NOT ALL TOKENS CONTRIBUTE EQUALLY: TOKEN-LEVEL REINFORCEMENT LEARNING FOR MULTIMODAL HALLUCINATION MITIGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning with verifiable rewards (RLVR) is increasingly used to reduce hallucination in multimodal large language models (MLLMs). A key limitation is that the RLVR reward is defined only at the sequence level, so every token in a response shares the same advantage, without explicitly distinguishing its contribution to the error. Two common failure modes motivate our approach. The model may attend insufficiently to the image and rely on language priors, or make a wrong choice at an uncertain reasoning step. Neither failure mode is directly observable, but the model provides two relevant internal signals during training. The visual attention ratio of a token provides a proxy for visual grounding, and the predictive entropy indicates where the model is uncertain. Building on these two signals, we take a token-level view of RLVR and propose **VIEW** (**V**isual-grounding and **I**nformation-**E**ntropy **W**eighting), which reshapes the learning signal at the token level. VIEW first rescales each token's advantage using visual attention, then focuses the policy gradient on high-entropy tokens where reasoning may fork. Both signals are used only during training, and VIEW leaves standard autoregressive inference unchanged. Extensive experiments across hallucination, visual-reasoning, and general multimodal benchmarks show that VIEW improves the Qwen3-VL-8B average over GRPO and PAPO.

1 INTRODUCTION

Multimodal large language models (MLLMs) have advanced rapidly and are now applied to tasks such as visual question answering and image understanding. They remain prone to hallucination, however, generating objects, attributes, or events that are absent from the image, which undermines their reliability in applications that require faithful answers (Li et al., 2023; Guan et al., 2024; Zhang et al., 2024a). Reducing hallucination is therefore essential for reliable MLLMs.

A growing body of work fine-tunes MLLMs with reinforcement learning from verifiable rewards (RLVR) (Shao et al., 2024; Yu et al., 2025; DeepSeek-AI, 2025), in which a rule-based reward favors correct answers and penalizes incorrect ones. Some approaches introduce visual rewards from judge models or perception checklists (Xiao et al., 2026; Zhang et al., 2025), while others add a perception loss or extra masked-image passes to encourage reliance on the image (Wang et al., 2025c; Huang et al., 2025). These methods are effective, yet the sequence-level reward alone does not distinguish individual token contributions. In standard GRPO/DAPO, a single advantage assigned to the entire response is shared across all tokens. As a result, tokens that faithfully describe the image and those that fabricate absent objects receive the same advantage despite contributing differently.

Our key observation is that not all tokens in an incorrect response contribute equally to the error. We focus on two common failure modes. In the first, the model pays too little attention to the image and relies on language priors, a source of object hallucination (Arif et al., 2025; Sun et al., 2026). In the second, the model reads the image correctly but makes a wrong choice at an uncertain reasoning step, and this error spreads through the rest of the response (Wang et al., 2025b). Handling them therefore calls for a token-level signal that separates tokens by how they behave, rather than a uniform reward applied to the entire response.

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.

