



Steer Where It Matters: Token-Level Visual-Sensitivity Steering for LVLMs Hallucination Mitigation



Ruipeng Zhang^{1,3}, Zhihao Li^{1,2,3}, C.L. Philip Chen^{1,2,3}, Tong Zhang^{1,2,3*}

¹Guangdong Provincial Key Laboratory of Computational AI Models and Cognitive Intelligence, School of Computer Science & Engineering, South China University of Technology

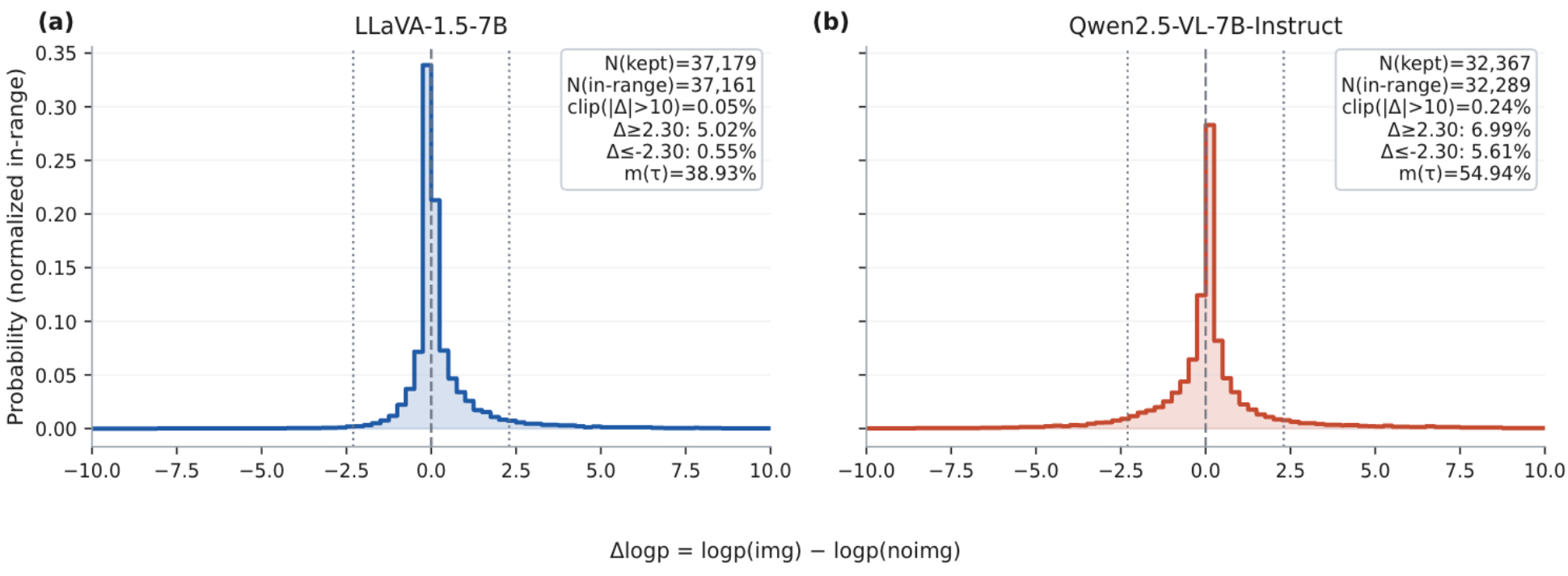
²Pazhou Lab, Guangzhou, China

³Engineering Research Center of the Ministry of Education on Health Intelligent Perception and Paralleled Digital-Human, Guangzhou, China

Motivation

Method

Partial Experimental Results



Our motivation comes from three observations.

- First, visual information only affects a small number of tokens during generation.
- Second, traditional activation steering averages visual signals across all tokens, which weakens the important image-related information.
- Third, applying the same steering signal to every token may change many tokens that do not need adjustment, so a token-level adaptive method is needed.

