VISCOT-STEER: DISENTANGLING AND TRANSFERRING VISUAL CHAIN-OF-THOUGHT VIA ACTIVATION STEERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Visual chain-of-thought (visual-CoT), also known as "thinking with images", equips LVLMs with the ability to locate and use key visual evidence for reasoning. However, acquiring such visual-CoT capabilities requires costly post-training and tightly couples them to a specific backbone, making transfer to other LVLMs difficult. Existing cross-model transfer methods typically treat visual-CoT as a monolithic capability, using a unified representation that ignores the functional distinction between grounding and reasoning and introduces stage-irrelevant interventions. To address this problem, we propose Viscot-steer, a training-free activation steering framework for transferring visual-CoT capabilities across heterogeneous LVLMs. Specifically, Viscot-steer first disentangles visual-CoT into grounding and reasoning vectors and extracts them from the source-model layers associated with the corresponding capabilities. It then aligns the vectors with the activation space of the target model and steers the grounding and reasoning layers during bounding-box prediction and reasoning-trace generation, respectively. Across benchmarks covering high-resolution visual understanding, general visual perception and reasoning, and mathematical visual reasoning, Viscot-steer transfers capabilities from different visual-CoT models to architecturally diverse LVLMs and outperforms the strongest capability-transfer baseline by an average of 1.86 percentage points on the primary evaluation metrics.

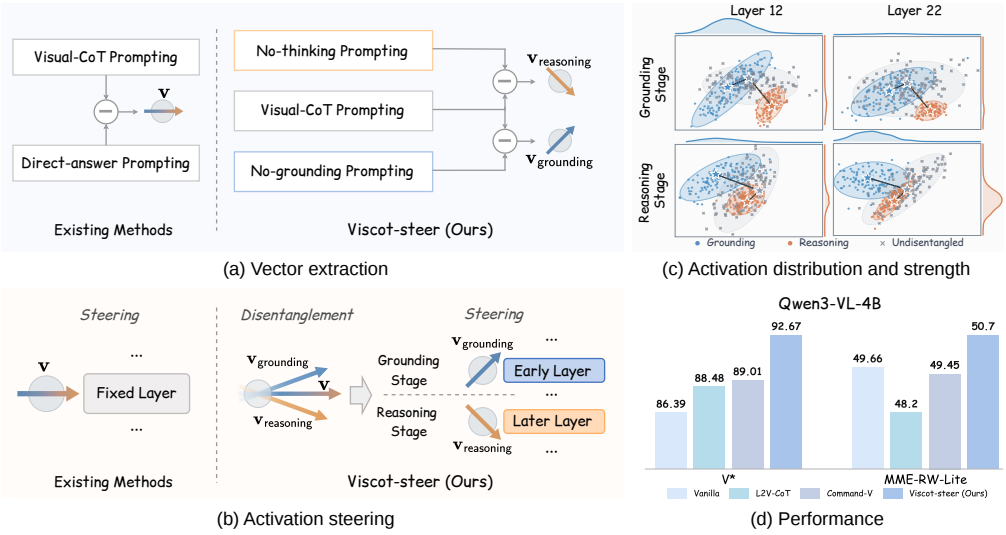
1 INTRODUCTION

Visual chain-of-thought (visual-CoT), also known as "thinking with images," enhances multimodal reasoning by interleaving textual reasoning with explicit grounding in task-relevant visual evidence. Existing methods acquire these capabilities by constructing multimodal reasoning trajectories with grounding supervision and post-training a specific base model through SFT, RL, or both (Zheng et al., 2026; Su et al., 2025; Fan et al., 2025). This tightly couples visual-CoT capabilities to the trained backbone: moving to a different base LVLM requires repeating the post-training process, incurring substantial training cost. This raises a natural question: *can visual-CoT capabilities learned by one model be transferred directly to a new base LVLM, without repeating costly post-training?*

Activation steering offers a promising route toward this goal. Its key premise is that a model capability is reflected in systematic changes in its internal activations. By comparing hidden states elicited in a source model with and without the capability of interest under matched inputs, one can isolate its associated activation direction. Cross-model transfer then can be achieved by aligning this direction with the target model's activation space and injecting it during inference, without updating model parameters. Recent work has demonstrated this paradigm for cross-model transfer across tasks, modalities, and model architectures (Wang et al., 2026a; Gan et al., 2026; Zhan et al., 2026).

Despite this progress, existing cross-model steering methods typically represent a transferable capability with a single unified direction extracted from a **fixed** layer and apply it **uniformly** throughout generation. This treatment is poorly suited to visual-CoT, which couples functionally **distinct grounding and reasoning capabilities**. As shown in Fig. 1(c), the undisentangled visual-CoT direction shifts toward grounding during the grounding stage and toward reasoning during the reasoning stage; moreover, grounding is more active at layer 12, whereas reasoning is more active at layer

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.



(a) Vector extraction (c) Activation distribution and strength

(b) Activation steering (d) Performance

Figure 1: **Disentangling and transferring visual-CoT with our Viscot-steer**. (a) Existing methods extract a unified visual-CoT vector, whereas our Viscot-steer extracts separate grounding and reasoning vectors using capability-specific contrastive prompts. (b) Existing methods inject a unified vector at a fixed layer, whereas our Viscot-steer steers each vector at its corresponding layer and generation phase. (c) Activation distributions and strengths of grounding, reasoning, and undisentangled visual-CoT at the early (12th) and later (22nd) layers during the grounding and reasoning stages. (d) Our Viscot-steer outperforms the existing baselines by a substantial margin.

22. These stage- and layer-dependent patterns motivate disentangling visual-CoT and steering each capability only at its specialized layer and corresponding generation phase.

To address this problem, we propose Viscot-steer, a training-free activation steering framework for transferring visual-CoT capabilities across heterogeneous LVLMs. Viscot-steer consists of two stages. **(1) Disentangled capability extraction (what to transfer).** We construct capability-specific contrastive prompt pairs and extract their activation differences at specialized source layers, yielding separate grounding and reasoning capability vectors (Figure 1(a)). **(2) Phase-aware capability steering (where & when to steer).** After aligning the vectors to the target activation space, we inject the reasoning vector into reasoning-specific layers during reasoning-trace generation and the grounding vector into grounding-specific layers during bounding-box prediction (Figure 1(b)). The answer phase remains unsteered, and no model parameters are updated during the transfer process.

We compare Viscot-steer with three capability-transfer baselines, including L2V-CoT (Zhan et al., 2026), on benchmarks spanning *high-resolution visual understanding, general visual perception and reasoning, and mathematical visual reasoning*. Across 25 comparable model–metric combinations, Viscot-steer achieves the best result in 21 combinations and remains close to the strongest baseline in the other four, improving over the strongest capability-transfer baseline by 1.86 percentage points on average. Ablation studies further demonstrate the effectiveness of capability disentanglement and phase-aware routing. Experiments with different source visual-CoT models provide further validation. Our main contributions are:

- **(Analysis)** Through layer- and stage-wise analyses, we show that grounding and reasoning specialize at different layers and serve bounding-box prediction and reasoning-trace generation, respectively; unified steering therefore causes cross-stage interference.
- **(Method)** We propose Viscot-steer, which separately extracts and aligns grounding and reasoning vectors and steers the corresponding capabilities at the required model layers and generation stages, enabling training-free visual-CoT transfer across heterogeneous LVLMs.
- **(Performance)** Across five LVLMs and five visual reasoning benchmarks, Viscot-steer outperforms the strongest capability-transfer baseline by an average of 1.86 pp over 25 comparable settings, with further validation across different source visual-CoT models.

2 Related Work

Visual-CoT (thinking with images). Visual chain-of-thought (visual-CoT) enhances the reasoning capabilities of LVLMs by interleaving language-based reasoning with explicit localization of relevant visual evidence (Shao et al., 2024; Lai et al., 2026; Zhang et al., 2026; Hu et al., 2024). For example, DeepEyes (Zheng et al., 2026), UV-CoT (Zhao et al., 2025a), NV-CoT (Zhao et al., 2026) and Pixel-Reasoner (Su et al., 2025) feed crops of localized image regions back to the LVLM for further reasoning, while Visual grounded reasoning (Wang et al., 2026b) and GRIT (Fan et al., 2025) associate intermediate reasoning steps with explicit points or bounding-boxes. Training visual-CoT typically requires carefully curated data and optimization through SFT, RL, or both, and must be repeated when the backbone changes. In contrast, our work transfers visual-CoT capabilities from a trained source model to architecturally heterogeneous LVLMs without additional training.

