Modality-Depth Routing for Visual Reasoning in VLM Post-Training

Yiming Ren[1,2*] **Yiran Xu**[1*] **Chufan Shi**[1] **Yu Qiao**[2] **Junjie Wang**[1†] **Yujiu Yang**[1†]
[1]Tsinghua University [2]Shanghai Artificial Intelligence Laboratory

Abstract

The prevailing route to stronger visual reasoning in vision-language models (VLMs) builds costly Chain-of-Thought corpora or curated step-by-step rationales, concentrating progress in well-resourced industry labs. We ask whether the reasoning signal already present in generic visual instruction data can instead be unlocked through a structural change to the post-training interface itself. At fixed data and capacity, standard supervised fine-tuning (SFT) on a generic instruction mixture partially improves reasoning but homogenizes the modality-depth pathways through which reasoning and perception inputs flow. We isolate two coupled patterns. First, visual and textual representations couple at the deepest layer: cross-modal Centered Kernel Alignment (CKA) rises by roughly 12% over the frozen baseline. Second, perception and reasoning inputs are routed through similar late-depth profiles, leaving no mechanism for per-input depth allocation. We propose **M**odality-**D**epth **A**ttention (MDA), a lightweight side pathway that adds modality-specific query projections and a sigmoid-gated learnable depth residual. Under a matched protocol on Qwen3-VL-2B, MDA improves the reasoning average over SFT by $+3.9$ and keeps perception almost intact, at 0.05% extra parameters. Multiple mechanistic probes converge on the same routing account. The gains extend to Qwen3-VL-8B and transfer to LLaVA-OneVision-7B, suggesting that modality-depth routing may be a general lever for unlocking reasoning from generic instruction data without compromising perception. Code will be released upon publication.

1 Introduction

Vision-language models (VLMs) [1–4] are typically adapted through visual instruction tuning. On generic instruction mixtures that span many domains, this yields only modest reasoning gains on visual reasoning benchmarks [5–7], while the model routes reasoning and perception inputs through increasingly similar modality-depth profiles, preventing either regime from using task-adapted representations. Two factors are commonly invoked to close this gap. The first is *adding data*: generic image-text mixtures lack step-by-step supervision, motivating reasoning-oriented data pipelines [8–10]. The second is *adding capacity*: the backbone or its adapter may be too small to absorb stronger supervision, motivating larger models or expert-style adapters with higher trainable-parameter budgets [11].

A third factor receives less attention. At fixed data and capacity, a pre-trained VLM already carries routing flexibility across depth and modality that standard supervised fine-tuning (SFT) does not preserve. The reasoning signal in generic visual instruction data is not absent; it is present but stranded, because the post-training interface fails to deliver it to the pathways that could use it. The bottleneck is structural: it lies in how fine-tuning maps perception and reasoning inputs onto

*Equal contribution.
†Corresponding authors.

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

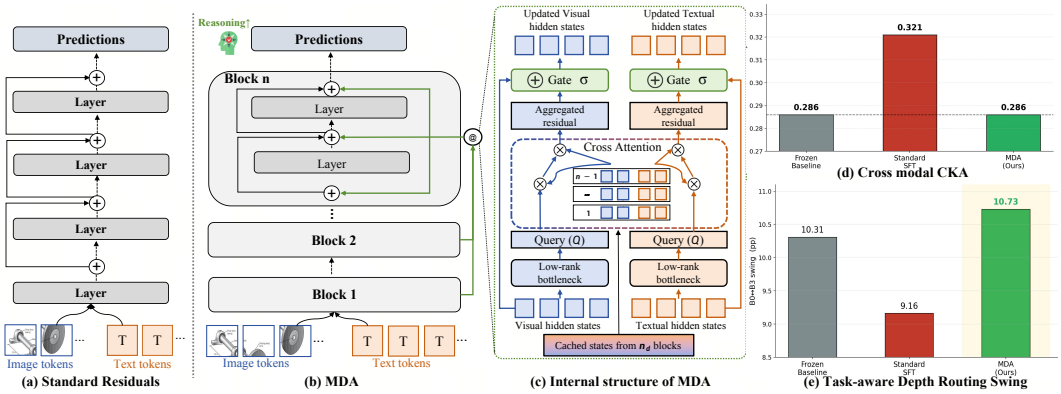

Figure 1: Modality-depth homogenization and our routing repair. **(a)** Standard SFT shares a residual stream across modalities. **(b)** MDA groups layers into blocks and adds modality-specific, Sigmoid-gated cross-block attention. **(c)** Internal structure of an MDA block: modality-specific low-rank queries attend over cached block states and inject a gated depth residual. **(d)** CKA shows MDA reverses SFT's modality coupling. **(e)** Routing swing suggests task-aware depth allocation is restored.

the modality-depth structure of a multimodal transformer [12–14]. On reasoning tasks, the visual modality often supplies information that the textual modality cannot recover (e.g., the structure of a chart or fine-grained spatial relations), so the two modalities should remain distinguishable rather than collapse onto each other; high-level inferences are then composed in the deeper blocks of the backbone. On perception tasks, textual routing is more balanced across depth, retaining lexical details from shallow blocks alongside high-level context from deep ones.

Standard post-training does not preserve this distinction. It couples visual and textual representations at the deepest layer (cross-modal Centered Kernel Alignment, CKA, rises sharply above the frozen baseline) and routes inputs from different regimes through similar late-depth profiles. We call this modality-depth homogenization: the modality axis flattens, and the depth axis rigidifies. As a result, the implicit reasoning supervision present in generic instruction data is not fully exploited. On the modality side, once visual and textual streams collapse onto each other in deep layers, the two evidence sources cannot be kept separable enough to support multimodal reasoning, which requires combining heterogeneous signals while keeping them distinguishable. On the depth side, the model cannot meaningfully tell reasoning inputs apart from perception inputs at the routing level, so neither regime receives a depth-allocation pattern tailored to its computational demand, and the reasoning signal carried by the data is absorbed only partially.

Both failures point to the interface, not capacity. We introduce MDA (Sec. 2), a small side pathway that runs alongside the backbone and restores both axes. Visual and textual tokens get separate query projections, so the two modalities are aggregated through independent routing queries rather than competing through a single shared one. A learnable Sigmoid gate, initialized near zero, controls how much aggregated cross-block information enters the residual stream, keeping the side pathway dormant early in training and ramping up only after the main optimization stabilizes.

Empirically, on Qwen3-VL-2B (Sec. 3), MDA lifts the reasoning benchmarks average (over MMMU, MathVista, MMStar) by +3.9 over standard post-training, including +6.9 on MathVista, while also keeping the perception benchmarks (over POPE, AI2D, TextVQA) average essentially intact (+0.4 over standard SFT, within 0.6 of the frozen baseline), at only 0.05% extra parameters. Among the configurations we evaluate, this is the strongest joint improvement: standard SFT itself loses 1.0 on perception, and most baselines pay for reasoning gains with perception loss or vice versa; only LoRA-MoE achieves a positive perception delta over SFT, but with a much larger parameter budget and a smaller reasoning gain (+2.7 vs. +3.9). The same design trend extends to Qwen3-VL-8B and transfers to LLaVA-OneVision-7B.

