uniform stage-1 records are not imputed.

Interaction, weighting	Stage	Epoch	Time (h)	Alloc. (GiB)	Reserv. (GiB)	Ex./s
Bilinear, learned	1	1	8.557	28.84	33.04	14.62
Bilinear, learned	2	1	8.534	29.57	34.03	14.66
Bilinear, learned	2	2	8.623	29.57	34.03	14.51
Bilinear, uniform	2	1	8.473	29.26	33.05	14.76
Bilinear, uniform	2	2	8.526	29.26	33.50	14.67
Sum-based, learned	1	1	8.596	28.84	33.04	14.55
Sum-based, learned	2	1	8.538	29.57	34.03	14.65
Sum-based, learned	2	2	8.598	29.57	34.03	14.55

The three recorded epochs sum to 25.714 hours for bilinear interactions with learned weighting and 25.732 hours for sum-based interactions with learned weighting. The two recorded stage-2 epochs for uniform weighting sum to 16.999 hours and are not a complete two-stage training total. All three models contain 3,774,372,430 parameters. The learned-weighting runs record 36,422,989 parameters with active updates in stage 1 and 133,678,413 in stage 2; the uniform stage-2 run records 129,139,015.

H TRAINING AND ROUTER DIAGNOSTICS

The recorded training diagnostics characterize optimization and gate behavior. Figures 4–6 reproduce the saved numerical records for B-L and S-L in both stages and B-U in stage 2. Stage-1 B-U training measurements are unavailable. These are the existing seed-42 runs.

Sampling and aggregation. The logs record every 100 optimizer steps and each epoch end: 13 records in stage 1 and 27 in stage 2, with 1,251 updates per epoch. Step counts restart at the beginning of each stage. Training language-model loss and router loss are weighted by the number of supervised answer tokens across the distributed gradient-accumulation window. They describe the logged window, not an epoch average or an average of the preceding 100 steps. Total loss is reconstructed from the rounded logged language loss and weighted router contribution. Validation language loss is evaluated at each epoch end on the training–validation split, separately from the seven benchmark evaluations.

Gate statistics describe the *last training microbatch* of the logged window, aggregated across the six ranks. They do not cover the full accumulation window or the training set. Recorded runs use length-grouped batches, so within-microbatch variation should not be interpreted as variation over an unrestricted sample of inputs. The archived implementation and log metadata specify the following definitions; source-version correspondence is described in Appendix F.

Clean and noisy gates. For logits z_{be} of example b and expert e, the recorded clean gates use $g_{be}^{\text{clean}} = \text{softmax}_e(z_b/\tau)$, whereas the gates used for training use $g_{be}^{\text{noisy}} = \text{softmax}_e((z_b + \epsilon_b)/\tau)$, with $\tau = 0.7$ and $\epsilon_{be} \sim \mathcal{N}(0, 0.3^2)$. "Clean" removes the explicit logit noise; it still uses training-mode activations, including connector dropout. It is therefore not a held-out or deterministic evaluation-mode measurement.

For either gate vector, the reported entropy and gate standard deviation are

$$H = -\frac{1}{B}\sum_{b=1}^{B}\sum_{e=1}^{6} g_{be}\log g_{be}, \qquad s_g = \frac{1}{6}\sum_{e=1}^{6}\sqrt{\frac{1}{B}\sum_{b=1}^{B} g_{be}^2 - \bar{g}_e^2}, \quad \bar{g}_e = \frac{1}{B}\sum_{b=1}^{B} g_{be}, \tag{14}$$

where B is the actual number of examples in that distributed microbatch. The implementation clamps small negative variances caused by rounding to zero. Entropy is in nats and is at most $\ln 6 \approx 1.791759$. The statistic s_g averages each expert's population standard deviation across examples; it is neither a standard error nor the deviation of each gate from $1/6$.

Table 32: Gate diagnostics at the last logged optimizer window of each available learned-routing stage. Every entry describes that window's final distributed microbatch. B-U has fixed uniform gates and no recorded gate-diagnostic fields.