Activation steering and cross-model transfer. Activation steering controls model behavior through inference-time interventions in intermediate activations and has been widely applied to safety, factuality, generation style, and reasoning (Turner et al., 2023; Li et al., 2023; Rimsky et al., 2024; Arditi et al., 2024; Tang et al., 2025b; Zhao et al., 2025b; Wu et al., 2025; Wang et al., 2024). While early approaches primarily steer capabilities within the same model, recent studies extend activation steering to cross-model capability transfer across tasks, modalities, and model architectures (Wang et al., 2026a; Zhan et al., 2026; Tang et al., 2025b;a; Huang et al., 2025). However, these methods typically represent a transferred capability with a unified direction, which is inadequate for visual-CoT because grounding and reasoning exhibit distinct activation directions, layer-wise specialization, and generation-time roles. We instead disentangle visual-CoT into grounding and reasoning capability vectors and steer each at its corresponding model depth and generation stage.

3 Method

We first formalize visual-CoT reasoning (§3.1). Our pipeline first extracts disentangled grounding and reasoning capability vectors from a source visual-CoT LVLM (§3.2). It then aligns the extracted vectors with the activation space of a target LVLM and injects them at their corresponding generation phases via activation steering (§3.3). Figure 2 shows the overall pipeline of Viscot-steer.

3.1 Preliminaries

Setup. Let $\mathcal{D} = \{(I_i, q_i)\}_{i=1}^{N}$ denote a multimodal dataset, where I_i and q_i are the image and textual query, respectively. We denote the source visual-CoT LVLM by $\mathcal{M}_s$ and the target LVLM by $\mathcal{M}_t$. The source model $\mathcal{M}_s$ has visual-CoT capabilities, which we transfer to the target model $\mathcal{M}_t$. Let d_s and d_t denote the hidden dimensions of $\mathcal{M}_s$ and $\mathcal{M}_t$, respectively.

Visual-CoT reasoning. Visual-CoT, also known as *"thinking with images"*, enables an LVLM to interleave textual reasoning with explicit visual grounding before producing the final answer. Although existing visual-CoT models adopt different generation and interaction interfaces, we represent their output using the following unified format:

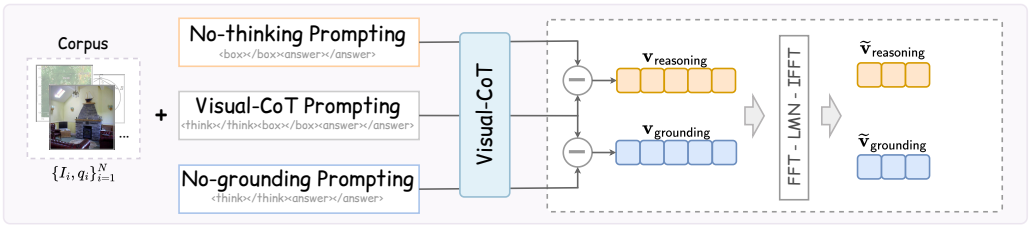
$$o = \left(\left[r_k, \mathbf{b}_k, \tilde{I}_k \right]_{k=1}^{K}, a \right), \qquad \tilde{I}_k = \mathrm{Crop}(I, \mathbf{b}_k). \tag{1}$$

This formulation represents a visual-CoT trajectory with K interleaved reasoning and grounding steps. At step k, r_k denotes the textual reasoning trace, $\mathbf{b}_k = [x_1, y_1, x_2, y_2]^\top$[1] specifies the grounded region in the original image I, $\tilde{I}_k$ is the cropped image, and a denotes the final answer.

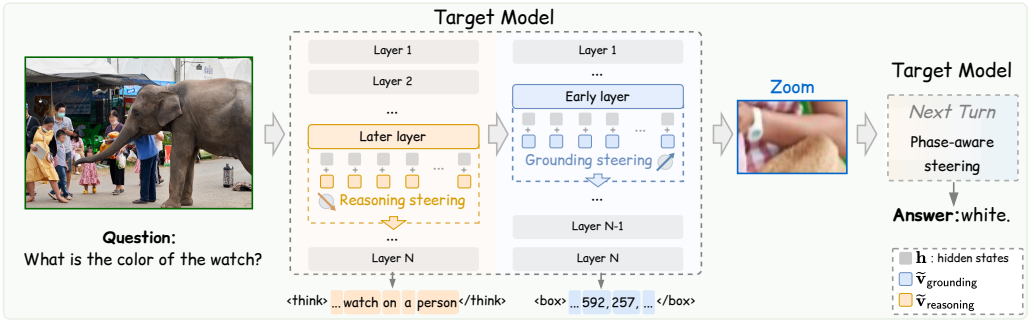
3.2 Disentangled visual-CoT capability extraction

Visual-CoT comprises two complementary capabilities: **visual grounding** for localizing task-relevant regions and **textual reasoning** for analyzing the question and deriving the final answer. These two capabilities operate at different layers of the model (Chen et al., 2025). We therefore disentangle visual-CoT into grounding and reasoning capability vectors by constructing controlled

[1] Notation x is temporarily reused for coordinates; it otherwise denotes the input.



(a) Disentangled vector extraction



(b) Phase-aware activation steering

Figure 2: **Overview of our Viscot-steer framework.** (a) Disentangled vector extraction. Disentangled reasoning and grounding vectors are extracted from capability-specific contrastive prompts. (b) Phase-aware activation steering. The reasoning and grounding vectors are routed only to their corresponding generation phases and injected at capability-specific target layers.

contrastive inputs that isolate the activation changes associated with each capability, enabling them to be extracted and steered independently.

Capability vector extraction. For each image–query pair (I_i, q_i), we construct a positive input x_i^+ that prompts $\mathcal{M}_s$ to follow the complete visual-CoT format in Eq. (1). We then construct two negative counterparts with the image and query unchanged. The grounding-negative input $x_{i,g}^-$ retains the reasoning and answering instructions but removes the bounding-box prediction and associated visual interaction. In contrast, the reasoning-negative input $x_{i,r}^-$ retains the grounding and answering instructions but omits the intermediate reasoning trace. The symbols g and r index the grounding and reasoning capabilities, respectively. Thus, each pair differs only in the capability of interest, reducing interference from visual content and task semantics.

We feed each contrastive input to the source model and collect the hidden state of its final input token. For capability $c \in \{g, r\}$, let ℓ_s^c denote its source extraction layer, and let $\mathbf{h}_{i,c}^+, \mathbf{h}_{i,c}^- \in \mathbb{R}^{d_s}$ denote the positive and negative final token activations obtained from this layer. Because grounding and reasoning exhibit distinct layer-wise specialization, we select ℓ_s^g from the early-to-middle layers and ℓ_s^r from the middle-to-late layers, following the layer-wise specialization observed in our analysis (§1). For each capability, we compute the mean activation difference over the N examples:

$$\mathbf{v}_c = \frac{1}{N} \sum_{i=1}^{N} \left(\mathbf{h}_{i,c}^+ - \mathbf{h}_{i,c}^- \right), \qquad c \in \{g, r\}. \tag{2}$$

The resulting grounding vector $\mathbf{v}_g$ represents the visual grounding capability of visual-CoT, whereas the reasoning vector $\mathbf{v}_r$ represents its reasoning capability. This disentangled construction allows the two capabilities to be transferred and controlled independently.

3.3 Phase-aware visual-CoT activation steering

Because the source and target LVLMs may have different hidden dimensions, we first map each extracted capability vector into the target activation space. We then perform phase-aware activation steering: the reasoning vector is injected at the target reasoning layer during reasoning-trace generation, and the grounding vector is injected at the target grounding layer during bounding-box

prediction. Unlike prior stage-agnostic interventions (Zhan et al., 2026; Wang et al., 2026a), our steering is dynamically routed according to the current generation phase.