The benchmark results are matched by multiple mechanistic probes (Sec. 4). We probe representational similarity, optimization dynamics, linear recoverability, and depth allocation. Cross-modal centered kernel alignment (CKA) at the deepest layer rises under SFT and returns to the frozen baseline under MDA. A modality-specific update rate rises from 46% under SFT to 98% under

MDA. Linear probing of reasoning concepts at the deepest layer recovers 4 points of separability. And a parameter-free depth-contribution probe shows reasoning inputs routing more heavily to deep blocks under MDA while perception maintains a balanced shallow-deep allocation. All four signatures shift in the same direction: standard SFT compresses the modality and depth axes, MDA restores them.

We highlight our contributions as follows:

- A diagnostic frame. We identify a structural pathology of standard post-training on generic data, *modality-depth homogenization*, and isolate it with four complementary measurements spanning cross-modal representation, optimization, task-relevant separability, and depth allocation.
- A regime-adaptive routing module. MDA addresses both axes with modality-specific query projections and a Sigmoid-gated depth residual, at 0.05% extra parameters.
- Empirical and mechanistic evidence. MDA delivers the largest reasoning improvement over standard post-training among the configurations we evaluate, while keeping perception essentially intact; mechanistic probes converge on the routing account, and the effect extends to Qwen3-VL-8B and LLaVA-OneVision-7B.

2 Method: Modality-Depth Attention (MDA)

2.1 Shared Residual Adaptation and Modality-Depth Homogenization

Let $\{H_\ell\}_{\ell=1}^{L}$ denote the per-layer hidden states of an L-layer multimodal transformer backbone, with residual update $H_\ell = H_{\ell-1} + \Delta H_\ell$ at layer ℓ. We group the backbone into B contiguous depth blocks of size K, and denote the corresponding block-level hidden states by $\{H_b\}_{b=1}^{B}$. Under the residual-stream view of transformers [12–14], the final representation is obtained by accumulating residual updates across successive layers or blocks. Standard visual instruction tuning adapts this stream through a shared interface, where visual tokens, textual tokens, and different input regimes are optimized through the same backbone pathway.

This shared interface induces two structural limitations for VLM post-training. The first limitation is *modality homogenization*: visual and textual tokens are adapted through a common residual pathway, even though they carry different forms of evidence and may require different update directions during post-training [15, 16]. This coupling is useful for late semantic alignment, but it can be restrictive for visual reasoning, where visual evidence and textual operators should remain distinguishable before being composed for prediction. A routing interface for post-training should therefore allow the two modalities to retrieve depth information through separate routes.

The second limitation is *depth homogenization*: the standard residual interface provides no explicit input-conditioned mechanism for selecting which depth blocks should contribute most strongly to the current prediction. Reasoning-oriented inputs particularly benefit from deeper-block processing, where the model concentrates textual evidence in later blocks for symbolic composition and multi-step inference, whereas perception-oriented inputs are less sensitive to depth allocation. A shared depth profile across input regimes is therefore insufficient: it averages out the deep-block emphasis that reasoning specifically needs, leaving the model unable to allocate extra capacity to inputs whose computational demand differs from the average.

These limitations motivate three design requirements. (1) the routing should be *modality-specific*, so that visual and textual tokens are not forced to share a single query over depth. (2) it should be *input-adaptive*, so that different examples can induce different block allocations. (3) it should be *conservatively gated*, so that the auxiliary pathway starts close to the pretrained residual stream and does not destabilize early post-training. The next subsection instantiates these requirements as MDA, a lightweight side pathway with modality-specific queries and a sigmoid-gated cross-block residual. These requirements turn the diagnosis of modality–depth homogenization into an architectural constraint: the post-training interface must separate modality routes, adapt depth allocation to the input, and preserve the pretrained residual stream during early optimization.

2.2 MDA: Modality-Specific Gated Cross-Block Routing

MDA is a lightweight side pathway inserted at the block boundaries of the backbone. It implements the routing requirements in Sec. 2.1 by combining modality-specific queries, input-conditioned

cross-block attention, and gated residual injection. The main transformer stream is not structurally modified; the side pathway only determines how information from previous depth blocks is reintroduced into the current hidden state.

Setup. Let d be the hidden size of the backbone, L the number of transformer layers, and K the block size. We partition the backbone into $B = L/K$ contiguous depth blocks. At the end of block $b \in \{1, \dots, B\}$, the backbone produces hidden states $H_b \in \mathbb{R}^{n \times T \times d}$, where n is the batch size and T is the sequence length. Visual and textual token positions are identified by a binary modality mask $M \in \{0,1\}^{n \times T}$, where $M_{i,t} = 1$ indicates that token t of example i is visual. When applied to hidden states, M is broadcast along the feature dimension. At block boundary b, MDA reads the current block output H_b and the available block outputs $\{H_1, \dots, H_b\}$.

Modality-specific query construction. For each modality $m \in \{\mathrm{vis}, \mathrm{txt}\}$, MDA first summarizes the current hidden state into an example-level modality context. The visual and textual contexts are obtained by masked mean pooling, $c_i^m = \sum_t M_{i,t}^{(m)} H_b[i,t,:] / (\sum_t M_{i,t}^{(m)} + \epsilon)$, where $M^{(\mathrm{vis})} = M$, $M^{(\mathrm{txt})} = 1 - M$, and ϵ is a small constant guarding against degenerate masks. Each context is then mapped to a depth-query by an independent bottleneck projection $q_i^m = W_q^m c_i^m$, where $W_q^m = W_{2,q}^m W_{1,q}^m$ with $W_{1,q}^m \in \mathbb{R}^{r_b \times d}$ and $W_{2,q}^m \in \mathbb{R}^{d \times r_b}$. The projections W_q^{vis} and W_q^{txt} are not shared. This separation allows visual and textual tokens to induce different depth-retrieval patterns instead of competing through a single query over blocks. We use example-level modality contexts rather than token-level queries, so that routing reflects the input regime while avoiding unstable token-wise depth selection.

Cross-block attention over depth. Given a modality-specific query q_i^m, MDA attends over previous block outputs along the block dimension. The keys are RMS-normalized block states, while the values are the original unnormalized block states. For token t of example i, the score assigned to block $b' \le b$ is $s_{i,t,b'}^m = \langle q_i^m, \mathrm{RMSNorm}(H_{b'})[i,t,:] \rangle$, and the depth weights $\pi_{i,t,b'}^m = \mathrm{softmax}_{b'}(s_{i,t,b'}^m)$ are normalized over the b available block indices. The modality-specific cross-block aggregate is then

$$A^m[i,t,:] = b \cdot \sum_{b'=1}^{b} \pi_{i,t,b'}^m H_{b'}[i,t,:]. \tag{1}$$

Gated residual injection. The cross-block aggregate is not directly written into the backbone stream. Instead, a per-block and per-modality scalar gate interpolates between the current hidden state and the modality-specific aggregate:

$$H_b^{m,\mathrm{out}} = \big(1 - \sigma(\alpha_b^m)\big) H_b + \sigma(\alpha_b^m) A^m, \qquad m \in \{\mathrm{vis}, \mathrm{txt}\}. \tag{2}$$

The visual and textual outputs are then spliced back via the modality mask, $H_{b+1} = M \odot H_b^{\mathrm{vis,out}} + (1 - M) \odot H_b^{\mathrm{txt,out}}$, preserving the backbone interface so that the next transformer block receives a standard hidden-state tensor in $\mathbb{R}^{n \times T \times d}$.