Problem discussion

Observation 1

We design a token-level diagnosis using teacher forcing with and without the image. For each token, we compute

$$\Delta_t = \log p(y_t | x, I, y_{<t}) - \log p(y_t | x, \emptyset, y_{<t})$$

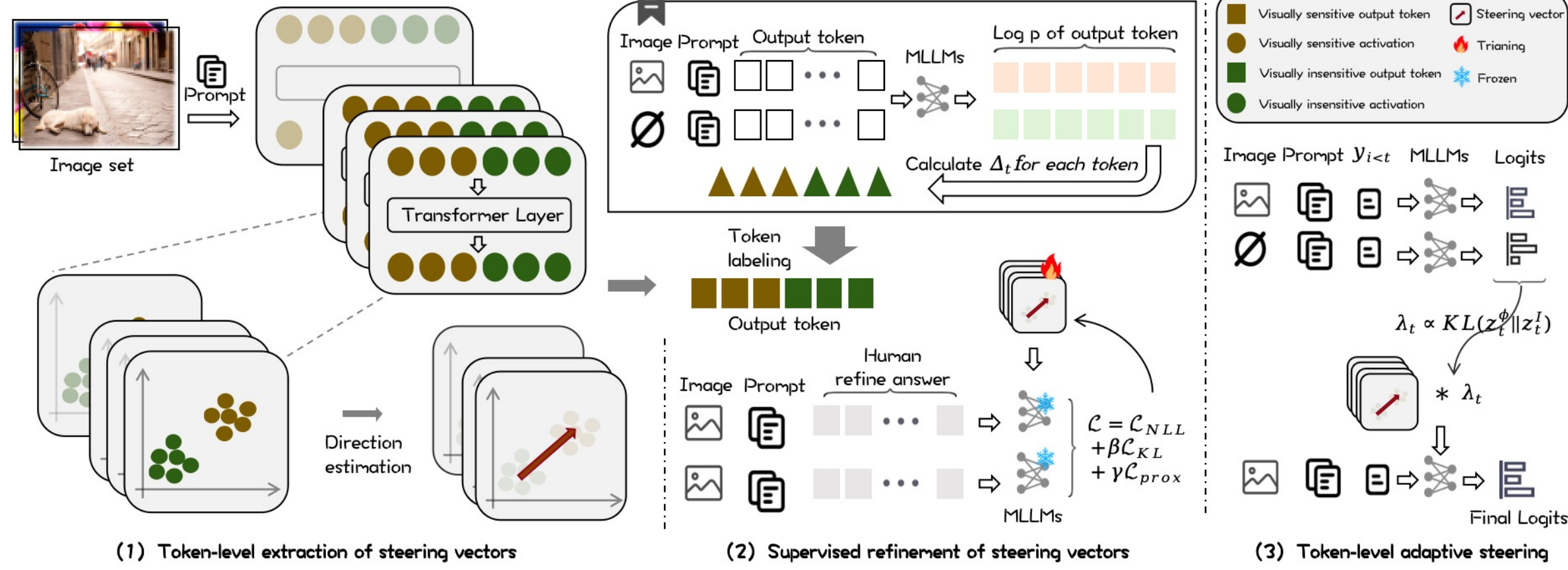
We find that most tokens have ($\Delta_t \approx 0$), while only a few tokens are strongly affected by the image.

Observation 2

We further test how global steering affects each token using teacher forcing. For each token, we measure

$$\Delta \text{NLL}_t(\text{method} - \text{vanilla}) = \text{NLL}_t^{\text{method}} - \text{NLL}_t^{\text{vanilla}}$$

We find that global steering changes many non-critical tokens, while token-aware steering mainly affects visually important tokens. This shows that global steering misallocates the intervention budget and motivates adaptive token-level steering.



Step 1: Token-level extraction of steering vectors.