Cross-architecture dimension alignment. Following Zhan et al. (2026), we adopt the LMN (Gerlach et al., 2024) for frequency-domain resampling to map each source capability vector to the target hidden dimension. For a capability vector $\mathbf{v}_c \in \mathbb{R}^{d_s}$, we obtain its target space representation as

$$\begin{aligned} \overline{\mathbf{v}}_c &= \mathrm{Re}\left[\mathcal{F}_{d_t}^{-1}\left(\mathrm{LMN}_{d_s \to d_t}\left(\mathcal{F}_{d_s}(\mathbf{v}_c)\right)\right)\right], \\ \widetilde{\mathbf{v}}_c &= \frac{\|\mathbf{v}_c\|_2}{\|\overline{\mathbf{v}}_c\|_2}\overline{\mathbf{v}}_c, \qquad c \in \{g, r\}, \end{aligned} \tag{3}$$

where $\mathcal{F}_{d_s}$ and $\mathcal{F}_{d_t}^{-1}$ denote the Fourier and inverse Fourier transforms, respectively, $\mathrm{LMN}_{d_s \to d_t}$ denotes frequency-domain resampling from d_s to d_t, and $\mathrm{Re}[\cdot]$ extracts the real part of the inverse-transformed vector. The subsequent step restores the norm of the source vector, preventing dimension alignment from changing the intervention magnitude. This procedure produces the target-space grounding and reasoning vectors $\widetilde{\mathbf{v}}_g, \widetilde{\mathbf{v}}_r \in \mathbb{R}^{d_t}$ without updating either model.

Phase-aware activation steering. Applying both capability vectors throughout generation can introduce cross-capability interference: the grounding direction may perturb textual reasoning, while the reasoning direction may interfere with bounding-box prediction. We use a phase-aware routing strategy to synchronize activation steering with the structured visual-CoT output. Let z_j denote the generation phase at decoding step j, and define the active capability as

$$c_j = \begin{cases} r, & z_j = \textsf{reasoning}, \\ g, & z_j = \textsf{grounding}. \end{cases} \tag{4}$$

The routing strategy activates the reasoning vector during reasoning-trace generation and the grounding vector during bounding-box generation. We leave the answer phase unsteered, allowing $\mathcal{M}_t$ to generate the final response based on the accumulated visual evidence and reasoning context.

At decoding step j with an active capability, let $\mathbf{h}_j \in \mathbb{R}^{d_t}$ denote the current token activation at target layer $\ell_t^{c_j}$. To preserve the scale of the original hidden state, we normalize the steered activation as

$$\begin{aligned} \overline{\mathbf{h}}_j &= \mathbf{h}_j + \alpha_c \widetilde{\mathbf{v}}_{c_j}, \\ \widehat{\mathbf{h}}_j &= \frac{\|\mathbf{h}_j\|_2}{\|\overline{\mathbf{h}}_j\|_2}\overline{\mathbf{h}}_j, \end{aligned} \tag{5}$$

where α_g and α_r control the grounding and reasoning steering strengths, respectively. The intervention is applied to the current decoding position during autoregressive generation. The target injection layers ℓ_t^g and ℓ_t^r are likewise defined separately for grounding and reasoning, preserving their layer-wise specialization.

4 Experiments

4.1 Setup

Implementation details. We use DeepEyes-7B (Zheng et al., 2026), a strong visual-CoT model built on Qwen2.5-VL-7B (Bai et al., 2025b), as our source model. For same-architecture transfer, we use Qwen2.5-VL-7B, the foundation model of DeepEyes-7B, as the target LVLM. For cross-model transfer, we use Qwen2.5-VL-3B, Qwen3-VL-4B (Bai et al., 2025a), InternVL2.5-8B (Chen et al., 2024), and the substantially larger Qwen3-VL-32B as target LVLMs, covering different model scales, families, layer depths, and hidden dimensions. Further implementation details are provided in §A, and results for Qwen2.5-VL-3B and Qwen3-VL-32B are reported in §B.

Capability vector extraction data. To extract the grounding and reasoning capability vectors, we use DeepEyes-7B as the source visual-CoT model and construct a 1,000-example subset from its training corpus, DeepEyes-Datasets-47k (Zheng et al., 2026). Following the source distribution of the original corpus, the subset contains examples of real-world visual reasoning, mathematical reasoning, and chart reasoning, covering major visual reasoning scenarios. Details are in §A.

Benchmarks. Following DeepEyes, we evaluate our method on five benchmarks spanning three visual reasoning scenarios. **(1) High-resolution visual understanding:** V* Bench (Wu & Xie, 2024)

Table 1: **Main results on high-resolution visual understanding benchmarks.** We transfer visual-CoT capabilities from DeepEyes-7B to Qwen2.5-VL-7B, Qwen3-VL-4B and InternVL2.5-8B. Results for additional target LVLMs (Vanilla) are reported in Table 10. We compare against the base target LVLMs, L2V-CoT, Command-V, and Transport and Merge on V* and HR-Bench. Best and second-best results are in **bold** and underlined, respectively.

Model	Method	V*			HR-Bench 4K			HR-Bench 8K		
		Overall	Attr.	Spatial	Overall	FSP	FCP	Overall	FSP	FCP
DeepEyes-7B	Base	88.48	90.44	85.53	74.00	91.25	56.75	71.13	84.50	57.75
Qwen2.5-VL-7B	Vanilla	75.92	71.30	82.90	68.13	80.50	55.75	62.88	76.50	49.25
	L2V-CoT	84.29	86.96	80.26	71.50	83.25	**59.75**	**67.50**	**81.25**	**53.75**
	Command-V	86.91	86.09	88.16	68.75	81.00	56.50	64.63	76.50	52.75
	Ours	**93.72**	**91.30**	**97.37**	**73.13**	**88.00**	58.25	66.88	80.25	53.50
Qwen3-VL-4B	Vanilla	86.39	89.57	81.58	78.13	89.50	66.75	72.75	86.75	58.75
	L2V-CoT	88.48	88.70	88.15	**79.75**	91.50	**68.00**	72.50	83.00	**62.00**
	Command-V	89.01	90.44	86.84	78.38	89.50	67.25	72.75	85.50	60.00
	Transport and Merge	88.48	92.17	82.90	78.50	89.00	**68.00**	73.88	85.75	**62.00**
	Ours	**92.67**	**93.91**	**90.79**	79.38	**91.75**	67.00	**74.75**	**87.50**	**62.00**
InternVL2.5-8B	Vanilla	69.63	66.96	73.68	36.25	29.50	43.00	36.50	27.50	45.50
	L2V-CoT	70.68	68.70	73.68	37.75	34.00	41.50	37.75	27.25	48.25
	Command-V	70.68	68.70	73.68	36.63	29.75	43.50	37.88	28.50	47.25
	Transport and Merge	70.68	67.83	75.00	37.63	28.75	46.50	37.38	26.50	48.25
	Ours	**72.25**	**69.57**	**76.32**	**50.38**	**44.75**	**56.00**	**49.75**	**44.25**	**55.25**

Note. Transport and Merge is omitted for Qwen2.5-VL-7B; with the same backbone, full replacement with source weights recovers DeepEyes-7B (reported separately).

and HR-Bench (Wang et al., 2025a), which evaluate fine-grained visual understanding under high-resolution inputs (HR-Bench includes both 4K and 8K settings); **(2) General visual perception and reasoning:** MME-RealWorld-Lite (Zhang et al., 2025), which assesses visual perception and reasoning across diverse real-world applications; and **(3) Mathematical visual reasoning:** MathVista (Lu et al., 2024) and MathVerse (Zhang et al., 2024), which evaluate reasoning over figures, diagrams, and other structured visual inputs.

Baselines. We compare Viscot-steer with the base target LVLMs without capability transfer (Vanilla), evaluated under the DeepEyes prompting setting, and three representative cross-model capability-transfer methods. L2V-CoT (Zhan et al., 2026) and Command-V (Wang et al., 2026a) transfer capabilities through activation steering, while Transport and Merge (Cui et al., 2026) performs cross-architecture model merging.

4.2 Main Results

Tables 1, 2, and 3 summarize the transfer results from DeepEyes-7B to three target LVLMs spanning different model families, architectures, and scales.

High-resolution benchmarks. On V* and HR-Bench, our method delivers stable improvements across most target models and evaluation settings while generally outperforming the capability-transfer baselines. Representative gains over the corresponding base LVLMs include improvements on V* with Qwen3-VL-4B (+6.28%) and HR-Bench 4K with InternVL2.5-8B (+14.13%), demonstrating that the transferred capabilities remain effective across model architectures and scales. In the same-architecture setting, our method applied to Qwen2.5-VL-7B even surpasses the source visual-CoT model DeepEyes-7B on V*.