Trainable components. The computation above introduces only two types of trainable components. The first is the pair of modality-specific query projections W_q^{vis} and W_q^{txt}. The second is the set of scalar gates $\{\alpha_b^{\mathrm{vis}}, \alpha_b^{\mathrm{txt}}\}_{b=1}^B$. No learned key or value projection is introduced; the block outputs themselves serve as keys and values for cross-block routing. Thus, the added parameters are concentrated on routing rather than on feature transformation.

2.3 Design Properties

MDA is designed as a conservative routing interface rather than a new feature-transformation module. We initialize every gate as $\alpha_b^m = -2$, so that $\sigma(\alpha_b^m) \approx 0.12$ at the start of post-training. By Equation (2), the side pathway is weakly active at initialization, while the pretrained residual stream dominates. This avoids introducing an always-on cross-depth path before the post-training dynamics have stabilized, while still allowing gradients to update the routing pathway from the beginning.

The parameter overhead is small. For Qwen3-VL-2B [3], we use $d = 2048$, $L = 28$, $K = 7$, $B = 4$, and $r_b = 128$. The two modality-specific query projections and the $2B$ scalar gates add about 1.05M parameters, corresponding to approximately 0.05% of the 2B backbone. Because MDA introduces

Table 1: Main results on Qwen3-VL-2B (matched evaluation protocol). Reasoning slice = {MMMU, MathVista, MMStar}; Perception slice = {POPE, AI2D, TextVQA}. MDA achieves the largest improvement of reasoning and perception over standard post-training among the methods we evaluate.

Method	R-Avg (3-bench)	MMMU	MathVista	MMStar	P-Avg (3-bench)	POPE	AI2D	TextVQA
Frozen Baseline	47.3	42.2	49.1	50.7	80.9	88.4	75.4	78.8
Standard SFT	49.2	41.6	54.4	51.5	79.9	87.4	75.1	77.3
Weight Decay	50.6	43.1	56.7	51.9	79.0	86.3	73.6	77.1
rsLoRA [18]	47.5	42.0	50.9	49.5	79.3	88.2	73.2	76.6
PiSSA [19]	41.9	37.4	40.6	47.6	77.0	88.8	66.4	75.7
LoRA-MoE [11]	51.9	43.9	58.3	53.6	80.1	87.5	75.3	77.6
Attention Residual [17]	47.6	44.0	50.7	48.2	76.3	84.7	69.1	75.2
MDA (Ours)	53.1	44.6	61.3	53.3	80.3	88.3	75.2	77.3

no learned key or value projections, the added capacity is concentrated on selecting routes over existing block states rather than constructing a parallel adapter.

MDA is closest to attention residuals, which provide cross-block residual access through depth-wise aggregation [17]. The difference is that MDA makes this access modality-specific and post-training-compatible. Prior attention-residual designs use a shared query over depth and an effectively always-on residual path. MDA instead separates the depth query by modality and gates the residual contribution during adaptation. These two changes help adapt generic cross-block aggregation into a modality-aware routing interface for VLM post-training.

3 Experiments

3.1 Experimental Setup

Our primary controlled study post-trains Qwen3-VL-2B [3] on a generic, open-source visual instruction mixture (LLaVA-instruct-mix-VSFT, 259K samples), without reasoning-specific filtering, additional reasoning data, or curated chain-of-thought rationales. All methods share the same data, optimizer, batch size, learning-rate schedule, number of epochs, decoding protocol, and answer-extraction rules. At evaluation time we disable chain-of-thought prompting; each model produces a direct short answer to objective-format questions, so that differences reflect the post-training interface rather than response-length or prompting effects. We evaluate on two slices: reasoning (MMMU, MathVista, MMStar) and perception (POPE, AI2D, TextVQA). Benchmark-specific item filters and full protocol details are given in Sec. B.2.

Run-to-run stability. Main benchmark numbers are reported as single-run point estimates, which is common in compute-intensive large-scale VLM evaluations [3, 4].

3.2 Main Results

Table 1 reports the head-to-head comparison on six benchmarks chosen to probe both reasoning and perception. The headline observation is that MDA substantially improves reasoning while keeping perception essentially intact; among the configurations we evaluate, no other method achieves a comparable reasoning gain without giving up perception. Standard SFT improves Reasoning from 47.3 to 49.2 but reduces Perception from 80.9 to 79.9, consistent with the reasoning–perception trade-off reported in prior work on low-rank post-training. Stronger regularization (Weight Decay) and the variants rsLoRA and PiSSA shift this trade-off but do not remove it. LoRA-MoE, the capacity-adding baseline with a larger trainable-parameter budget, reaches a higher Reasoning average (51.9) but still falls 1.2 points short of MDA, and its Perception (80.1) only marginally exceeds Standard SFT and remains below the Frozen Baseline; simply scaling adapter capacity does not fully close the gap. The most informative baseline is the un-gated Attention Residual: although it provides cross-depth access, it performs worse than Standard SFT on Reasoning (47.6) and substantially worse on Perception (76.3). This pattern is consistent with the view that cross-depth aggregation alone is not

Table 2: MDA ablations on Qwen3-VL-2B on the same six benchmarks as Table 1. Removing the modality split or the input-adaptive query each costs a non-trivial Reasoning point. The final two rows probe the inner bottleneck of the modality-specific query projection.

Variant	**R-Avg**	MMMU	MathVista	MMStar	**P-Avg**
MDA (Default)	53.1	44.6	61.3	53.3	80.3
Shared query (no modality split)	50.9	46.5	54.6	51.5	79.9
Fixed query (input-independent)	50.4	45.2	54.8	51.3	80.2
Token-level routing (no block structure)	50.4	44.7	54.6	51.9	80.0
Inner bottleneck r_b=4 (under-capacity)	50.4	45.6	54.6	50.9	78.3
Inner bottleneck r_b=256 (over-capacity)	51.7	43.9	58.2	52.9	79.9

sufficient, and that the side pathway must be conservative at initialization and modality-specific during routing. MDA satisfies both conditions and achieves the largest joint improvement we observe: $+3.9$ Reasoning and $+0.4$ Perception over Standard SFT under a matched trainable-parameter budget. The only other configuration with a positive perception delta over SFT is LoRA-MoE ($+2.7$, $+0.2$), but with a substantially larger trainable-parameter budget and a smaller reasoning gain. We defer the mechanistic interpretation to Sec. 4.

3.3 Ablation Study

We ablate MDA on the same six-benchmark slice (Table 2), stripping one design component at a time. Removing the modality split costs 2.2 Reasoning points, and degrading the input-adaptive query to a fixed query costs 2.7. The last two rows ablate the *inner bottleneck* of the modality-specific query projection (the dimension r_b in W_q^m). Both extremes hurt: r_b=4 under-parameterizes the routing query and r_b=256 over-parameterizes it, leaving r_b=128 as the working point we use throughout.

4 Mechanistic Analysis

We complement the empirical results with four mechanistic probes, organized around the two pathologies of Sec. 2. The probes are deliberately diverse so that they examine the hypothesized mechanism from independent angles. Three are quantitative population-level measurements: cross-modal CKA at the deepest layer (Sec. 4.1.1), a modality-specific update rate during training (Sec. 4.1.2), and linear concept probing (Sec. 4.3). The fourth is a qualitative per-input visualization of how the trained models route concrete examples through different depths (Sec. 4.2). The probes are not designed to isolate a single causal pathway; rather, they test whether independent signatures predicted by the modality-depth homogenization hypothesis move coherently across model variants. Across representation similarity, optimization direction, linear recoverability, and residual-depth allocation, we observe a coherent pattern across all four probes, suggesting that standard SFT reduces modality-depth differentiation while MDA restores it. Taken together, the probes provide converging evidence that the gains of MDA are consistent with reduced modality coupling and more adaptive depth allocation, rather than with incidental capacity differences.