Variant	Stage	Clean H	Noisy H	Clean s_g	Noisy s_g
B-L	1	1.218156	1.168369	0.012170	0.059030
B-L	2	1.783823	1.711984	0.011340	0.064340
S-L	1	1.551411	1.490089	0.030560	0.070130
S-L	2	1.789708	1.719354	0.009397	0.064490

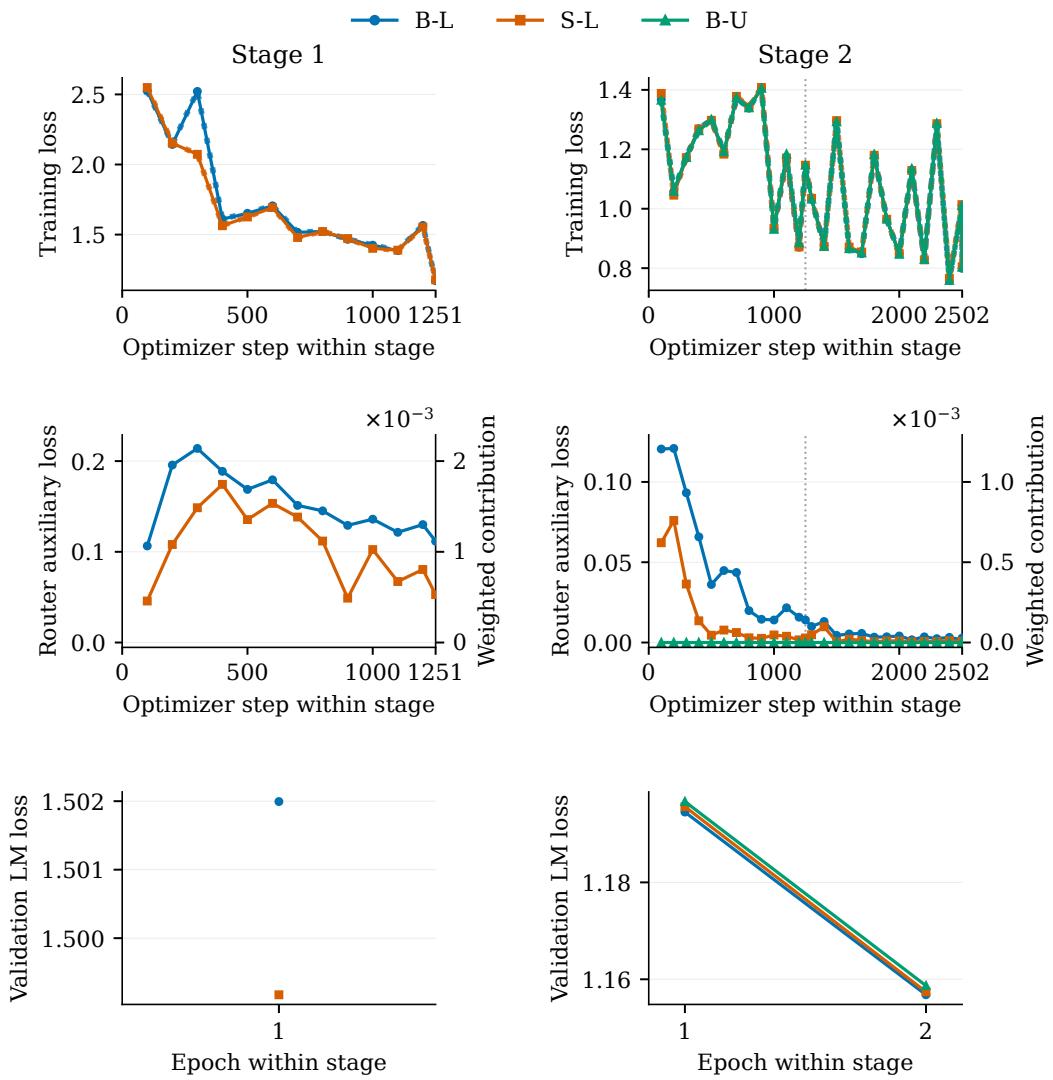


Figure 4: Recorded optimization and validation losses. Top: language-model loss (solid) and reconstructed total loss (dotted) nearly overlap because the weighted router contribution is small. Middle: the logged router auxiliary loss $0.1\mathcal{L}_{\text{bal}} + 0.001\mathcal{L}_z$; the right axis gives its 0.01-weighted contribution to the objective. The right axis is an exact rescaling; separately rounded logged weighted values can differ in their last displayed digit. B-U's router term is zero. Bottom: epoch-end validation language loss; stage 1 has one observation per available run. Curves connect logged measurements without smoothing. The vertical dotted line at step 1,251 marks the stage-2 epoch boundary. Stage-1 B-U measurements are unavailable.