We compare token probabilities with and without the image:

$$\Delta_t = \log p(y_t | x, I, y_{<t}) - \log p(y_t | x, \emptyset, y_{<t}). \quad \mathcal{T}^+ = \text{TOP}_p(\{|\Delta_t|\}_{t \in \mathcal{T}}), \quad \mathcal{T}^- = \text{BOTTOM}_p(\{|\Delta_t|\}_{t \in \mathcal{T}}).$$

$$d^{(\ell)} = \frac{1}{|\mathcal{T}^+|} \sum_{t \in \mathcal{T}^+} \tilde{h}_t^{(\ell)} - \frac{1}{|\mathcal{T}^-|} \sum_{t \in \mathcal{T}^-} \tilde{h}_t^{(\ell)}.$$

Tokens with large ($|\Delta_t|$) are treated as visually sensitive tokens ($\mathcal{T}^+$), while tokens with small ($|\Delta_t|$) are treated as visually insensitive tokens ($\mathcal{T}^-$). For each layer, we contrast their normalized hidden states to obtain an initial steering direction:

$$d^{(\ell)} = \frac{1}{|\mathcal{T}^+|} \sum_{t \in \mathcal{T}^+} \tilde{h}_t^{(\ell)} - \frac{1}{|\mathcal{T}^-|} \sum_{t \in \mathcal{T}^-} \tilde{h}_t^{(\ell)}. \quad W_{\text{init}}^{(\ell)} = \text{PCA}_1(\{d_i^{(\ell)}\}_{i=1}^N), \quad W_{\text{init}}^{(\ell)} \leftarrow \frac{W_{\text{init}}^{(\ell)}}{\|W_{\text{init}}^{(\ell)}\|_2}.$$

Step 2: Supervised refinement of steering vectors

We then keep the LVLm frozen and only optimize the steering vectors ($W(\ell)$) using correction data:

$$\mathcal{L} = \mathcal{L}_{\text{NLL}} + \beta \mathcal{L}_{\text{KL}} + \gamma \mathcal{L}_{\text{prox}}. \quad \mathcal{L}_{\text{NLL}} = -\frac{1}{|\mathcal{A}|} \sum_{t \in \mathcal{A}} \log p_{\theta, W}(y_t | x, I, y_{<t}).$$

$$\mathcal{L}_{\text{KL}} = \frac{1}{|\mathcal{A}|} \sum_{t \in \mathcal{A}} \text{KL}(p_{\theta, W}(\cdot | x, I, y_{<t}) \| p_{\theta}(\cdot | x, I, y_{<t})). \quad \mathcal{L}_{\text{prox}} = \frac{1}{|\mathcal{S}|} \sum_{t \in \mathcal{S}} \|W^{(\ell)} - W_{\text{init}}^{(\ell)}\|_2^2.$$

This lightweight refinement improves the steering direction without updating the model parameters.

Step 3: Adapt steering during decoding.

At each decoding step, we measure visual sensitivity by comparing image-conditioned and image-ablated logits:

$$\text{VS}_t = \text{KL} \left(\text{softmax} \left(\frac{z_t^I}{T_{KL}} \right) \parallel \text{softmax} \left(\frac{z_t^{\emptyset}}{T_{KL}} \right) \right), \quad \text{and ap} \quad \bar{\text{VS}}_t = \frac{\text{VS}_t - \mu}{\sigma + \epsilon}.$$

We then map (VS

$$\tilde{\lambda}_t = \lambda_{\min} + (\lambda_{\max} - \lambda_{\min}) g_t. \quad \hat{\lambda}_t = \alpha \hat{\lambda}_{t-1} + (1 - \alpha) \tilde{\lambda}_t. \quad \lambda_t = \min(\hat{\lambda}_t, \lambda_t^{\text{cap}}).$$

Thus, TLVS applies stronger steering to visually important tokens and weaker steering to non-critical tokens.

Table 1. Main results. TLVS consistently improves hallucination-related metrics across POPE, HallusionBench, CHAIR (COCO), and AMBER on both LLaVA-1.5-7B and Qwen2.5-VL-7B. $\uparrow/\downarrow$ indicate that higher/lower is better; **bold/underline** mark the best/second-best within each backbone; gray highlights TLVS, with $\uparrow/\downarrow$ showing absolute gains over the vanilla baseline.