4.1 Reversing Deep Modality Collapse

4.1.1 Cross-Modal CKA at the Deepest Layer

We compute Centered Kernel Alignment between visual-token and textual-token hidden states at every transformer layer of Qwen3-VL-2B. A high CKA at a given layer signals representational coupling, a low value signals separation. Figure 1(d) summarizes the result at the deepest layer (layer 27): the frozen baseline sits at CKA=0.286, standard post-training raises this to 0.321 ($+0.035$ absolute, 12% relative), consistent with the modality homogenization identified in Sec. 2, and MDA returns the deepest-layer CKA to 0.286, matching the frozen-baseline level of modality separation. This direction is consistent with the hypothesis that the side pathway mitigates the modality coupling that would otherwise affect the main residual stream.

4.1.2 Modality-Specific Update Rate

CKA captures modality separation at the level of resulting representations. A complementary, optimization-level view asks whether the post-training updates themselves keep the two modalities on distinct directions. At each training step we run a single forward and backward pass and read two gradients: g_{txt} at the text-embedding output and g_{vis} at the visual-encoder output (both from the same forward pass; only the entry point differs). We compute the cosine similarity $\cos(g_{\text{vis}}, g_{\text{txt}})$ and report the fraction of sampled steps on which this cosine is negative. We call this fraction the *modality-specific update rate*: a high rate indicates that the optimizer is moving the visual-side and the textual-side gradients in opposing directions on most steps, while a rate near 50% indicates that the two gradients are not aligned in either direction. Full implementation details are given in Appendix B.3.

Standard SFT yields a modality-specific update rate of 46%, indicating that the visual and textual gradient signals are not reliably aligned in either direction across training. Under MDA the rate rises to 98% (Table 3), indicating that the two modalities receive opposing gradient directions on nearly every training step. This is consistent with the representational separation observed by CKA, and is consistent with the side pathway providing the optimizer a structural mechanism for keeping modality-specific contributions more distinct during adaptation.

4.2 Adaptive Depth Allocation

The depth homogenization problem predicts that standard post-training should force every input to draw its final-layer evidence from the same depth profile, while MDA should restore a regime-conditional depth-allocation pattern. We test this prediction with a parameter-free per-sample probe and a single population-level summary statistic.

Probe construction. Following the residual-stream analysis in mechanistic interpretability [13], any transformer's final hidden state can be written as

$$X_{\text{final}} = X_{\text{embed}} + \Delta X_{B_0} + \Delta X_{B_1} + \Delta X_{B_2} + \Delta X_{B_3},$$

where ΔX_{B_i} is the cumulative residual update produced by block i. We probe each block's *actual information-update contribution* via the normalized L2 norm of its residual output,

$$\text{Contribution}(B_i) = \frac{\|\Delta X_{B_i}\|_2}{\sum_j \|\Delta X_{B_j}\|_2}.$$

This probe reads no model parameter; it measures the actual energy each block deposits into the residual stream, a quantity defined for any transformer regardless of whether it carries an explicit depth-gating parameter. The Frozen Baseline, Standard SFT, and MDA can therefore be placed on exactly the same footing.

Routing swing. For each input x, we summarize its textual-channel depth allocation by the *routing swing*, defined as $\text{swing}(x) = \text{Contribution}(B_3 \mid x) - \text{Contribution}(B_0 \mid x)$, which measures the net shift of textual routing from the shallowest block to the deepest block.A large positive swing indicates that the input concentrates its textual evidence in deep blocks (semantic processing at depth); a moderate swing indicates a more balanced shallow-deep allocation. At the population level, we measure the *task-aware swing*

$$\Delta_{\text{task}} = \overline{\text{swing}}_R - \overline{\text{swing}}_P,$$

the gap between the mean reasoning swing and the mean perception swing: a large gap means the model allocates different depth profiles to the two regimes, while a small gap means inputs share a task-agnostic profile. We compute both quantities on 1500 samples (250 per benchmark $\times$ 6 main-table benchmarks).

Per-sample view (Figure 2, left). Under Standard SFT the dot clouds are relatively more similar across the six benchmarks: all means fall in a narrow 9–23 pp band, and the shape of the reasoning clouds is relatively less distinguishable from the perception clouds. This is the signature of a regime-agnostic late-depth profile. Under MDA, the picture reorganizes along the task axis: the three reasoning benchmarks produce concentrated warm clouds around 30–37 pp, deeper than the perception clouds around 21–23 pp. The Frozen Baseline sits between the two extremes with a weaker but visible reasoning-vs-perception separation. The per-sample view therefore reveals two effects

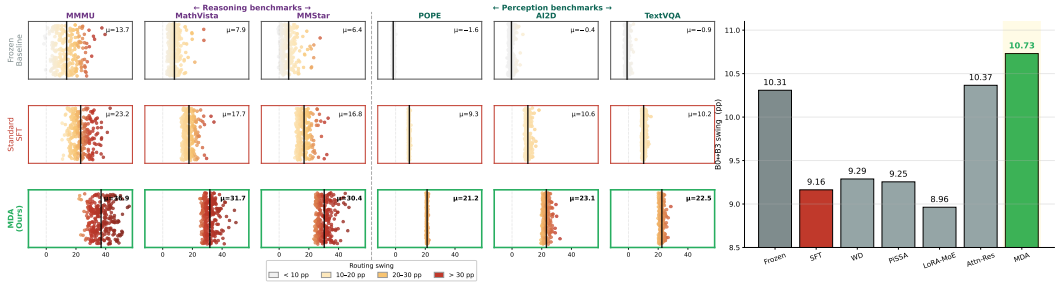

Figure 2: Adaptive depth allocation across reasoning and perception benchmarks. **(Left)** Per-sample routing swing ($B_3 - B_0$, pp). Each dot is one sample; black line = mean; green border = MDA. MDA shows large swings on reasoning inputs and moderate swings on perception; SFT is uniformly pale across both task types. **(Right)** Population-level task-aware swing Δ_{task}. MDA achieves the largest value, a post-training configuration that clearly exceeds the Frozen Baseline.

Table 3: Modality-specific update rate. Fraction of training steps on which at the visual-encoder output and at the text-embedding output point in opposing directions.

Condition	Update Rate
Standard SFT	46%
MDA (Ours)	**98%**

Table 4: Linear probing accuracy at layer 27 on a held-out reasoning benchmark. A linear classifier trained on the deepest hidden states to predict the correct multiple-choice answer.

Condition	LP Acc.
Standard SFT	52.3%
MDA (Ours)	**56.3%**

simultaneously: MDA pushes reasoning inputs toward deep-block routing (a semantics-at-depth regime), and it keeps perception inputs on a more balanced shallow-deep allocation, consistent with perception needing to retain early-block lexical cues alongside high-level context from deep blocks.

Aggregate task-aware swing (Figure 2, right). The population-level Δ_{task} sharpens the ranking. MDA attains the largest task-aware swing (10.73 pp), above the un-gated Attention Residual (10.37 pp) and the Frozen Baseline (10.31 pp). In contrast, Standard SFT (9.16 pp), Weight Decay (9.29 pp), PiSSA (9.25 pp), and LoRA-MoE (8.96 pp) all fall below the Frozen Baseline, suggesting that standard post-training may compress the backbone's natural task-aware depth routing rather than improving it. MDA most clearly restores and amplifies task-aware routing beyond the pre-trained level among the post-training configurations we evaluate.