Loss interpretation. Validation loss decreases between the two stage-2 epochs in all three variants, reaching 1.156804 for B-L, 1.157402 for S-L, and 1.158682 for B-U. These nearby values do not establish a reliable ranking across training seeds. The displayed training losses vary with the particular optimizer window and its answer-token composition; they are sampled window losses, whereas validation loss averages the full validation pass. The regularization term decreases markedly during stage 2, but a small scalar contribution alone does not determine the influence of its gradients.

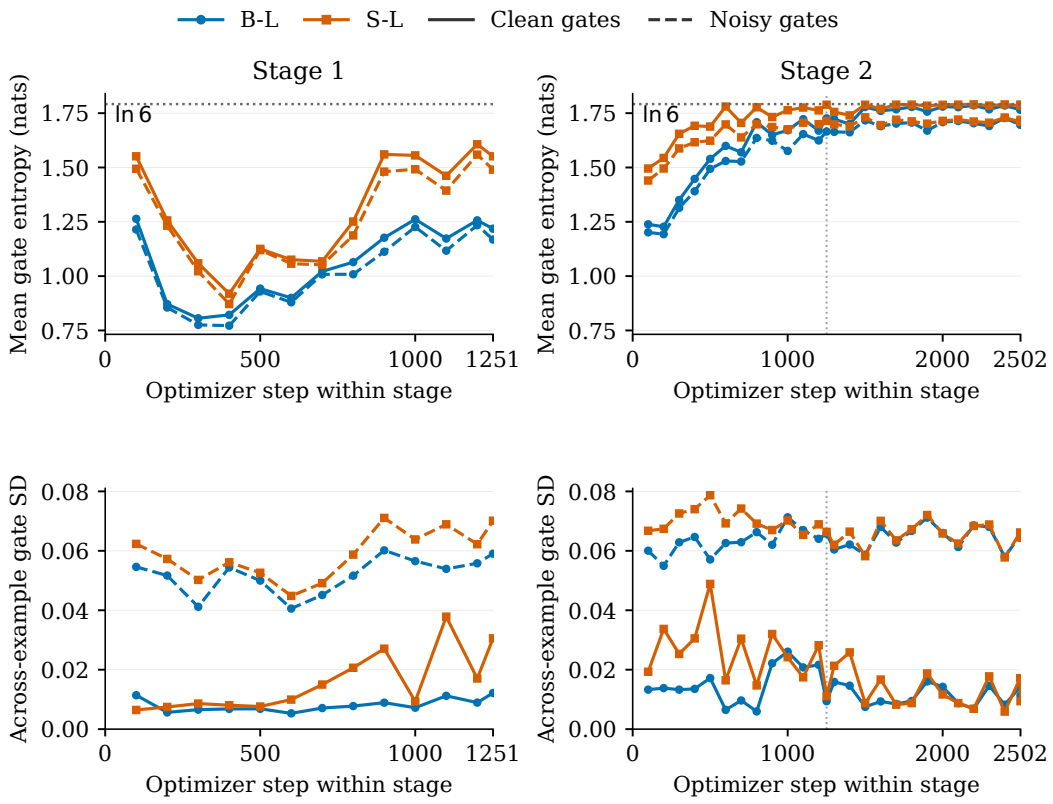


Figure 5: Recorded training gate diagnostics, retaining both clean and noisy measurements. Color identifies the interaction variant; solid and dashed lines identify the clean and noisy gates, respectively. Top: mean per-example entropy, with the uniform reference $\ln 6$. Bottom: the mean across experts of their gate standard deviations across examples, defined in Equation 14. Curves connect the available microbatch snapshots without smoothing. The vertical dotted line marks the stage-2 epoch boundary. The same axis ranges are used across the two stages.