Model	POPE				HallusionBench			CHAIR (COCO)			AMBER							
	Adv.↑	Pop.↑	Rand.↑	Avg.↑	qAcc↑	fAcc↑	aAcc↑	CHAIRs↓	CHAIRi↓	Recall↑	CHAIR↓	Cover↑	Hal↓	Cog↓	F1-E↑	F1-A↑	F1-R↑	F1↑
LLaVA-1.5-7B	81.80	84.36	89.12	85.09	13.19	16.18	45.70	46.0	12.8	78.9	12.0	50.3	36.4	4.6	83.2	65.6	62.4	74.7
+ VCD	81.33	85.06	87.16	84.52	12.75	19.08	46.41	47.8	14.1	82.7	10.1	51.2	43.6	4.3	84.1	63.6	66.4	74.8
+ SPARC	81.03	86.05	89.00	85.36	13.19	17.34	35.25	43.4	16.8	61.1	14.5	44.6	<u>23.1</u>	2.4	88.5	66.7	65.5	77.9
+ ASD	—	—	—	<u>87.87</u>	—	—	—	40.0	11.3	82.0	—	—	—	—	—	—	—	—
+ SHARP	79.00	83.20	85.30	82.50	—	—	—	<u>34.8</u>	<u>10.6</u>	59.7	8.5	52.1	39.2	4.8	—	—	—	—
+ VTI	80.43	85.53	88.50	84.82	13.19	15.90	44.11	35.8	11.1	76.8	4.7	47.2	27.0	<u>1.8</u>	85.8	67.5	<u>68.3</u>	77.1
+ VISTA	80.98	<u>88.87</u>	<u>89.25</u>	86.37	14.29	19.36	45.82	44.6	13.4	77.0	6.0	39.9	27.1	2.4	87.6	67.3	55.9	77.6
+ DPO (RLHF-V)	<u>84.03</u>	<u>85.94</u>	89.12	<u>86.36</u>	16.70	20.81	51.31	43.2	11.9	76.4	5.7	49.7	27.3	2.6	<u>90.7</u>	72.6	64.6	80.9
+ TLVS (ours)	87.85 $\uparrow_{6.8}$	89.42 $\uparrow_{3.6}$	92.08 $\uparrow_{2.9}$	89.78 $\uparrow_{4.6}$	15.83 $\uparrow_{2.6}$	19.95 $\uparrow_{3.7}$	46.50 $\uparrow_{0.8}$	31.2 $\downarrow_{4.8}$	9.7 $\downarrow_{1.1}$	82.2 $\uparrow_{3.3}$	3.8 $\downarrow_{4.2}$	51.8 $\uparrow_{1.5}$	19.7 $\downarrow_{16.7}$	1.5 $\downarrow_{3.1}$	91.5 $\uparrow_{3.3}$	73.0 $\uparrow_{7.4}$	69.7 $\uparrow_{1.3}$	82.2 $\uparrow_{1.5}$
Qwen2.5-VL-7B	84.92	86.78	89.01	86.90	23.74	31.21	55.54	24.0	7.0	86.7	5.4	51.6	29.0	1.9	97.3	83.8	<u>72.6</u>	89.4
+ VCD	83.92	85.80	88.41	86.04	31.43	35.26	61.82	29.8	9.7	58.0	6.2	50.3	36.5	2.1	91.6	78.6	68.0	83.8
+ DPO (RLHF-V)	<u>84.99</u>	<u>86.81</u>	<u>89.72</u>	<u>87.17</u>	34.95	39.60	62.62	13.4	3.5	62.7	3.1	50.7	21.9	<u>1.1</u>	98.5	82.5	65.9	88.3
+ TLVS (ours)	85.10 $\uparrow_{0.11}$	87.01 $\uparrow_{0.21}$	89.88 $\uparrow_{0.67}$	87.33 $\uparrow_{0.43}$	33.08 $\uparrow_{0.34}$	37.19 $\uparrow_{1.98}$	63.23 $\uparrow_{0.8}$	12.0 $\downarrow_{12.0}$	5.2 $\downarrow_{1.8}$	83.9 $\downarrow_{2.8}$	3.8 $\downarrow_{1.6}$	51.2 $\downarrow_{0.4}$	16.7 $\downarrow_{12.3}$	1.0 $\downarrow_{0.9}$	97.3	82.7 $\downarrow_{1.1}$	73.0 $\uparrow_{0.4}$	89.8 $\uparrow_{0.4}$