Interpretation. Taken together, the two panels place reasoning and perception on opposite sides of an input-conditional depth-allocation axis that MDA appears to maintain. Reasoning inputs concentrate in deep blocks, consistent with a regime in which the model integrates high-level semantic evidence from both modalities. Perception inputs, in contrast, retain a more balanced shallow-deep allocation in the textual channel: low-level lexical cues need to be preserved in early blocks and then fused with deep context at the output. Standard post-training tends to collapse both regimes onto a single compressed profile; MDA restores the distinction, achieving the largest task-aware swing among the variants we tested.

4.3 Linear Concept Probing

Our fourth probe asks a different question. Regardless of how representations are organized internally, can a downstream classifier still recover reasoning concepts from them? For this probe, we use a held-out reasoning-oriented benchmark from our evaluation suite, freeze each post-trained model, extract hidden states at the final pre-output position of layer 27, and train a linear classifier on the selected benchmark's samples to predict the correct multiple-choice answer (Table 4).

Standard post-training leaves reasoning concepts only weakly linearly recoverable from the deepest hidden states (52.3%). MDA recovers 4.0 percentage points of this separability, lifting the linear-probe accuracy to 56.3%. The direction of the change is consistent with the +3.9-point Reasoning gain in Table 1 and corroborates the view that the gains of MDA are accompanied by more linearly separable reasoning-relevant structure in deep representations.

Table 5: Extensibility of MDA across scales and architectures. Reasoning benchmarks (MMMU, MathVista, MMStar) are drawn from the main-paper slice. MDA improves reasoning over Standard SFT at both the 8B scale and on a different architecture family.

Model	Method	MMMU	MathVista	MMStar	**R-Avg**
Qwen3-VL-8B	Standard SFT	58.0	73.0	63.6	64.9
	MDA (Ours)	**58.0**	**75.6**	**64.4**	**66.0**
LLaVA-OneVision-7B	Standard SFT	46.7	65.4	52.3	54.8
	MDA (Ours)	**50.2**	65.2	**53.5**	**56.3**

4.4 Extensibility

Table 5 reports reasoning numbers of MDA under two stress tests: scaling the same architecture to Qwen3-VL-8B, and transferring to a different visual-encoder family (LLaVA-OneVision-7B). The improvement over Standard SFT is preserved in both settings, indicating that MDA is not specific to one backbone or one encoder family. R-Avg improves by $+1.1$ on Qwen3-VL-8B and $+1.5$ on LLaVA-OneVision-7B, with single-benchmark gains of $+2.6$ on MathVista at the 8B scale and $+3.5$ on MMMU on LOV-7B. The 8B improvement is smaller than the $+3.9$ observed at 2B, consistent with stronger backbones leaving less reasoning headroom for post-training to recover. The two backbones also differ in their fusion design (Qwen3-VL uses native multimodal pretraining, while LLaVA-OneVision adopts a connector-based interface), so consistent gains across both suggest that the routing repair is not tied to a specific fusion architecture.

5 Related Work

Cross-block residuals. Mechanistic-interpretability work [13, 20, 21] treats the transformer as an unweighted sum of per-block residual updates; this view underlies our per-block contribution probe in Sec. 4.2. Recent depth-aggregation residuals [17] add cross-block access but use an always-on auxiliary stream and a single shared query. Our ablations (Sec. 3) show that removing either component costs several reasoning points. MDA turns generic cross-block aggregation into a regime-adaptive interface that addresses the depth and modality axes simultaneously.

Modality collapse. A separate line of work documents how visual tokens dilute the language stream [15] and how multimodal training induces representation alignment that is sometimes detrimental [16]. We connect this thread to a deep-layer footprint via cross-modal CKA (Sec. 4.1.1) and to an optimization-level analogue via modality gradient divergence (Sec. 4.1.2). Architectural methods such as Q-Former [22] redesign fusion before the backbone; MDA instead decouples modalities *inside* the backbone via a side pathway, composable with front-end fusion modules.

Post-training trade-offs. Prior work reports that visual instruction tuning trades perception against reasoning [16], that low-rank post-training can suppress pre-trained capabilities [23], and that adapter variants [19, 18] or MoE adapters [11] mitigate the cost via alternative parameterizations. These works characterize the cost as inherent to low-rank adaptation rather than localizing its structural origin in the post-training interface itself. We give a structural diagnosis, measure it with four probes, and show that a small architectural intervention removes the trade-off rather than rebalancing it.

6 Conclusion

Complex visual reasoning is shaped not only by supervision quality or adaptation capacity, but also by how post-training routes different inputs through the pre-trained backbone. We characterize a structural pattern of standard post-training, *modality-depth homogenization*, in which visual and textual representations couple at the deepest layer and inputs from different regimes are routed through similar depth profiles. MDA, a parameter-efficient side pathway with modality-specific queries and a conservative Sigmoid-gated residual, improves reasoning by $+3.9$ over Standard SFT on Qwen3-VL-2B while keeping perception essentially intact, and four mechanistic probes provide

converging evidence that the gains are consistent with the routing interpretation. The same design trend extends to Qwen3-VL-8B and transfers to LLaVA-OneVision-7B.

Acknowledgments and Disclosure of Funding

This work was partly supported by the National Natural Science Foundation of China (Grant No. 62576191) and the Shenzhen Science and Technology Program (ZDCY20250901103533010).

References

[1] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

[2] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. LLaVA-OneVision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

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

[4] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. InternVL3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

[5] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9556–9567, 2024.

[6] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. MathVista: Evaluating mathematical reasoning of foundation models in visual contexts. In *The Twelfth International Conference on Learning Representations*, 2024.

[7] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? In *Advances in Neural Information Processing Systems*, volume 37, pages 27056–27087, 2024.

[8] Huanjin Yao, Jiaxing Huang, Wenhao Wu, Jingyi Zhang, Yibo Wang, Shunyu Liu, Yingjie Wang, Yuxin Song, Haocheng Feng, Li Shen, et al. Mulberry: Empowering MLLM with o1-like reasoning and reflection via collective monte carlo tree search. *arXiv preprint arXiv:2412.18319*, 2024.

[9] Weikang Shi, Aldrich Yu, Rongyao Fang, Houxing Ren, Ke Wang, Aojun Zhou, Changyao Tian, Xinyu Fu, Yuxuan Hu, Zimu Lu, et al. MathCanvas: Intrinsic visual chain-of-thought for multimodal mathematical reasoning. *arXiv preprint arXiv:2510.14958*, 2025.

[10] Renrui Zhang, Xinyu Wei, Dongzhi Jiang, Ziyu Guo, Shicheng Li, Yichi Zhang, Chengzhuo Tong, Jiaming Liu, Aojun Zhou, Bin Wei, et al. MAVIS: Mathematical visual instruction tuning with an automatic data engine. *arXiv preprint arXiv:2407.08739*, 2024.

[11] Xun Wu, Shaohan Huang, and Furu Wei. Mixture of LoRA experts. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

[12] Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, 2019.

[13] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. https://transformer-circuits.pub/2021/mathematical-framework/index.html.

[14] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, 2021.