General visual perception and reasoning. On MME-RealWorld-Lite, our method improves the overall performance of all three target LVLMs and generally outperforms the capability-transfer baselines. For example, it improves Qwen2.5-VL-7B (+4.74%), demonstrating that the transferred visual-CoT capabilities also benefit general visual perception and reasoning.

Mathematical reasoning benchmarks. On MathVista and MathVerse, our method improves all three target LVLMs. For example, it improves InternVL2.5-8B on MathVista (+3.50%) and

Table 2: **Main results on MME-RealWorld-Lite for general visual perception and reasoning.**

Model	Method	Overall	Perception					Reasoning			
			OCR	RS	DT	MO	AD	OCR	DT	MO	AD
DeepEyes-7B	Base	52.27	90.00	50.67	89.00	44.51	34.00	73.00	70.00	46.00	35.00
Qwen2.5-VL-7B	Vanilla	38.56	82.00	33.33	**87.00**	22.26	20.86	66.00	**70.00**	26.67	19.50
	L2V-CoT	41.17	**87.20**	34.67	85.00	24.76	21.43	62.00	62.00	**34.00**	**26.50**
	Command-V	42.05	86.40	40.67	82.00	**26.65**	22.29	**71.00**	68.00	30.67	25.00
	Ours	**43.30**	84.80	**44.67**	86.00	**26.65**	**27.71**	67.00	64.00	31.33	**26.50**
Qwen3-VL-4B	Vanilla	49.66	86.00	40.67	80.00	38.87	**39.14**	**74.00**	**74.00**	38.67	32.50
	L2V-CoT	48.20	85.20	32.67	81.00	39.50	36.00	69.00	**74.00**	38.67	32.25
	Command-V	49.45	86.40	39.33	80.00	39.50	**39.14**	72.00	68.00	30.67	25.00
	Transport and Merge	47.84	86.40	34.00	**83.00**	36.36	38.00	73.00	73.00	38.67	28.75
	Ours	**50.70**	**86.80**	**41.33**	80.00	**39.81**	**39.14**	**74.00**	70.00	**46.67**	**34.00**
InternVL2.5-8B	Vanilla	44.09	80.00	42.00	65.00	31.66	36.00	61.00	50.00	**40.67**	29.75
	L2V-CoT	45.49	80.00	41.33	**69.00**	32.60	**42.57**	61.00	51.00	35.33	31.00
	Command-V	44.14	79.60	41.33	64.00	32.92	36.00	**63.00**	48.00	38.00	30.75
	Transport and Merge	45.23	76.40	38.00	**69.00**	**35.11**	40.57	58.00	46.00	38.67	**33.75**
	Ours	**46.17**	**80.80**	**44.67**	**69.00**	34.48	36.57	**63.00**	**52.00**	**40.67**	33.50

Table 3: **Main results on mathematical visual reasoning benchmarks.**

Model	Method	MathVista			MathVerse					
		Overall	General	Math	Overall	TD	TL	VI	VD	VO
DeepEyes-7B	Base	66.5	69.35	64.07	43.35	50.76	44.67	42.01	39.72	39.59
Qwen2.5-VL-7B	Vanilla	64.9	69.78	60.74	38.93	45.69	38.32	36.29	35.66	38.71
	L2V-CoT	**67.2**	70.00	**64.81**	40.25	47.46	41.12	38.07	38.07	36.55
	Command-V	64.1	68.48	60.37	39.06	46.19	39.09	36.55	35.53	37.94
	Ours	65.6	**71.30**	60.74	**42.89**	**49.37**	**45.18**	**39.85**	**40.74**	**39.34**
Qwen3-VL-4B	Vanilla	66.0	62.39	**69.07**	49.72	58.63	50.38	46.45	44.92	48.22
	L2V-CoT	66.3	**67.39**	65.37	48.17	55.08	49.62	44.80	44.92	46.45
	Command-V	66.1	62.61	**69.07**	50.63	59.52	51.02	**47.34**	**46.83**	48.48
	Transport and Merge	64.5	63.91	65.00	49.39	58.12	49.62	44.80	45.81	**48.60**
	Ours	**66.9**	65.43	68.15	**50.69**	**61.29**	**51.14**	45.81	46.57	**48.60**
InternVL2.5-8B	Vanilla	58.0	61.74	54.81	35.03	42.26	35.79	33.76	**33.38**	29.95
	L2V-CoT	60.4	63.48	57.78	34.16	41.50	36.29	34.52	32.11	26.40
	Command-V	58.8	61.74	56.30	34.37	**43.40**	35.66	33.38	31.98	27.41
	Transport and Merge	58.7	64.13	54.07	34.09	41.50	33.76	33.76	32.36	29.06
	Ours	**61.5**	**65.22**	**58.32**	**35.74**	42.77	**37.44**	**34.90**	**33.38**	**30.20**

Qwen2.5-VL-7B on MathVerse (+3.96%), showing that the transferred visual-CoT capabilities generalize to mathematical visual reasoning over figures, diagrams and charts.

4.3 Ablation Studies and Visualization

Effect of capability disentanglement. We evaluate capability disentanglement on Qwen3-VL-4B by comparing the full method with variants that use only the grounding or reasoning vector or replace them with a unified visual-CoT vector. As shown in Table 4, using either capability vector alone improves performance in selected settings, while combining the two disentangled vectors achieves the best overall results on V*, MME-RealWorld-Lite, and MathVista. The full method also consistently outperforms the unified-vector variant across the three benchmarks, with the largest improvement on V* (92.67% vs. 88.48%), suggesting that using disentangled grounding and reasoning vectors at their respective generation phases is more effective than applying a unified visual-CoT vector.

Effect of phase-aware routing. We compare our phase-aware routing with a variant that applies both capability vectors throughout generation. As shown in Table 5, removing phase-aware routing

Table 4: **Ablation study of disentangled capability vectors on Qwen3-VL-4B.** $\widetilde{\mathbf{v}}_g$ and $\widetilde{\mathbf{v}}_r$ denote the grounding and reasoning capability vectors, respectively, while $\widetilde{\mathbf{v}}$ denotes the unified visual-CoT capability vector.

Method	V*			MME-RealWorld-Lite			MathVista		
	Overall	Attr.	Spatial	Overall	Percep.	Reason.	Overall	General	Math
Vanilla	86.39	89.57	81.58	49.66	52.78	44.80	66.0	62.39	**69.07**
$+\widetilde{\mathbf{v}}_g$	86.91	89.56	82.90	49.08	52.70	43.47	64.4	61.74	66.67
$+\widetilde{\mathbf{v}}_r$	92.15	**95.65**	86.84	50.03	52.78	45.73	66.3	63.70	68.52
$+\widetilde{\mathbf{v}}$	88.48	90.44	85.53	50.44	**53.98**	44.93	66.8	**65.65**	67.78
$+\widetilde{\mathbf{v}}_g, +\widetilde{\mathbf{v}}_r$	**92.67**	93.91	**90.79**	**50.70**	53.29	**46.67**	**66.9**	65.43	68.15

Table 5: **Effect of phase-aware routing on Qwen3-VL-4B.** "w/o Routing" applies both $\widetilde{\mathbf{v}}_g$ and $\widetilde{\mathbf{v}}_r$ at every decoding step.

Method	V*		
	Overall	Attr.	Spatial
Vanilla	86.39	89.57	81.58
w/o Routing	85.34	91.30	76.32
Ours	**92.67**	**93.91**	**90.79**

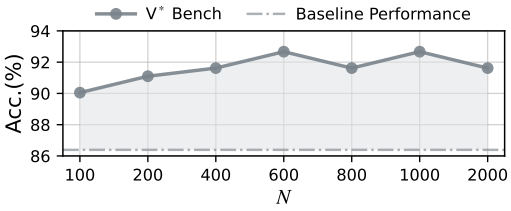


Figure 3: **Effect of N on Qwen3-VL-4B.**

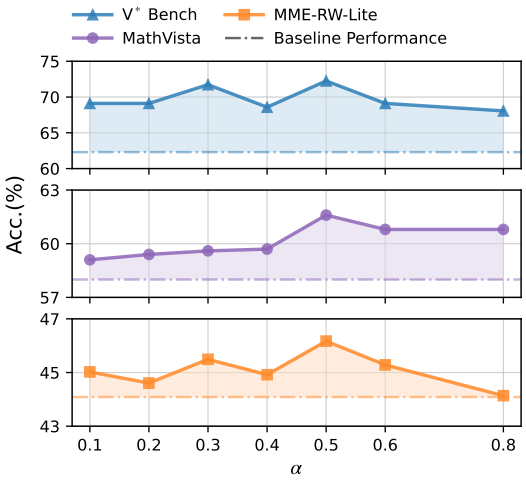


Figure 4: **Effect of α on InternVL-2.5-8B.**

reduces V* accuracy from 92.67% to 85.34%, even below the base model's 86.39%. This result shows that interventions irrelevant to the current generation phase can offset the benefits of the two capability vectors, while routing each vector to its corresponding phase effectively combines them.