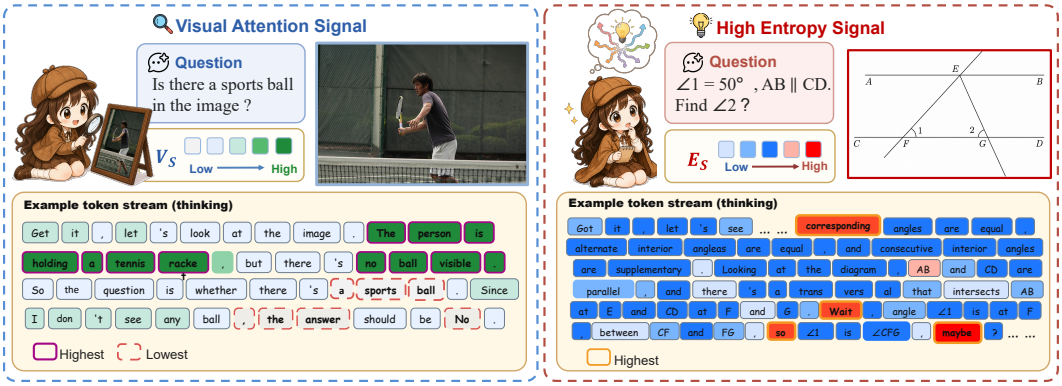

Figure 1: Two internal signals for credit assignment, shown on model thinking traces. **Left:** visual attention (V_S). The visual description receives higher attention than the repeated question about a sports ball. **Right:** predictive entropy (E_S). Uncertain reasoning steps show spikes in entropy.

Two internal signals provide token-level information that can be recomputed during training. A token's *visual attention ratio* (V_S), the fraction of its decoder attention directed to image positions, serves as a proxy for visual grounding. Figure 1 (left) illustrates this variation: the visual description receives higher attention than the repeated question about a sports ball. Uncertainty is captured by the *predictive entropy* (E_S) of the next-token distribution, which can peak at forking tokens where the model hesitates between possible continuations (Wang et al., 2025b). In Figure 1 (right), entropy rises sharply at the step where the model reconsiders its answer. Together, the two signals provide complementary cues for token-level credit assignment.

Building on these two signals, we take a token-level *view* of RLVR, which we implement as **VIEW** (**V**isual-grounding and **I**nformation-**E**ntropy **W**eighting), a method that reshapes the learning signal at the token level. VIEW introduces two token-level modifications to the standard policy-gradient objective. *Visual-Attention Grounding* (VAG) rescales each token's advantage by its visual attention ratio, assigning greater relative weight to tokens with stronger visual attention. *Entropy-Guided Refinement* (EGR) restricts the policy gradient to the top-ρ highest-entropy forking tokens, focusing learning on uncertain decisions. The two modifications are applied in two successive phases, grounding first and refinement second, to avoid restricting attention-based weighting to a small high-entropy subset where its variation may be limited (Section 4.4). Both signals are used only during training, so VIEW leaves standard autoregressive inference unchanged.

We make the following contributions.

- **A token-level view of multimodal hallucination.** We consider insufficient visual grounding and uncertain reasoning as complementary failure modes, and use their proxy signals for per-token credit assignment (Sections 3 and 4.2).
- **A token-level training method.** VIEW uses the model's own visual-attention and predictive-entropy signals, requiring no external model or change at inference (Sections 4.2 and 4.3).
- **Consistent gains.** VIEW improves the controlled average over GRPO and PAPO, and the ablations isolate both signals and their ordering (Section 5).

2 Related Work

Hallucination Mitigation in MLLMs. A persistent weakness of MLLMs is their tendency to assert objects, attributes, or events that the visual input does not support, a behavior probed systematically by recent benchmarks (Li et al., 2023; Guan et al., 2024; Zhang et al., 2024a). Existing remedies occupy a broad cost spectrum. Data-centric alignment intervenes at the source through corrective supervision, as in RLHF-V (Yu et al., 2023) and fact-augmented preference optimization (Sun et al., 2024), though such approaches hinge on expensive human annotation. At lower cost, training-free decoding methods operate only at inference time. Visual Contrastive Decoding (Leng et al., 2024) and OPERA (Huang et al., 2024) recalibrate the output distribution of a frozen model, while a related line

of work redirects attention toward visual tokens during generation (Arif et al., 2025; Sun et al., 2026). A recurring observation across these attention-based studies is that hallucination can be associated with reduced attention to the image, motivating visual attention as a proxy for grounding. VIEW employs this proxy during training, converting per-token visual attention into an RL credit term that requires no additional supervision and leaves the inference path the same at test time.