[15] Zeliang Zhang, Phu Pham, Wentian Zhao, Kun Wan, Yu-Jhe Li, Jianing Zhou, Daniel Miranda, Ajinkya Kale, and Chenliang Xu. Treat visual tokens as text? but your MLLM only needs fewer efforts to see. *arXiv preprint arXiv:2410.06169*, 2024.

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

[17] Kimi Team, Guangyu Chen, Yu Zhang, Jianlin Su, Weixin Xu, Siyuan Pan, Yaoyu Wang, Yucheng Wang, Guanduo Chen, Bohong Yin, Yutian Chen, Junjie Yan, Ming Wei, Y. Zhang, Fanqing Meng, Chao Hong, Xiaotong Xie, Shaowei Liu, Enzhe Lu, Yunpeng Tai, Yanru Chen, Xin Men, Haiqing Guo, Y. Charles, Haoyu Lu, Lin Sui, Jinguo Zhu, Zaida Zhou, Weiran He, Weixiao Huang, Xinran Xu, Yuzhi Wang, Guokun Lai, Yulun Du, Yuxin Wu, Zhilin Yang, and Xinyu Zhou. Attention residuals. *arXiv preprint arXiv:2603.15031*, 2026.

[18] Damjan Kalajdzievski. A rank stabilization scaling factor for fine-tuning with LoRA. *arXiv preprint arXiv:2312.03732*, 2023.

[19] Fanxu Meng, Zhaohui Wang, and Muhan Zhang. PiSSA: Principal singular values and singular vectors adaptation of large language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

[20] nostalgebraist. Interpreting GPT: the logit lens. LessWrong, 2020. URL https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens.

[21] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads. *Transformer Circuits Thread*, 2022. https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

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

[23] Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, Cody Blakeney, and John P. Cunningham. LoRA learns less and forgets less. *arXiv preprint arXiv:2405.09673*, 2024.

[24] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3519–3529. PMLR, 2019.

A Limitations and Scope

The current study evaluates MDA on two transformer families (Qwen3-VL and LLaVA-OneVision) at the 2B and 7–8B scales, trained on a single generic instruction mixture. While the consistent gains across these settings suggest the approach generalizes, scaling to larger backbones and more diverse training distributions remains an open question. The mechanistic probes are designed to characterize observable signatures of modality-depth routing rather than to intervene on them; extending the analysis with controlled ablations at the representation level would further strengthen the structural account. The current parameter budget (0.05% extra parameters) reflects a deliberate design choice for low-overhead post-training; higher-capacity routing variants may offer additional headroom that we have not yet explored. Finally, MDA is evaluated in an English-dominant regime; how task-conditional routing behaves under multilingual or code-heavy instruction mixtures is left for future investigation. MDA operates inside the backbone and does not alter the vision–language fusion front end, so it is in principle composable with front-end fusion modules such as Q-Former [22]; we have not empirically evaluated this combination.

B Implementation and Evaluation Details

This appendix documents the training configuration, evaluation protocol, and mechanistic probes used throughout the main paper. All information is shared across methods unless stated otherwise; no baseline receives special treatment.

B.1 Training Hyperparameters

Table 6 reports the full training configuration shared by every method in the main paper. The base models (Qwen3-VL-2B and Qwen3-VL-8B [3], and LLaVA-OneVision-7B [2]) are publicly released. All LoRA-style baselines and MDA use the same optimizer, schedule, batch size, and number of epochs. Method-specific choices are limited to (i) the trainable-parameter budget of each PEFT variant and (ii) the inner bottleneck r_b of MDA, which is internal to our side pathway and unrelated to any LoRA rank.

Fixed design decisions. The generic visual instruction mixture (LLaVA-instruct-mix-VSFT, 259K samples) is used without any reasoning-oriented filtering or chain-of-thought rationale supplementation. No method uses a custom learning-rate schedule, and no method uses warm-restart. Seeds are fixed to the same value across methods. MDA uses LoRA (rank r=64, α=128, dropout 0.05) applied to both the attention projection matrices (q, k, v, o) and the MLP projections (gate, up, down) as its primary adaptation interface, with the modality-depth routing side pathway added on top. The trainable-parameter budget is otherwise matched between LoRA-style PEFT baselines and MDA's own LoRA component; MDA adds only the modality-specific projections and the gate, approximately 1.05M additional parameters (about 0.05% of the 2B backbone). The pre-trained transformer weights are left unchanged in all experiments.

B.2 Evaluation Protocol

Table 7 summarizes the decoding and answer-extraction choices used for every benchmark reported in the main paper. A single protocol is applied across all methods.

Benchmark-specific filters. For MMMU we evaluate only the single-image subset and skip items whose answers are open-ended, because multi-image handling and open-ended scoring depend on instruction-following behavior that varies across methods independently of visual reasoning quality. For MathVista we apply the same open-ended filter. For the remaining benchmarks (POPE, AI2D, MMStar, TextVQA) we evaluate on the full public release of each benchmark, using the task-specific prompts and extraction/scoring rules documented in the paragraphs below and in Fig. 3; our rules deviate from the official leaderboards for POPE (substring match) and TextVQA (soft score without punctuation/article normalization), so the corresponding numbers in Table 1 are not directly comparable to the official versions.

Rationale for the open-ended filter. The filter is a deliberate match between the training distribution and the evaluation format, not a benchmark-difficulty shortcut. All post-trained methods in this paper (Standard SFT, all PEFT baselines, and MDA) are trained on the same generic visual

Table 6: **Training hyperparameters** shared across methods for all three backbones. Gradient checkpointing is enabled only for the 8B and LLaVA-OneVision runs. The LLaVA-OneVision column reports the configuration of our LOV-7B transfer run (bottleneck r_b=64 and a $200\times$ routing-LR multiplier reflecting a different scale of the side-pathway loss on the LLaVA-OneVision backbone).

Hyperparameter	Qwen3-VL-2B	Qwen3-VL-8B	LLaVA-OneVision-7B
Training data	LLaVA-instruct -mix-VSFT	LLaVA-instruct -mix-VSFT	LLaVA-instruct -mix-VSFT
Number of samples	259K	259K	259K
Epochs	1	1	1
Optimizer	AdamW	AdamW	AdamW
Learning rate	5×10^{-6}	5×10^{-6}	2×10^{-5}
(β_1, β_2)	$(0.9, 0.999)$	$(0.9, 0.999)$	$(0.9, 0.999)$
Weight decay	0.0 except WD: 0.1	0.0	0.0
LR schedule	cosine, linear warm-up	cosine, linear warm-up	cosine, linear warm-up
Warm-up ratio	0.03	0.03	0.03
Per-device batch size	16	2	2
Gradient accumulation	1	8	8
Effective batch size	16	16	16
Max sequence length	2048	2048	2048
Precision	bf16	bf16	bf16
Gradient checkpointing	disabled	enabled	enabled
LoRA rank r	64	32	64
LoRA α	$2r$	$2r$	$2r$
LoRA dropout	0.05	0.05	0.05
LoRA target modules	q, k, v, o + MLP	q, k, v, o + MLP	q, k, v, o + MLP
MDA bottleneck r_b	128	128	64
MDA block size K	7	9	7
MDA gate init α_b^m	-2	-2	0
MDA routing-LR mult.	$50\times$	$50\times$	$200\times$
Hardware	$1\times$ NVIDIA H200 140 GB	$1\times$ NVIDIA H200 140 GB	$1\times$ NVIDIA H200 140 GB
Approx. wall-clock	$\sim$ 4.5 h/run	$\sim$ 28 h/run	$\sim$ 35 h/run

Table 7: **Evaluation hyperparameters** shared across methods and benchmarks. The same decoding configuration is applied to the Frozen Baseline, Standard SFT, all PEFT baselines, and MDA.