Effect of steering strength. We evaluate the effect of different steering strengths α on performance using InternVL2.5-8B. The results are shown in Figure 4. Viscot-steer maintains stable improvements over a range of α values, demonstrating robustness to the steering strength, while $\alpha = 0.5$ yields the strongest and most consistent performance across the three benchmarks.

Effect of extraction data size. We investigate how the number of examples N used for capability-vector extraction affects performance. The results are shown in Figure 3. With only 100 examples, Viscot-steer already achieves 90.05%, substantially outperforming the base model's 86.39%. Performance generally improves as N increases, reaching 92.67% at $N = 600$, after which further increases yield no consistent gains. This suggests that our method requires only a small extraction set to capture general and robust visual-CoT representations.

Effect of steering layers. We sweep the grounding and reasoning steering layers on Qwen3-VL-4B using V*. As shown in Figure 6, earlier or intermediate grounding layers generally perform best when paired with later reasoning layers, while injecting the grounding vector into very late layers substantially degrades performance. The optimal combination, grounding at layer 9 and reasoning at layer 20, achieves 92.67% (+6.28%). These results support steering the two capabilities at distinct, capability-specific layers and are consistent with grounding preceding higher-level reasoning.

Visualization. Figure 5 presents representative examples comparing the outputs of the base LVLM and Viscot-steer. By steering the model toward task-relevant regions, Viscot-steer corrects the erroneous output trajectories of the base model and produces the correct answers.

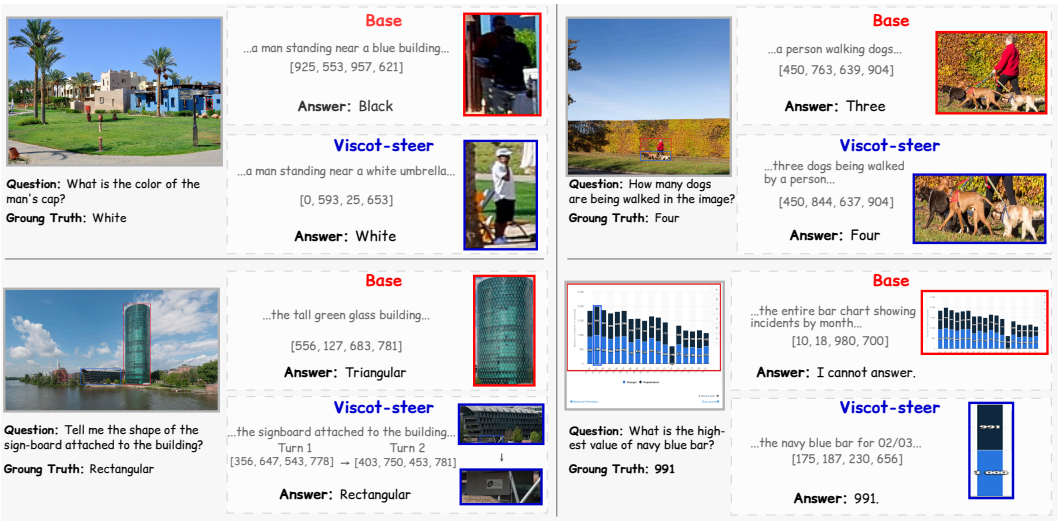


Figure 5: Visualization of the responses generated by the base LVLM and Viscot-steer.

Table 6: **Transfer results** with GRIT as the source Visual-CoT model.

Model	Method	VSR	TallyQA
GRIT-3B	Base	75.09	47.45
Qwen3-VL-4B	Vanilla	55.96	51.53
	L2V-CoT	61.73	52.34
	Command-V	58.84	53.56
	Transport and Merge	54.87	50.71
	Ours	**64.26**	**54.58**
InternVL3-2B	Vanilla	33.57	38.90
	L2V-CoT	**48.01**	40.33
	Command-V	36.82	40.94
	Transport and Merge	42.60	38.49
	Ours	46.57	**42.36**

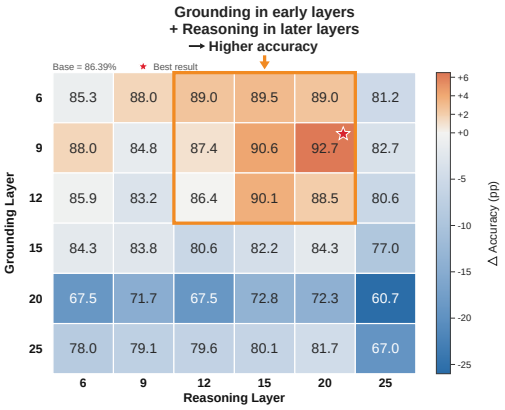


Figure 6: **Effect of steering layer**

4.4 Transfer from an alternative source model

To evaluate transfer from a different source visual-CoT model, we use GRIT (Fan et al., 2025), built on Qwen2.5-VL-3B, as the source model. We use Qwen3-VL-4B, and InternVL3-2B (Zhu et al., 2025) as target LVLMs and evaluate on VSR (Liu et al., 2023) and TallyQA (Acharya et al., 2019). We follow the evaluation protocol of GRIT (Fan et al., 2025).

As shown in Table 6, Viscot-steer improves all target LVLMs on both benchmarks. For example, it improves Qwen3-VL-4B on VSR (+8.30%) and TallyQA (+3.05%). These results further demonstrate that Viscot-steer can transfer visual-CoT capabilities from different source models.

5 Conclusion

In this paper, we proposed Viscot-steer, a training-free framework for transferring visual-CoT capabilities across heterogeneous LVLMs. Viscot-steer disentangles visual-CoT into grounding and reasoning capability vectors, aligns them with the target activation space, and steers each capability at its corresponding model layer and generation phase. Extensive experiments across five target LVLMs and five visual reasoning benchmarks demonstrate that Viscot-steer outperforms representative capability-transfer baselines in most model–benchmark settings, while experiments with different source visual-CoT models further validate its generalizability.

AI USE STATEMENT

In this work, we used generative AI tools, primarily OpenAI's GPT-5.6 Sol and GPT-5.5, solely for language polishing, translation, and manuscript formatting. We did not use generative AI tools for other research tasks, including developing research ideas, designing or evaluating methodology or experiments, implementing methods, generating synthetic data, processing research data, or interpreting results. All AI-assisted text and formatting were manually reviewed by the authors, who take full responsibility for the final manuscript and research artifacts.

ETHICS STATEMENT

This work adheres to the ICLR Code of Ethics. In this study, no human subjects or animal experimentation was involved. All datasets used, including V*, HR-Bench, and MathVista, among others, were sourced in compliance with usage guidelines, ensuring no violation of privacy. We have taken care to avoid any biases or discriminatory outcomes in our research process. No personally identifiable information was used, and no experiments were conducted that could raise privacy or security concerns. We are committed to transparency and integrity throughout the research process.

REPRODUCIBILITY STATEMENT

Our experiments use publicly available backbone models and benchmarks, including Qwen2.5-VL, Qwen3-VL, InternVL, V*, HR-Bench, and MathVista. The method section describes capability-vector extraction, cross-model dimension alignment, and phase-aware activation steering. Appendix A provides implementation details, including contrastive prompts, the composition of the extraction dataset, hyperparameter settings for the main experiments, and inference and evaluation settings. Together with the publicly available models and datasets, these details provide sufficient information to reproduce our method and independently verify the reported results.

REFERENCES

Manoj Acharya, Kushal Kafle, and Christopher Kanan. Tallyqa: Answering complex counting questions. In *AAAI*, 2019.