What the gate records show. The saved clean-gate entropies are substantially below the uniform reference during stage 1 and increase toward it during stage 2, with fluctuations across sampled microbatches. The final values in Table 32 support the description of *nearly uniform gates late in these training runs*. For B-L, exponentiating the final mean clean entropy gives $\exp(1.783823) \approx 5.953$; for S-L it gives $\exp(1.789708) \approx 5.988$. This is the geometric mean of the per-example entropy-based effective counts, not the number of experts executed: all six experts execute for every example.

At the stage-1 endpoint, B-L combines nonuniform clean weights ($H = 1.218156$) with limited across-example variation ($s_g = 0.012170$) in the sampled microbatch. Its stage-2 endpoint additionally approaches uniform weighting. Small s_g alone does not imply uniform weights: a common nonuniform vector would also have $s_g = 0$. Nor does a small within-microbatch standard deviation establish a constant gate vector over all inputs. Noisy gates have lower entropy and larger across-example variation than clean gates in the final records; this variation includes the deliberately added logit noise.

The records document late-training near-uniformity under both interaction operators. They do not measure held-out routing or distinguish the effects of the training objective, interaction representation, and direct prefix paths. Functional specialization or redundancy requires evidence about expert outputs and their effect on predictions in addition to gate statistics.

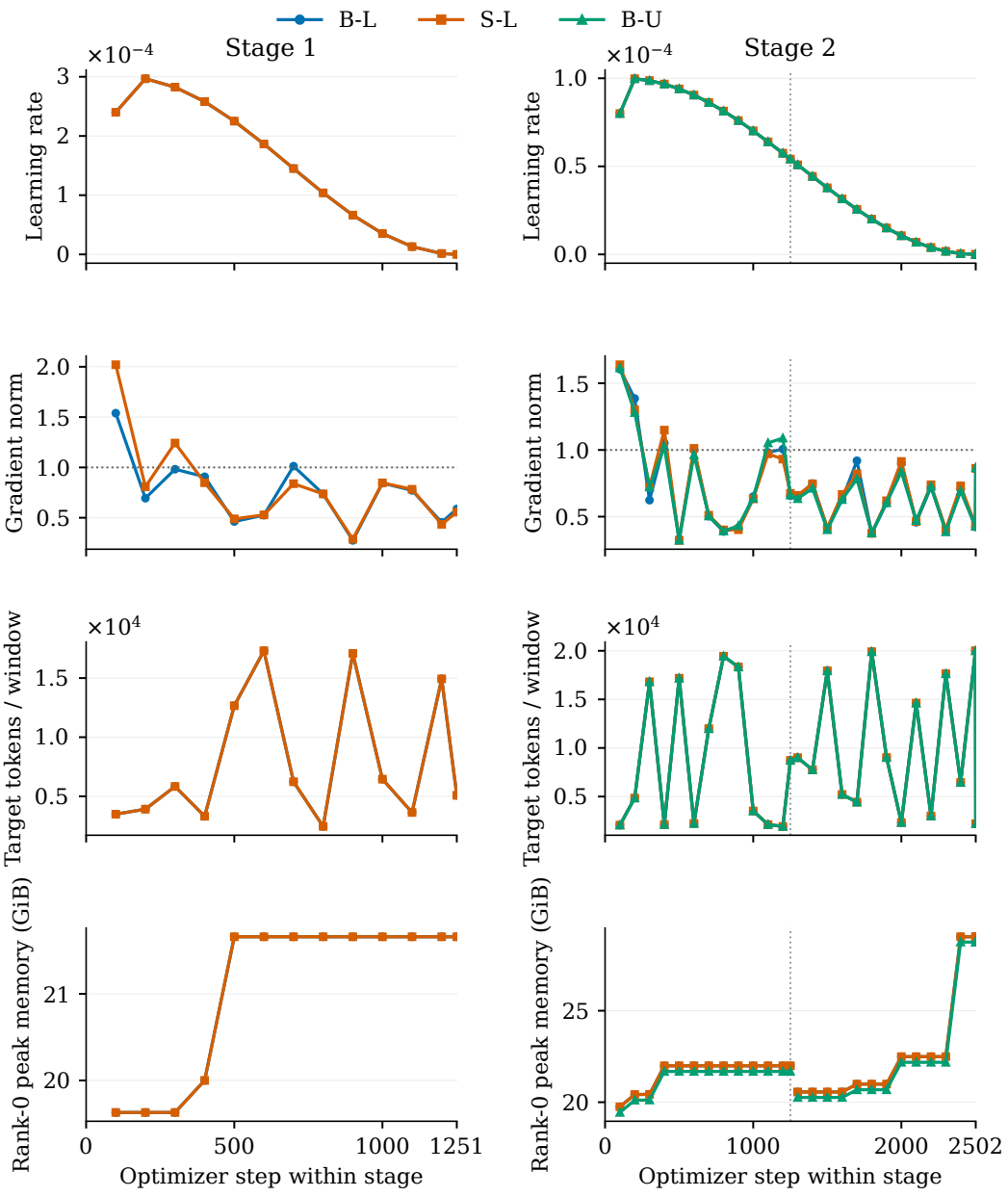


Figure 6: Other recorded training quantities. Learning rate is the first optimizer group's post-step logged value; it need not represent every parameter group. Gradient norm is measured before clipping on active learning-rate groups, with the dotted horizontal line marking the configured threshold of 1. Target-token counts are totals for the logged optimizer window across ranks. Memory is the running peak allocated memory on rank 0 within each epoch, so its stage-2 curve is separated at the epoch boundary. This differs from the maximum across ranks reported in the epoch cost records. Overlapping curves are retained. Points are connected without smoothing; unlogged intervals are not additional observations.

The training records also contain epoch-level timing, throughput, sample-count, and allocated/reserved-memory fields, summarized in Appendix G.