Setting	Value
Decoding	greedy (temperature = 0)
Top-k / Top-p	disabled
Beam search	disabled
Chain-of-thought suppression	`/no_think` system message on `<think>`-capable backbones
Max generated tokens (default)	128
Max generated tokens (POPE, yes/no)	32
Multi-image handling (MMMU)	single-image questions only
Open-ended filtering (MMMU, MathVista)	open-ended items skipped
Option-extraction fallback	mark incorrect (no retry, no rerank)
POPE parsing	output substring-matches "yes" $\Rightarrow$ yes, else no
TextVQA scoring	VQA-style soft score $\min(\text{count}/3, 1)$ (not official normalization)

instruction mixture (LLaVA-instruct-mix-VSFT, 259K samples), which contains short-form, single-answer supervision and no explicit chain-of-thought rationales. A model post-trained on this mixture is therefore not equipped to produce long free-form reasoning traces or multi-paragraph open-ended explanations, and scoring such generations against reference answers would probe response-format learning rather than visual reasoning. Because our central claim is about modality-depth routing — not about open-ended generation — we restrict MMMU and MathVista to items whose answer format matches the short-form supervision seen at training time. The same filter is applied identically to every method (including the Frozen Baseline, which skips post-training entirely and shares only the evaluation protocol), so it does not advantage MDA relative to any baseline.

Answer formatting. For all multiple-choice benchmarks the model is asked to produce the answer letter directly (e.g., `A`). For short-answer benchmarks the model is asked for a short phrase. We

never concatenate rationales before the final answer. The option-extraction layer first searches the generated text for an explicit answer phrase ("answer is X", "the answer is X", "correct answer X"), taking the last such match to accommodate chain-of-thought-style outputs. If no explicit phrase is found, it falls back in order to: a single-letter output, a trailing standalone letter on the last line, a leading letter at the start, the last standalone option letter elsewhere in the text, and finally a bracketed option letter. If none of these patterns match, the item is scored as incorrect.

Evaluation prompts. Figure 3 lists the prompt templates used for each benchmark class, exactly as issued by our evaluation code. Templates are shared across methods and backbones. For backbones that expose a `<think>` control token (Qwen3-VL-2B/8B), a `/no_think` *system* message is prepended to each prompt to disable chain-of-thought behavior; backbones without such a token (LLaVA-OneVision-7B) receive the user prompt alone.

Multiple-choice VQA with external options (MMMU, AI2D, MathVista multiple-choice):

```
<image>
Question: {question}
Options:
A. {option_A}
B. {option_B}
...
Please select the correct answer from the options above. Answer with the
option's letter from the given choices directly.
```

Multiple-choice VQA with inline options (MMStar; options are already embedded in the question field):

```
<image>
Question: {question}
Please select the correct answer from the options above. Answer with the
option's letter from the given choices directly.
```

Binary VQA (POPE):

```
<image>
{question}
Answer with Yes or No.
```

Short-answer VQA (TextVQA):

```
<image>
{question}
Answer the question using a single word or phrase.
```

Figure 3: **Evaluation prompt templates**, reproduced verbatim from our evaluation code. A single template family is applied to every method and backbone. On backbones that expose a `<think>` control token, a `/no_think` message is prepended as a *system* turn (not shown above); on backbones without such a token, no system turn is added.

Scoring. For the multiple-choice benchmarks (MMMU, MMStar, AI2D, and the MathVista multiple-choice subset) and for POPE, we report accuracy as the number of correct items divided by the number of evaluated items. For TextVQA we report the mean VQA-style soft score, defined per item as $\min(\text{count}(\text{pred})/3, 1)$ where pred is the generated phrase after case-folding and whitespace trimming and $\text{count}(\cdot)$ counts matching reference answers; this follows the VQA-v2 soft-score definition but omits the official punctuation and article normalization, so the reported TextVQA numbers are not directly comparable to the official TextVQA leaderboard. For POPE, the model prediction is normalized by case-insensitive substring match: outputs containing the substring "yes" are mapped to `yes`, all others (including empty strings and refusals) are mapped to `no`. All reported numbers in the main tables are shown as percentages.

B.3 Mechanistic Probe Details

Cross-Modal CKA

We use the linear CKA formulation of Kornblith et al. [24]. Given two centered feature matrices $X, Y \in \mathbb{R}^{n \times d}$ on the same n inputs,

$$\text{CKA}(X, Y) = \frac{\|X^\top Y\|_F^2}{\|X^\top X\|_F \cdot \|Y^\top Y\|_F}.$$

At each transformer layer ℓ, for each multimodal input we extract the hidden states at that layer, mean-pool over the visual-token positions to obtain a single visual feature vector, and mean-pool over the textual-token positions to obtain a single textual feature vector. Visual and textual positions are identified from the tokenizer output. To keep the matrices well-conditioned across layers, we keep the leading 64 feature dimensions per token before pooling. We collect 200 multimodal inputs from MMMU with a fixed seed, stack the per-sample visual and textual features into matrices X_ℓ and Y_ℓ, and report $\mathrm{CKA}(X_\ell, Y_\ell)$ at the deepest layer (layer 27 for Qwen3-VL-2B).

Modality-Specific Update Rate

We sample training steps from each post-trained method and, at each sampled step, compute two modality-aligned gradients on the same multimodal training example:

- g_{txt}: the gradient of the standard causal-LM loss with respect to the output of the text-embedding layer (model.embed_tokens), captured via a forward hook with `retain_grad`.
- g_{vis}: the gradient of the same loss with respect to the output of the visual encoder (model.visual or model.vision_tower), captured via a forward hook with `retain_grad`.

Both gradients come from a single forward and backward pass; only the entry point at which we read the gradient differs. We flatten the two gradient tensors, truncate them to a common length, and compute their cosine similarity $\cos(g_{\mathrm{vis}}, g_{\mathrm{txt}})$. The *modality-specific update rate* is the fraction of sampled steps on which this cosine is negative, i.e., the gradient signals at the visual and textual entry points point in opposing directions. A high rate indicates that the optimizer's updates separate the two modalities at the input boundary; a rate near 50% indicates that the two gradients are not reliably aligned in either direction.

Linear Concept Probing

We pre-specify a single reasoning-oriented benchmark from our evaluation suite as the held-out target for linear probing. For each method (Frozen Baseline, Standard SFT, and MDA), we

1. freeze the post-trained model;
2. extract the hidden state at the final pre-output token position of layer 27 on every evaluated item of the selected benchmark;
3. train a multinomial logistic-regression classifier (`LogisticRegression` with `solver=lbfgs`, `C=1.0`, `max_iter=1000`) to predict the correct multiple-choice letter from this hidden state;
4. report the mean of 5-fold cross-validated accuracy on the evaluated subset.

The same layer, hidden-state extraction procedure, and classifier configuration are used across methods, so the difference between Standard SFT and MDA reflects changes in the representation rather than changes in the classifier.