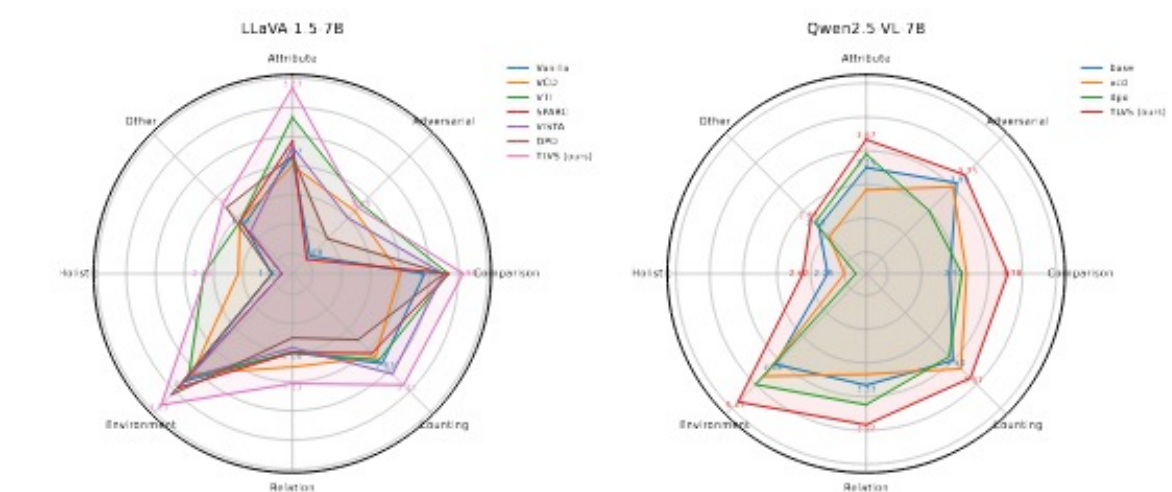


Figure 7. MMHal-Bench performance across question categories (attributes, adversarial objects, comparisons, counting, spatial, environmental, holistic, and others).