Reinforcement Learning for MLLMs. Verifiable-reward reinforcement learning has become the standard route to stronger reasoning in large language models, with GRPO (Shao et al., 2024) and DAPO (Yu et al., 2025) as the prevailing algorithms (DeepSeek-AI, 2025), and these recipes extend to multimodal models with only minor modification. A central limitation lies in the granularity of the reward. A single sequence-level advantage is shared across tokens, without explicitly distinguishing whether an intermediate token is grounded in the visual input. Recent work addresses this limitation along two complementary directions. The first augments the reward with external perception signals, for instance, Perception-R1 (Xiao et al., 2026) scores responses with a judge model, and PEARL (Zhang et al., 2025) relies on a structured visual checklist. The second reshapes the optimization objective itself; a representative method is PAPO (Wang et al., 2025c), which introduces an implicit perception loss that penalizes a policy whose predictions remain unchanged once the visual evidence is removed, thereby encouraging greater reliance on the image. VIEW belongs to the second direction but differs from it in two respects. It derives the grounding signal directly from decoder attention; unlike PAPO's masked-image auxiliary objective and judge-based reward methods, it requires no extra masked-image pass or external perception judge. It also combines two distinct quantities, visual attention for grounding and predictive entropy for uncertainty, and applies them in separate stages so that attention weighting first operates over all response tokens (Section 4.4).

3 Preliminaries

Problem Setup. Given an image (or video) $\mathbf{I}$ and prompt $\mathbf{x}$, a multimodal policy π_θ generates $\mathbf{y} = (y_1, \ldots, y_T)$ and receives a rule-based correctness reward $R(\mathbf{y})$. We fine-tune π_θ to improve both accuracy and visual grounding.

Group Relative Policy Optimization (GRPO). GRPO (Shao et al., 2024) samples G responses per query and estimates the advantage without a value network:

$$\hat{A}^{(i)} = \frac{R(\mathbf{y}^{(i)}) - \mathrm{mean}(\{R_j\}_{j=1}^{G})}{\mathrm{std}(\{R_j\}_{j=1}^{G})}, \tag{1}$$

All tokens in response i share $\hat{A}^{(i)}$, and the policy maximizes the clipped surrogate

$$\mathcal{J}_{\mathrm{GRPO}}(\theta) = \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|\mathbf{y}^{(i)}|}\sum_{t=1}^{|\mathbf{y}^{(i)}|}\min\Big(r_{i,t}(\theta)\,\hat{A}^{(i)},\ \ \mathrm{clip}\big(r_{i,t}(\theta),\,1{-}\epsilon,\,1{+}\epsilon\big)\,\hat{A}^{(i)}\Big)\right], \tag{2}$$

where $r_{i,t}$ is the per-token importance ratio and ϵ the clipping bound. Equation (2) gives the symmetric GRPO form for exposition. The reported DAPO and VIEW runs use DAPO's asymmetric clipping and token-level normalization; these implementation settings are listed in Section B.2.

Decoupled Clip and Dynamic Sampling Policy Optimization (DAPO). DAPO (Yu et al., 2025) stabilizes GRPO at scale through asymmetric clipping, dynamic sampling, token-level loss normalization, and overlong-response filtering. VIEW retains DAPO's rollout, reward, and clipping logic and changes only token-level credit assignment.

Hallucination and Internal Signals. Sequence-level rewards do not identify which tokens cause a hallucination, but two internal signals recomputed during training expose a useful proxy for this structure (Figure 1). The *visual attention ratio* (V_S) measures how strongly a generated token attends to image positions: grounded tokens tend to attend more, while fabricated content relies more on language priors. The *predictive entropy* (E_S) peaks at uncertain *forking tokens*, where one choice can redirect the reasoning chain (Wang et al., 2025b). Thus, one signal measures whether a token uses the image and the other identifies where a decision is uncertain, motivating VIEW's non-uniform credit assignment.