Per-Block Residual Contribution

Any transformer's final hidden state can be written as a residual stream

$$X_{\text{final}} = X_{\text{embed}} + \Delta X_{B_0} + \Delta X_{B_1} + \Delta X_{B_2} + \Delta X_{B_3},$$

where ΔX_{B_i} is the cumulative residual update produced by block i. The normalized per-block L2 norm contribution is

$$\mathrm{Contribution}(B_i) \;=\; \frac{\|\Delta X_{B_i}\|_2}{\sum_j \|\Delta X_{B_j}\|_2}.$$

For the visualization in Figure 2, we evaluate this contribution on the final pre-output token position of 1500 samples (250 per benchmark, six main-table benchmarks), separately for the visual and textual channels. The probe reads no model parameter and is well-defined for any transformer (Frozen Baseline, Standard SFT, MDA), so the three conditions are compared on the same footing.

B.4 Perception Extensibility

To complement the reasoning-only Extensibility results of Table 5, we report perception-side scores on the two larger backbones in Table 8. The three perception benchmarks (POPE, AI2D, TextVQA)

and their evaluation protocol are the same as in Table 1. On both Qwen3-VL-8B and LLaVA-OneVision-7B, the MDA P-Avg is within 0.1 points of the corresponding Standard SFT value, supporting the claim that MDA improves reasoning while preserving perception on larger models and across architecture families.

Table 8: **Perception-side extensibility.** Perception benchmark scores on Qwen3-VL-8B and LLaVA-OneVision-7B for Standard SFT and MDA (same evaluation protocol as Table 1).

Model	Method	POPE	AI2D	TextVQA	**P-Avg**
Qwen3-VL-8B	Standard SFT	89.3	82.8	82.9	85.0
	MDA	88.5	83.7	83.2	**85.1**
LLaVA-OneVision-7B	Standard SFT	88.7	78.2	76.8	81.3
	MDA	88.5	79.9	75.9	**81.4**

B.5 Reproducibility

We will release training and evaluation code, the trained adapters for the main Qwen3-VL-2B comparison and the Qwen3-VL-8B and LLaVA-OneVision-7B configurations, and the diagnostic scripts used to compute each of the four probes above. All numbers reported in the main paper can be reproduced from the released artifacts under the training and evaluation configurations documented in Appendices B.1 and B.2.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the three contributions: (1) identifying modality-depth homogenization as a structural pathology of standard post-training, (2) proposing MDA as a minimal architectural fix, and (3) providing four mechanistic probes that support the diagnosis. All quantitative claims in the abstract ($+3.9$ reasoning improvement, perception kept almost intact, 0.05% parameter overhead) are directly supported by Table 1 and the appendix. The scope is explicitly bounded to post-training open-source VLMs under generic instruction mixtures (Limitations and Scope section in the Appendix).

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: A dedicated "Limitations and Scope" section discusses: (1) the scope boundary to post-training under generic instruction mixtures and evaluations on two VLM families (Qwen3-VL at 2B/8B and LLaVA-OneVision at 7B); (2) the parameter-budget deliberateness and the open question of higher-capacity routing variants; (3) the English-dominant evaluation regime and generalization to multilingual settings; and (4) the complementarity of the routing interface with front-end fusion modules such as Q-Former.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

   Justification: This paper makes no formal theoretical claims and contains no theorems, lemmas, or proofs. The contributions are empirical (experimental benchmarking) and mechanistic (diagnostic probes). All mathematical formulas in the paper describe the architecture and probe definitions, not theoretical results requiring proof.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Appendix B provides complete training hyperparameters (optimizer, learning rate, batch size, LoRA configuration, hardware, training time) for both the 2B and 8B settings, the full evaluation protocol (decoding parameters, benchmark filters, answer extraction rules, evaluation prompt templates), and the precise configuration of all four mechanistic probes. The base models (Qwen3-VL, LLaVA-OneVision) and training data (LLaVA-instruct-mix-VSFT) are publicly available.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [No]

   Justification: Code and trained adapter checkpoints are not released at submission time to preserve anonymity. However, Appendix B (Reproducibility) describes our plan to release training code, evaluation code, and adapter checkpoints for all reported model configurations upon publication. The training data (LLaVA-instruct-mix-VSFT) and base models are already publicly available, and the hyperparameter and protocol details in Appendix B are sufficient for an independent replication.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Appendix B provides a complete training configuration table covering optimizer, learning rate, batch size, warm-up, LoRA rank and target modules, MDA-specific hyperparameters (r_b, block size, gate initialization), hardware, and wall-clock time; it also

documents all evaluation details including decoding parameters, token budgets, benchmark-specific filters, and the prompt templates used for each benchmark class.

Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [No]

Justification: Main benchmark results are reported as single-run point estimates, as is standard in the VLM post-training literature; running multiple seeds for 28-hour training jobs on 8B+ parameter backbones is prohibitively expensive. The evaluation benchmarks themselves provide inherent statistical stability (MMMU evaluates 805 single-image items after the filters documented in Appendix B, MathVista evaluates 540 multiple-choice items after the same filter, and the remaining benchmarks are evaluated on their full public release: MMStar 1500, POPE 9000, AI2D 3088, TextVQA 5000), yielding low-variance accuracy estimates. The linear-probing result (Table 4) is reported as a 5-fold cross-validated mean.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Appendix B (Table 6) reports hardware (single NVIDIA H200, 140 GB), approximate wall-clock time per run ($\sim$4.5 h for 2B, $\sim$28 h for 8B), and gradient checkpointing settings. All main results and baselines in Table 1 use the same hardware and time budget.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.

- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research uses publicly available pretrained models and open-source instruction-tuning data. No private or sensitive data were collected or generated. No human subjects were involved. The work improves visual reasoning capabilities of open-source models, which poses no identifiable direct risk of misuse beyond the risks already associated with the base models.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: This paper presents foundational research on the post-training interface design of vision-language models. It does not introduce a new model or application deployment; it studies a structural property of existing post-training pipelines. There is no direct path from the contribution—a side-pathway architecture that improves reasoning accuracy of open-source VLMs—to specific negative societal impacts beyond those already associated with the base models studied, which are widely deployed and have their own risk assessments.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.

- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper does not introduce a new large-scale pretrained model or a novel large-scale dataset. The released artifacts will consist of lightweight adapter checkpoints and training code. The base models (Qwen3-VL, LLaVA-OneVision) are publicly distributed by their respective developers with their own terms of use; our adapter checkpoints do not materially alter the risk profile of those models.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All assets are properly cited in the bibliography. The training data (LLaVA-instruct-mix-VSFT) and the base models (Qwen3-VL series under Apache 2.0, LLaVA-OneVision under Apache 2.0) are used in accordance with their respective open-source licenses. All evaluation benchmarks (MMMU, MathVista, MMStar, POPE, AI2D, TextVQA) are standard academic benchmarks with publicly available evaluation code and are used solely for non-commercial research purposes.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

Answer: [Yes]

Justification: The new assets—MDA adapter checkpoints and training/evaluation code—are documented in Appendix B, which covers architecture configuration, training hyperparameters, evaluation protocol, and probe implementation details. The assets will be released at publication time with a README providing usage instructions.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

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

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

Answer: [N/A]

Justification: No crowdsourcing or human subjects research was conducted. All experiments use existing publicly available benchmarks evaluated with automated accuracy metrics.

Guidelines:

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

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

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper involves no human subjects research, crowdsourcing, or participant-based evaluation. IRB review is therefore not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: This paper studies VLMs as the *subject* of research (analyzing and improving their post-training interface), not as tools within the research methodology. LLMs/VLMs do not appear as novel algorithmic components of the proposed method. The core method is MDA, a lightweight architectural side pathway; no LLM is used as a generator, evaluator, or decision-maker within the method itself.

Guidelines:

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