I Interpreting Gate and Prediction Behavior

This appendix relates the recorded behavior to the architecture. Expert activations and gradients were not measured, so the derivations identify possible mechanisms rather than an observed internal cause. Source-version limits are given in Appendix F.

I.1 What Determines the Router's Answer-Loss Gradient?

For one example, write $u_{ej} = \mathrm{LN}_{\mathrm{mix}}(E_e(M)_j)$ and $z_j = \sum_k g_k u_{kj}$. Define $\delta_j = \partial \mathcal{L}_{\mathrm{ans}} / \partial z_j$, which includes the expert-prefix projector and all subsequent decoder computations. Holding the expert features and a sampled noise vector fixed, the softmax Jacobian gives

$$\frac{\partial g_k}{\partial \ell_e} = \frac{1}{\tau} g_k(\mathbf{1}_{k=e} - g_e), \qquad \frac{\partial z_j}{\partial \ell_e} = \frac{g_e}{\tau}(u_{ej} - z_j). \tag{15}$$

Applying the chain rule gives

$$\frac{\partial \mathcal{L}_{\mathrm{ans}}}{\partial \ell_e} = \frac{g_e}{\tau} \sum_{j=1}^{N} \langle \delta_j, u_{ej} - z_j \rangle . \tag{16}$$

This is a partial derivative with respect to router logits; shared interaction-encoder parameters also receive gradients through expert and direct-prefix paths. Auxiliary losses contribute further router gradients. The identity is not an empirical estimate, and small local derivatives would not by themselves imply insensitivity to finite interventions or held-out inputs. Three conditions can reduce the answer-loss gradient:

1. **Low downstream sensitivity.** If δ_j is small, changing the mixture has little local influence on the answer loss. Other prefix streams and ordinary text may already provide sufficient predictive information. Their existence makes this possible but does not demonstrate that the decoder uses them in this way.
2. **Small answer-relevant differences.** Experts may produce similar features, or distinct features whose differences have small inner products with δ_j. Thus, a small router answer-loss gradient does not require identical expert representations. Conversely, large representation distances alone would not establish useful specialization.
3. **Cancellation across positions.** For a shared gate, positive and negative contributions from different patches are summed. Spatially heterogeneous preferences can therefore give a small net update even if individual positions favor different experts. No such cancellation was measured here.