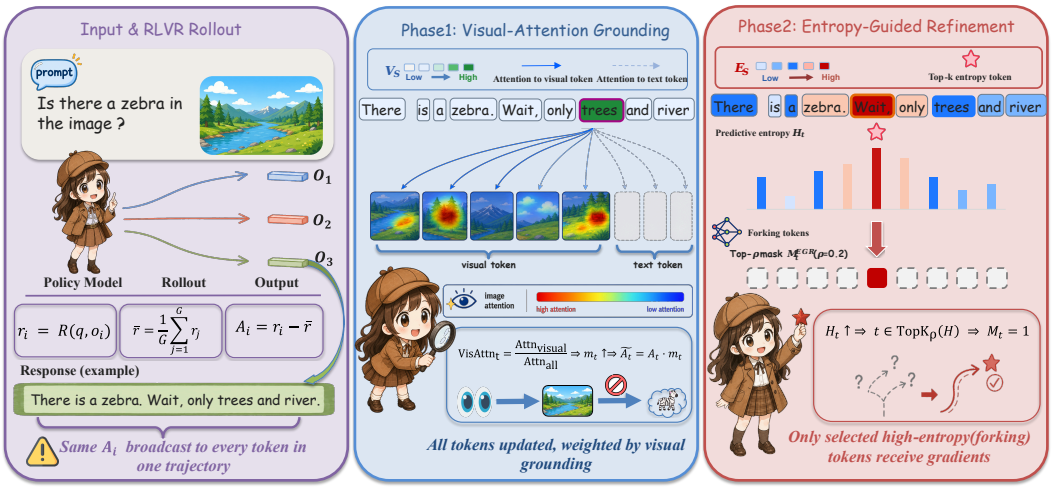

Figure 2: Overview of VIEW. **Left (input & rollout):** as in GRPO/DAPO, a single group-relative advantage $A_i = r_i - \bar{r}$ is broadcast to every token of a sampled trajectory, with no token-level notion of grounding or uncertainty. **Middle (Phase 1, VAG):** the visual attention ratio $\mathrm{VisAttn}_t = \mathrm{Attn}_{\mathrm{visual}}/\mathrm{Attn}_{\mathrm{all}}$ is mapped to a per-token factor m_t that reweights the advantage, $\tilde{A}_t = A_t\, m_t$; all tokens are still updated, but those grounded in the image receive a larger signal. **Right (Phase 2, EGR):** resuming from the VAG checkpoint, only the top-ρ high-entropy forking tokens (mask $\mathcal{M}_t^{\mathrm{EGR}} = 1$) receive policy gradients, concentrating refinement on the decisive steps.

4 Method: Visual-Grounding and Information-Entropy Weighting

Figure 2 illustrates the complete pipeline. VIEW introduces two token-level modifications to the policy-gradient objective, applied in two successive phases.

Notation. For brevity, A_t denotes the token-level advantage at response position t. Under GRPO/DAPO it equals the shared group-relative advantage $\hat{A}^{(i)}$ of the response (Eq. (1)). Let $\mathcal{L}_t^{\mathrm{clip}}(\theta; A_t)$ denote the per-token clipped surrogate term inside Eq. (2), that is $\min\big(r_t A_t,\ \mathrm{clip}(r_t, 1{-}\epsilon_{\mathrm{low}}, 1{+}\epsilon_{\mathrm{high}})A_t\big)$, so that $\mathcal{L}_{\mathrm{PG}} = \mathbb{E}[\frac{1}{T}\sum_t \mathcal{L}_t^{\mathrm{clip}}]$. VIEW changes *only* A_t (Phase 1, VAG) and the set of tokens that contribute to the gradient (Phase 2, EGR). The rollout, reward, importance ratio r_t, and clipping bounds stay the same as in DAPO.

4.1 Visual Attention Ratio

For each generated token y_t, we average the attention weights from the last L decoder layers across all H heads at every key position in the sequence:

$$\bar{\alpha}_{t,j} = \frac{1}{L \cdot H} \sum_{\ell=1}^{L} \sum_{h=1}^{H} \alpha_{t,j}^{(\ell,h)}, \tag{3}$$

where $\alpha_{t,j}^{(\ell,h)}$ is the attention weight from query position t to key position j in layer ℓ, head h. Let $\mathcal{K}_t$ denote the set of all key positions causally visible to token t, and $\mathcal{V} \subset \mathcal{K}_t$ the subset corresponding to visual tokens (image or video patch embeddings). The **Visual Attention Ratio** is the share of this averaged attention that goes to visual tokens:

$$\mathrm{VisAttn}_t = \frac{\sum_{j \in \mathcal{V} \cap \mathcal{K}_t} \bar{\alpha}_{t,j}}{\sum_{j \in \mathcal{K}_t} \bar{\alpha}_{t,j} + \epsilon}. \tag{4}$$