Andy Arditi, Oscar Balcells Obeso, Aaquib Syed, Daniel Paleka, Nina Rimsky, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. In *NeurIPS*, 2024.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025a.

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

Shiqi Chen, Jinghan Zhang, Tongyao Zhu, Wei Liu, Siyang Gao, Miao Xiong, Manling Li, and Junxian He. Bring reason to vision: Understanding perception and reasoning through model merging. In *ICML*, 2025.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, Lixin Gu, Xuehui Wang, Qingyun Li, Yiming Ren, Zixuan Chen, Jiapeng Luo, Jiahao Wang, Tan Jiang, Bo Wang, Conghui He, Botian Shi, Xingcheng Zhang, Han Lv, Yi Wang, Wenqi Shao, Pei Chu, Zhongying Tu, Tong He, Zhiyong Wu, Huipeng

Deng, Jiaye Ge, Kai Chen, Kaipeng Zhang, Limin Wang, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling, 2024.

Chenhang Cui, Binyun Yang, Fei Shen, Yuxin Chen, Jingnan Zheng, Xiang Wang, An Zhang, and Tat-Seng Chua. Transport and merge: Cross-architecture merging for large language models. In *ICML*, 2026.

Yue Fan, Xuehai He, Diji Yang, Kaizhi Zheng, Ching-Chen Kuo, Yuting Zheng, Sravana Jyothi Narayanaraju, Xinze Guan, and Xin Eric Wang. Grit: Teaching mllms to think with images. In *NeurIPS*, 2025.

Woody Haosheng Gan, Deqing Fu, Julian Asilis, Ollie Liu, Vatsal Sharan, Robin Jia, and Willie Neiswanger. Textual steering vectors can improve visual understanding in multimodal large language models. In *ACL*, 2026.

Lino Gerlach, Wenqiang Gu, Nitish Nayak, Xin Qian, and Brett Viren. Waveform resampling with lmn method, 2024.

Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah Smith, and Ranjay Krishna. Visual sketchpad: Sketching as a visual chain of thought for multimodal language models. In *NeurIPS*, 2024.

Youcheng Huang, Chen Huang, Duanyu Feng, Wenqiang Lei, and Jiancheng Lv. Cross-model transferability among large language models on the platonic representations of concepts. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *ACL*, 2025.

Xin Lai, Junyi Li, Wei Li, Tao Liu, Tianjian Li, and Hengshuang Zhao. Mini-o3: Scaling up reasoning patterns and interaction turns for visual search. In *ICLR*, 2026.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *NeurIPS*, 2023.

Lei Li, Yuqi Wang, Runxin Xu, Peiyi Wang, Xiachong Feng, Lingpeng Kong, and Qi Liu. Multimodal ArXiv: A dataset for improving scientific comprehension of large vision-language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *ACL*, 2024.

Fangyu Liu, Guy Emerson, and Nigel Collier. Visual spatial reasoning. *TACL*, 2023.

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 *ICLR*, 2024.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering llama 2 via contrastive activation addition. In *ACL*, 2024.

Peter H. Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 1966.

Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. Visual cot: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. In *NeurIPS*, 2024.

Alex Su, Haozhe Wang, Weiming Ren, Fangzhen Lin, and Wenhu Chen. Pixel reasoner: Incentivizing pixel-space reasoning with curiosity-driven reinforcement learning. In *NeurIPS*, 2025.

Xinyu Tang, Zhihao Lv, Xiaoxue Cheng, Junyi Li, Wayne Xin Zhao, Zujie Wen, Zhiqiang Zhang, and Jun Zhou. Enhancing cross-task transfer of large language models via activation steering, 2025a.

Xinyu Tang, Xiaolei Wang, Zhihao Lv, Yingqian Min, Wayne Xin Zhao, Binbin Hu, Ziqi Liu, and Zhiqiang Zhang. Unlocking general long chain-of-thought reasoning capabilities of large language models via representation engineering. In *ACL*, 2025b.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2023.

Barry Wang, Avi Schwarzschild, Alexander Robey, Ali Payani, Charles Fleming, Mingjie Sun, and Daphne Ippolito. Command-v: Training-free representation finetuning transfer. In *ICLR*, 2026a.

Jiacong Wang, Zijian Kang, Haochen Wang, Xiao Liang, Ya Wang, Jiawen Li, Bohong Wu, Jiao Ran, Haiyong Jiang, Chao Feng, et al. Vgr: Visual grounded reasoning. In *ICLR*, 2026b.

Pengyu Wang, Dong Zhang, Linyang Li, Chenkun Tan, Xinghao Wang, Mozhi Zhang, Ke Ren, Botian Jiang, and Xipeng Qiu. InferAligner: Inference-time alignment for harmlessness through cross-model guidance. In *EMNLP*, 2024.

Wenbin Wang, Liang Ding, Minyan Zeng, Xiabin Zhou, Li Shen, Yong Luo, Wei Yu, and Dacheng Tao. Divide, conquer and combine: A training-free framework for high-resolution image perception in multimodal large language models. In *AAAI*, 2025a.

Xiyao Wang, Zhengyuan Yang, Chao Feng, Hongjin Lu, Linjie Li, Chung-Ching Lin, Kevin Lin, Furong Huang, and Lijuan Wang. Sota with less: Mcts-guided sample selection for data-efficient visual reasoning self-improvement. In *NeurIPS*, 2025b.

Junfei Wu, Yue Ding, Guofan Liu, Tianze Xia, Ziyue Huang, Dianbo Sui, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. SHARP: Steering hallucination in LVLMs via representation engineering. In *EMNLP*, 2025.

Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal llms. In *CVPR*, 2024.

Yuliang Zhan, Xinyu Tang, Han Wan, Jian Li, Ji-Rong Wen, and Hao Sun. L2v-cot: Cross-modal transfer of chain-of-thought reasoning via latent intervention. In *AAAI*, 2026.

Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, Peng Gao, and Hongsheng Li. Mathverse: Does your multimodal llm truly see the diagrams in visual math problems? In *ECCV*, 2024.

YiFan Zhang, Huanyu Zhang, Haochen Tian, Chaoyou Fu, Shuangqing Zhang, Junfei Wu, Feng Li, Kun Wang, Qingsong Wen, Zhang Zhang, Liang Wang, and Rong Jin. MME-realworld: Could your multimodal LLM challenge high-resolution real-world scenarios that are difficult for humans? In *ICLR*, 2025.

YiFan Zhang, Xingyu Lu, Shukang Yin, Chaoyou Fu, Wei Chen, Xiao Hu, Bin Wen, Kaiyu Jiang, Changyi Liu, Tianke Zhang, Haonan Fan, Kaibing Chen, Jiankang Chen, Haojie Ding, Kaiyu Tang, Zhang Zhang, Liang Wang, Fan Yang, Tingting Gao, and Guorui Zhou. Thyme: Think beyond images. In *ICLR*, 2026.

Kesen Zhao, Beier Zhu, Qianru Sun, and Hanwang Zhang. Unsupervised visual chain-of-thought reasoning via preference optimization. In *ICCV*, 2025a.

Kesen Zhao, Beier Zhu, Junbao Zhou, Xingyu Zhu, Zhongqi Yue, and Hanwang Zhang. Thinking with images as continuous actions: Numerical visual chain-of-thought, 2026.

Zekai Zhao, Qi Liu, Kun Zhou, Zihan Liu, Yifei Shao, Zhiting Hu, and Biwei Huang. Activation control for efficiently eliciting long chain-of-thought ability of language models. In *NeurIPS*, 2025b.

Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. Deepeyes: Incentivizing "thinking with images" via reinforcement learning. In *ICLR*, 2026.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, Zhangwei Gao, Erfei Cui, Xuehui Wang, Yue Cao, Yangzhou Liu, Xingguang Wei, Hongjie Zhang, Haomin Wang, Weiye Xu, Hao Li, Jiahao Wang, Nianchen Deng, Songze Li, Yinan He, Tan Jiang, Jiapeng Luo, Yi Wang, Conghui He, Botian Shi, Xingcheng Zhang, Wenqi Shao, Junjun He, Yingtong Xiong, Wenwen Qu, Peng Sun, Penglong Jiao, Han Lv, Lijun Wu, Kaipeng Zhang, Huipeng Deng, Jiaye Ge, Kai Chen, Limin Wang, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models, 2025.