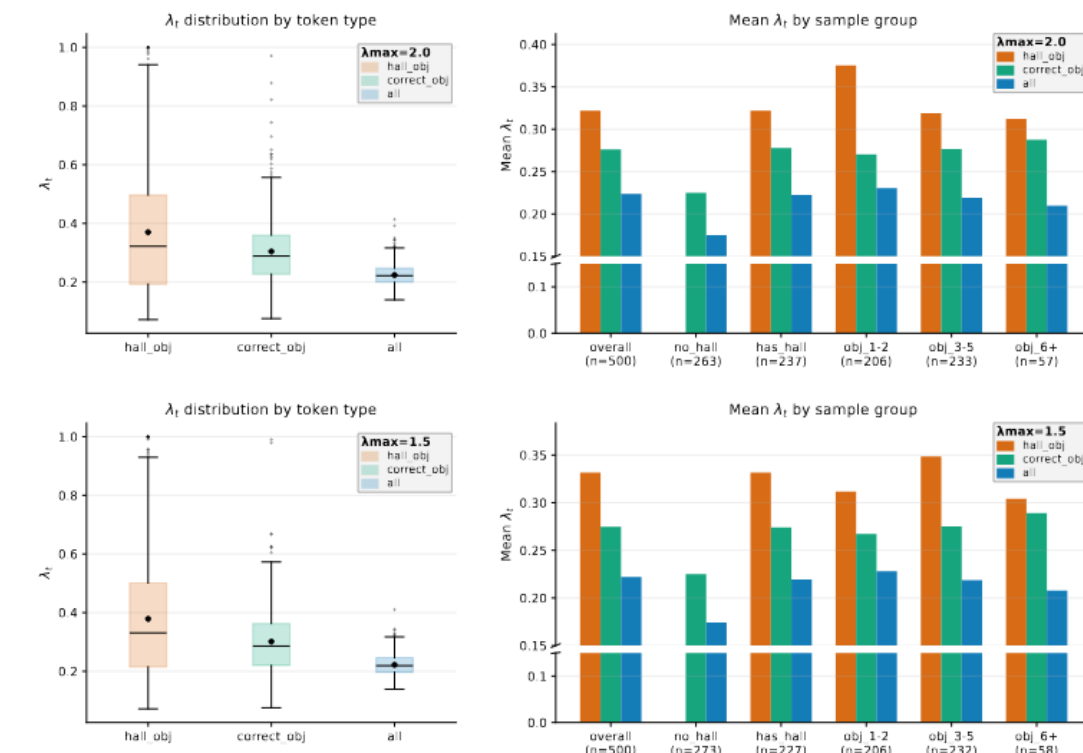


Figure 8. Adaptive steering focuses on critical tokens. λ_t distribution (left) and mean λ_t by token group (right).

Table 2. Ablation on refinement and token-level adaptive steering. $\uparrow/\downarrow$ indicate higher/lower is better; green/red arrows show absolute gains over the vanilla baseline within each backbone.

Method	Components		POPE		AMBER			
	Refine	Adapt.	Avg $\uparrow$	CHAIR $\downarrow$	Cover $\uparrow$	Hal $\downarrow$	F1 $\uparrow$	
LLaVA-1.5-7B								
Vanilla	—	—	85.09	12.0	50.3	36.4	74.7	
TLVS-Init	×	×	86.83 $\uparrow_{1.74}$	7.4 $\downarrow_{4.6}$	45.4 $\downarrow_{4.9}$	35.9 $\downarrow_{0.5}$	75.0 $\uparrow_{0.3}$	
TLVS-Refine	✓	×	88.71 $\uparrow_{3.62}$	4.1 $\downarrow_{7.9}$	44.5 $\downarrow_{5.8}$	28.1 $\downarrow_{8.3}$	80.8 $\uparrow_{6.1}$	
TLVS	✓	✓	89.78 $\uparrow_{4.69}$	3.8 $\downarrow_{4.2}$	51.8 $\uparrow_{1.5}$	19.7 $\downarrow_{16.7}$	82.2 $\uparrow_{7.5}$	
Qwen2.5-VL-7B								
Vanilla	—	—	86.90	5.4	51.6	29.0	89.4	
TLVS-Init	×	×	86.83 $\downarrow_{0.07}$	5.0 $\downarrow_{0.4}$	50.8 $\downarrow_{0.8}$	26.8 $\downarrow_{2.2}$	87.6 $\downarrow_{1.8}$	
TLVS-Refine	✓	×	87.00 $\uparrow_{0.10}$	4.2 $\downarrow_{1.2}$	48.9 $\downarrow_{2.7}$	20.0 $\downarrow_{9.0}$	88.8 $\downarrow_{0.6}$	
TLVS	✓	✓	87.33 $\uparrow_{0.43}$	3.8 $\downarrow_{1.6}$	51.2 $\downarrow_{0.4}$	16.7 $\downarrow_{12.3}$	89.8 $\uparrow_{0.4}$	

Table 3. Efficiency and Overhead. (i) Fwd/Token, the average number of forward passes per generated token; (ii) ms/Token, wall-clock latency per token; (iii) Sec/Sample, end-to-end generation time per sample; and (iv) Peak VRAM, the maximum GPU memory footprint during inference. Lower is better for all four metrics ($\downarrow$).

Method	Fwd/Token $\downarrow$	ms/Token $\downarrow$	Sec/Sample $\downarrow$	Peak VRAM (GB) $\downarrow$
Base	1.00	31.25	3.34	13.95
TLVS	2.00	64.01	6.79	14.03
Overhead (Ours/Base)	2.00×	2.05×	2.04×	1.01×