At the mixture interface, $\partial \mathcal{L}_{\mathrm{ans}} / \partial u_{ej} = g_e \delta_j$. Near-uniform gates give experts similarly scaled supervision at that interface, but their architectures and parameter Jacobians can still produce different functions. Also, $g_e \simeq 1/E$ does not suppress the softmax Jacobian to zero. High gate entropy therefore does not demonstrate gradient saturation, expert collapse, or an uninformative interaction representation.

I.2 Where the Interaction Representation Can Be Restrictive

Let $a_i = W_T \,\mathrm{LN}(t_i)$ and $b_j = W_V \,\mathrm{LN}(v_j)$. Before subsequent normalization and nonlinear processing, the product $p_{ij} = a_i \odot b_j / \sqrt{r}$ has derivatives

$$\frac{\partial p_{ij,c}}{\partial a_{i,c}} = \frac{b_{j,c}}{\sqrt{r}}, \qquad \frac{\partial p_{ij,c}}{\partial b_{j,c}} = \frac{a_{i,c}}{\sqrt{r}}. \tag{17}$$

Each modality modulates the other modality's coordinate and its local derivative. Summation instead has a constant derivative $1/\sqrt{r}$ at this point. This identifies a possible sensitivity to feature alignment and magnitude, rather than an observed optimization failure. Learned projections, layer normalization, GELU, and the magnitude channel alter the complete forward and backward maps; a small pre-normalization factor alone does not imply a vanishing end-to-end gradient. Product coordinates are signed learned features and cannot be read literally as semantic agreement scores.

The scalar magnitude channel preserves one summary of the product's size without recovering the original modality-specific vectors. Both variants compress their inputs. Learned projections allow product coordinates to combine original feature dimensions, while nonlinear processing after summation also permits cross-modal interactions. S-L is therefore not an interaction-free control. The unresolved question is which answer-relevant information each construction makes accessible to downstream processing. Replacing the operator also changes the magnitude exposed to the map network; that scale was not independently matched.

Patchwise aggregation introduces a second restriction shared by both variants. It produces one vector per image patch before any expert processes the map. Experts cannot choose their own aggregation over the original question-token axis, although they can transform the resulting contextual features differently. Likewise, the router's global gate cannot express separate expert mixtures for separate patches. These restrictions are distinct from spatial resolution: the grid and positional encodings remain present, and attention can model relations. A failure to learn useful geometric representations cannot be inferred merely from averaging or from the fact that the upstream encoders are frozen.

I.3 How the Recorded Evidence Constrains These Explanations

Both interaction operators approach uniform clean gates late in training, so a mechanism specific to multiplication cannot alone explain this shared trajectory. Validation answer loss decreases during stage 2 while gates approach uniformity; these records do not isolate the effects of direct prefix paths, global weighting, or regularization.

The balance objective permits selective per-example gates: whenever $\bar{g}_e = 1/E$, $\mathcal{L}_{\mathrm{bal}} = 0$, even if individual examples favor different experts. The squared log-sum-exp penalty is not an entropy penalty either: a common logit shift can set log-sum-exp to zero without changing softmax probabilities. Neither observation establishes the regularizers' empirical effects. Noisy-gate variation also includes the sampled perturbations.

The prediction analyses in Appendix D constrain output-level interpretations. Score differences between the recorded family runs can conceal opposing correctness transitions, and CV-Bench-3D accuracy accompanies strong answer-option imbalance. The family-run source versions differ, so their changes do not isolate the intended intervention. Neither observation identifies its internal cause. The opposite B-L versus S-L spatial-relation ordering on CV-Bench-2D and SEEDBench further limits any general claim that bilinear fusion harms relational reasoning.

I.4 Future Controls for Distinguishing Hypotheses

The following controls remain unperformed. At a fixed checkpoint, uniform, calibration-mean, and shuffled gates would distinguish average preferences from example-specific weighting. Prefix interventions that preserve sequence length would test reliance on each input stream. Expert outputs and the gradient projections in Equation 16 would probe feature diversity, answer-loss relevance, and cancellation across patches. Answer permutations and image interventions would test the CV-Bench-3D imbalance. Scale-matched operator replacements restricted to router inputs, and capacity-matched tests retaining the token–patch axis or using patchwise gates, would examine the representation restrictions. These interventions require distribution-shift checks and repeated training where applicable.