A Implementation Details

A.1 Prompts

We use three user prompts to extract contrastive capability vectors from DeepEyes-7B (Zheng et al., 2026), the source model in our main experiments. The official visual-CoT prompt serves as the positive input for both capabilities. The no-thinking prompt removes intermediate reasoning while retaining tool use, whereas the no-grounding prompt disables tool use while retaining reasoning. Each contrastive pair uses the same image and question. The prompts are below:

Visual-CoT prompt

System prompt: Official DeepEyes system prompt (Zheng et al., 2026).
User prompt: Think first, call image_zoom_in_tool if needed, then answer. Format strictly as:
<think>...</think> <tool_call>...</tool_call> (if tools are needed)
<answer>...</answer>

No-thinking prompt

System prompt: Official DeepEyes system prompt (Zheng et al., 2026).
User prompt: Call image_zoom_in_tool if close visual inspection is needed; otherwise answer directly. Do not write any thinking, rationale, analysis, or <think> tags. Output exactly one of these formats:
<tool_call>...</tool_call> or <answer>...</answer>
If answering, put only the selected option letter and answer text inside <answer>...</answer>.

No-grounding prompt

User prompt: Think first, then answer. Do not call tools. Format strictly as:
<think>...</think> <answer>...</answer>

A.2 Hyperparameter Settings

The source extraction layers, target steering layers, and steering strengths are the key hyperparameters for activation-based capability transfer. To ensure a fair comparison, we tune the key hyperparameters of Viscot-steer and all baselines using the same training split for each benchmark. For Viscot-steer, we fix the source grounding and reasoning extraction layers and tune the corresponding target steering layers and shared steering strength. For L2V-CoT, we tune the source extraction layer, target injection layer, and injection strength. For Command-V, we tune the injection layer and injection strength. The detailed settings are reported in Tables 7, 8, and 9. For Transport and Merge (Cui et al., 2026), we set $\alpha = 0.05$, `neuron-topk`$= 128$, and `layer-topk`$= 3$.

Table 7: **Hyperparameter settings of Viscot-steer.** ℓ_t^g and ℓ_t^r denote the target grounding and reasoning steering layers, respectively. α denotes the steering strength, where $\alpha = \alpha_g = \alpha_r$. The source extraction layers are fixed to $\ell_s^g = 12$ and $\ell_s^r = 22$ for all settings. MME-RW-Lite denotes MME-RealWorld-Lite.

Model	V*			HR-Bench			MME-RW-Lite			MathVista			MathVerse		
	ℓ_t^g	ℓ_t^r	α	ℓ_t^g	ℓ_t^r	α	ℓ_t^g	ℓ_t^r	α	ℓ_t^g	ℓ_t^r	α	ℓ_t^g	ℓ_t^r	α
Qwen2.5-VL-7B	12	22	1.0	12	22	1.0	12	22	1.0	12	25	1.0	12	25	1.0
Qwen3-VL-4B	9	20	0.5	6	25	0.5	9	20	0.5	15	23	0.5	15	23	0.5
InternVL2.5-8B	12	23	0.5	12	23	0.5	15	27	0.5	16	22	0.5	16	22	0.5

Table 8: **Hyperparameter settings of L2V-CoT (Zhan et al., 2026).** ℓ_s and ℓ_t denote the source extraction and target injection layers, respectively, and α denotes the injection strength.

Target model	V*			HR-Bench 4K			HR-Bench 8K			MME-RW-Lite			MathVista			MathVerse		
	ℓ_s	ℓ_t	α	ℓ_s	ℓ_t	α	ℓ_s	ℓ_t	α	ℓ_s	ℓ_t	α	ℓ_s	ℓ_t	α	ℓ_s	ℓ_t	α
Qwen2.5-VL-7B	8	8	1.0	8	8	1.0	9	9	1.0	9	9	1.0	9	9	1.0	9	9	1.0
Qwen3-VL-4B	8	14	0.5	8	13	0.5	9	7	0.5	9	8	0.5	9	13	0.5	9	10	0.5
InternVL2.5-8B	8	19	0.5	8	15	0.5	9	20	0.5	9	18	0.5	9	14	0.5	9	18	0.5

Table 9: **Hyperparameter settings of Command-V (Wang et al., 2026a) .** ℓ and α denote the injection layer and injection strength, respectively.

Target model	V*		HR-Bench		MME-RW-Lite		MathVista		MathVerse	
	ℓ	α	ℓ	α	ℓ	α	ℓ	α	ℓ	α
Qwen2.5-VL-7B	12	2.0	12	2.0	12	2.0	12	2.0	12	2.0
Qwen3-VL-4B	3	2.0	5	2.0	5	2.0	9	2.0	9	2.0
InternVL2.5-8B	10	2.0	10	2.0	10	2.0	10	2.0	10	2.0

A.3 More details

More details on capability vector extraction data. For capability vector extraction, we sample 1,000 examples from DeepEyes-Datasets-47k (Zheng et al., 2026), preserving its source proportions: 475 from the V* (Wu & Xie, 2024) training split, 290 from ArxivQA (Li et al., 2024), and 235 from ThinkLite-VL (Wang et al., 2025b). In terms of visual content, the subset comprises real-world scenes and objects (543 examples), scientific and mathematical imagery (453 examples), and other visual content (4 examples).

Inference and Evaluation. For all evaluations, we use **greedy decoding**, making the reported results deterministic in principle. We support multi-turn visual interaction and allow at most four interaction rounds for each example. All models are evaluated using the Hugging Face `Transformers` inference framework with a batch size of 1. To ensure a fair comparison, we keep the inference and evaluation settings identical across all methods and individual experimental runs.

B Additional Experiments

Additional target models. Tables 10, 11, and 12 report results for Qwen2.5-VL-3B and Qwen3-VL-32B. In particular, transferring capabilities from DeepEyes-7B improves Qwen3-VL-32B's V* accuracy from 92.15% to 94.24%, showing that Viscot-steer remains effective for substantially larger target LVLMs.

Additional results on capability disentanglement. Table 13 reports additional capability disentanglement results on InternVL2.5-8B. Combining the grounding and reasoning vectors achieves the best overall performance on V*, MME-RealWorld-Lite, and MathVista, outperforming the unified visual-CoT vector on all three benchmarks. This result further validates the effectiveness of capability disentanglement across different target models.

Table 10: **Additional results on high-resolution visual understanding benchmarks.** We transfer visual-CoT capabilities from DeepEyes-7B to Qwen2.5-VL-3B and Qwen3-VL-32B. We compare against the base target LVLMs (Vanilla), L2V-CoT, Command-V, and Transport and Merge on V* and HR-Bench.

Model	Method	V*			HR-Bench 4K			HR-Bench 8K		
		Overall	Attr.	Spatial	Overall	FSP	FCP	Overall	FSP	FCP
DeepEyes-7B	Base	88.48	90.44	85.53	74.00	91.25	56.75	71.13	84.50	57.75
Qwen2.5-VL-3B	Vanilla	54.45	53.91	55.26	48.13	50.25	46.00	41.75	45.75	37.75
	L2V-CoT	63.35	65.21	**60.53**	44.50	49.50	39.50	45.63	50.00	41.25
	Command-V	57.59	56.52	59.21	49.13	52.00	46.25	43.75	48.25	39.25
	Transport and Merge	53.93	53.91	53.95	47.25	50.75	43.75	39.38	43.75	35.00
	Ours	**66.49**	**70.43**	**60.53**	**58.25**	**68.25**	**48.25**	**48.75**	**52.50**	**45.00**
Qwen3-VL-32B	Vanilla	92.15	92.17	92.11	86.75	**94.25**	79.25	80.88	90.00	71.75
	L2V-CoT	93.72	93.91	**93.42**	87.00	93.75	80.25	82.88	89.50	76.25
	Command-V	93.72	93.91	**93.42**	86.75	93.50	80.00	82.00	90.25	73.75
	Transport and Merge	91.10	93.04	88.16	**88.00**	93.75	**82.25**	**84.00**	87.50	**80.50**
	Ours	**94.24**	**94.78**	**93.42**	86.75	94.00	79.50	83.75	**90.75**	76.75

Table 11: **Additional results on MME-RealWorld-Lite.**

Model	Method	Overall	Perception					Reasoning			
			OCR	RS	DT	MO	AD	OCR	DT	MO	AD
DeepEyes-7B	Base	52.27	90.00	50.67	89.00	44.51	34.00	73.00	70.00	46.00	35.00
Qwen2.5-VL-3B	Vanilla	33.72	68.80	31.33	44.00	25.71	29.14	58.00	33.00	28.67	16.50
	L2V-CoT	35.64	69.60	**36.00**	49.00	25.39	32.29	**59.00**	38.00	**30.00**	17.75
	Command-V	33.98	69.20	34.00	44.00	25.71	29.14	58.00	33.00	28.67	16.50
	Transport and Merge	31.74	65.60	27.33	35.00	26.33	27.43	55.00	34.00	25.33	15.50
	Ours	**39.71**	**70.40**	24.00	**59.00**	**26.96**	**37.43**	57.00	**52.00**	26.67	**31.25**
Qwen3-VL-32B	Vanilla	56.90	90.80	58.00	91.00	41.69	43.43	83.00	**91.00**	**54.67**	36.50
	L2V-CoT	56.80	**92.00**	**60.00**	91.00	41.38	42.00	84.00	87.00	50.67	38.25
	Command-V	56.70	90.80	55.33	91.00	41.69	43.43	83.00	**91.00**	**54.67**	36.50
	Transport and Merge	57.63	91.20	59.33	**92.00**	43.26	**45.71**	80.00	88.00	50.00	**39.00**
	Ours	**57.69**	91.20	57.33	88.00	**45.14**	44.57	**86.00**	90.00	52.00	37.75

Table 12: **Additional results on mathematical visual reasoning benchmarks.**

Model	Method	MathVista			MathVerse					
		Overall	General	Math	Overall	TD	TL	VI	VD	VO
DeepEyes-7B	Base	66.5	69.35	64.07	43.35	50.76	44.67	42.01	39.72	39.59
Qwen2.5-VL-3B	Vanilla	51.9	60.87	44.26	32.11	37.44	33.12	30.84	29.95	**29.19**
	L2V-CoT	**54.3**	62.39	**47.41**	31.93	**38.45**	31.85	31.09	30.96	27.28
	Command-V	52.5	61.30	45.00	32.03	36.42	33.38	30.84	30.58	28.93
	Transport and Merge	52.2	59.35	46.11	32.64	**38.45**	33.38	31.09	31.47	28.81
	Ours	53.9	**64.13**	45.19	**33.07**	37.31	**34.77**	**31.60**	**32.49**	**29.19**
Qwen3-VL-32B	Vanilla	78.1	77.17	78.89	59.57	66.24	60.79	55.84	55.58	59.39
	L2V-CoT	**79.4**	77.39	**81.11**	58.81	65.86	59.90	54.95	55.33	57.99
	Command-V	78.4	**78.04**	78.70	59.26	**67.01**	60.28	54.95	55.84	58.25
	Transport and Merge	78.1	77.61	78.52	59.04	65.23	60.03	55.58	55.46	58.88
	Ours	78.1	76.52	80.56	**59.90**	55.84	**61.42**	**56.35**	**58.63**	**67.26**

Additional results on phase-aware routing. Table 14 reports additional phase-aware routing results on InternVL2.5-8B. Removing routing reduces V* accuracy from 72.25% to 62.30%, below the vanilla model's 69.63%. This result further validates the effectiveness of phase-aware routing across different target models.

Table 13: **Additional ablation study of disentangled capability vectors on InternVL2.5-8B.** $\widetilde{\mathbf{v}}_g$ and $\widetilde{\mathbf{v}}_r$ denote the grounding and reasoning capability vectors, respectively, while $\widetilde{\mathbf{v}}$ denotes the unified visual-CoT capability vector.

Method	V*			MME-RealWorld-Lite			MathVista		
	Overall	Attr.	Spatial	Overall	Percep.	Reason.	Overall	General	Math
Vanilla	69.63	66.96	73.68	44.09	47.48	38.80	58.0	61.74	54.81
$+\widetilde{\mathbf{v}}_g$	70.68	67.83	75.00	44.14	46.96	39.73	58.3	60.87	56.11
$+\widetilde{\mathbf{v}}_r$	70.68	67.83	75.00	44.92	48.33	39.60	59.9	64.78	55.74
$+\widetilde{\mathbf{v}}$	70.16	67.83	73.68	45.23	48.16	40.67	60.0	63.48	57.04
$+\widetilde{\mathbf{v}}_g, +\widetilde{\mathbf{v}}_r$	**72.25**	**69.57**	**76.32**	**46.17**	**49.27**	**41.33**	**61.5**	**65.22**	**58.32**

Table 14: **Additional results on phase-aware routing with InternVL2.5-8B.**

Method	V*		
	Overall	Attr.	Spatial
Vanilla	69.63	66.96	73.68
w/o Routing	62.30	61.74	63.16
Ours	**72.25**	**69.57**	**76.32**

Table 15: **Alternative alignment.**

Method	V*		
	Overall	Attr.	Spatial
Vanilla	86.39	89.57	81.58
LMN	**92.67**	**93.91**	**90.79**
Projection-based	**92.67**	**93.91**	**90.79**

Table 16: **Effect of Viscot-steer on bounding-box IoU and response length on V* using Qwen2.5-VL-7B.** Resp. Len. is the mean number of tokens in the model-generated trajectory; IoU denotes the mean IoU of the bounding box. Δ denotes the change from Vanilla to Viscot-steer.

Method	Acc.	Resp. Len.	IoU
DeepEyes-7B	88.48	835.8	0.3911
Vanilla	75.92	98.9	0.3102
$+\widetilde{\mathbf{v}}_g$	86.39	100.8	0.3445
$+\widetilde{\mathbf{v}}_r$	86.39	161.1	0.2984
$+\widetilde{\mathbf{v}}_g, +\widetilde{\mathbf{v}}_r$	**93.72**	**162.0**	**0.3472**
Δ	+17.80	+63.1	+0.0370

Bounding-box overlap and response length. Table 16 compares grounding quality and response length on V* using Qwen2.5-VL-7B. Grounding steering improves grounding precision, while reasoning steering produces deeper and more accurate reasoning traces. Combining both vectors yields an IoU of 0.3472 with an average response length of 162.0 tokens. These results demonstrate that Viscot-steer jointly enhances precise visual grounding and high-quality reasoning.

Alternative alignment. To verify that our gains do not depend on the frequency-domain LMN component, we replace it with a parameter-free *Projection* alignment while keeping all other settings fixed. For each capability $c \in \{g, r\}$, let $\mathbf{X}_s^c \in \mathbb{R}^{N\times d_s}$ and $\mathbf{X}_t^c \in \mathbb{R}^{N\times d_t}$ be the centered, paired activation-difference matrices, and let $\mathbf{V}_s^c$ and $\mathbf{V}_t^c$ be their complete sample-supported PCA bases with $p = \min(N-1, d_s, d_t)$ components. Projection aligns the paired PCA coordinates with an orthogonal map (Schönemann, 1966) and transports the source vector as

$$\mathbf{R}_c^* = \underset{\mathbf{R}^\top\mathbf{R}=\mathbf{I}}{\arg\min} \|\mathbf{X}_s^c\mathbf{V}_s^c\mathbf{R} - \mathbf{X}_t^c\mathbf{V}_t^c\|_F^2, \qquad \widetilde{\mathbf{v}}_c = \mathbf{V}_t^c(\mathbf{R}_c^*)^\top(\mathbf{V}_s^c)^\top\mathbf{v}_c. \tag{6}$$

As shown in Table 15, Projection-based alignment achieves the same performance as LMN on V* using Qwen3-VL-4B under identical experimental settings, confirming that the effectiveness of our capability-transfer framework is not tied to a particular cross-architecture alignment operator.

Inference latency. We measure the inference latency of Viscot-steer and the vanilla model on V* using Qwen3-VL-4B and a single NVIDIA H100 80GB GPU with a batch size of 1. Under otherwise identical settings, the vanilla model requires 19.96 ms per generated token, whereas Viscot-steer requires 21.77 ms per token, corresponding to a latency ratio of only $1.09\times$. This result shows that Viscot-steer introduces negligible inference overhead while substantially improving accuracy.

Table 17: **Inference latency on V* using Qwen3-VL-4B.**

Method	Acc.	ms/token	Latency ratio
Vanilla	86.39	19.96	$1.00\times$
Ours	**92.67**	21.77	$1.09\times$