Here ϵ is a small constant for numerical stability. The ratio $\mathrm{VisAttn}_t \in [0, 1]$ serves as a per-token proxy for visual grounding. A larger value indicates a greater share of direct attention to visual

tokens. In the implementation, the training signal uses the post-token query at $q(t)$; the strict prediction-aligned alternative $q(t-1)$ is evaluated in the proxy audit in Section D.

4.2 Phase 1: Visual-Attention Grounding (VAG)

The visual attention ratio scales the per-token advantage, giving tokens with higher visual attention greater relative weight. Within each response, we apply *min–max normalization*:

$$s_t = \frac{\mathrm{VisAttn}_t - \min_{t' \in \mathcal{T}} \mathrm{VisAttn}_{t'}}{\max_{t' \in \mathcal{T}} \mathrm{VisAttn}_{t'} - \min_{t' \in \mathcal{T}} \mathrm{VisAttn}_{t'} + \epsilon}, \tag{5}$$

where $\mathcal{T} = \{1, \ldots, T\}$ indexes the response tokens; a flat attention sequence uses $s_t = 0.5$.

The normalized score is turned into a scaling factor by a simple linear map:

$$m_t = \beta_{\text{base}} + \beta_{\text{mod}} \cdot s_t, \tag{6}$$

and the modulated advantage is:

$$\tilde{A}_t \;=\; A_t \cdot m_t. \tag{7}$$

Here β_{base} sets the floor and β_{mod} the visual-grounding range.

4.3 Phase 2: Entropy-Guided Refinement (EGR)

Token-Level Predictive Entropy. During the actor update, we recompute the Shannon entropy of the current-policy next-token distribution:

$$H_t = -\sum_{v \in \mathcal{V}_{\text{vocab}}} p_\theta\big(y_t = v \mid \mathbf{I}, \mathbf{x}, \mathbf{y}_{<t}\big)\, \log\, p_\theta\big(y_t = v \mid \mathbf{I}, \mathbf{x}, \mathbf{y}_{<t}\big), \tag{8}$$

where $\mathcal{V}_{\text{vocab}}$ denotes the full vocabulary.

Forking-Token Selection. High values of H_t mark positions where the model hesitates between several likely continuations. Following the forking-token view of Wang et al. (2025b), we treat these positions as *decision forks*, where the choice has a large effect on whether the output stays factual. Within each sequence, we keep the top-ρ fraction of tokens ranked by entropy:

$$k = \max\big(1,\ \lceil \rho \cdot T \rceil\big), \qquad \mathcal{S}_{\text{EGR}} = \mathrm{TopK}\big(\{H_t\}_{t=1}^{T},\ k\big), \tag{9}$$

and define the binary EGR mask:

$$\mathcal{M}_t^{\text{EGR}} = \mathbf{1}\big[t \in \mathcal{S}_{\text{EGR}}\big]. \tag{10}$$

Masked Policy Gradient. The EGR loss limits the gradient to the masked subset and normalizes by the mask count:

$$\mathcal{L}_{\text{EGR}} \;=\; \mathbb{E}\left[\frac{\displaystyle\sum_{t=1}^{T} \mathcal{M}_t^{\text{EGR}} \cdot \mathcal{L}_t^{\text{clip}}\big(\theta;\, A_t\big)}{\displaystyle\sum_{t=1}^{T} \mathcal{M}_t^{\text{EGR}} + \epsilon}\right]. \tag{11}$$

In the implementation, the expectation is evaluated with token-average aggregation in each policy microbatch (`loss_avg_mode=token`): the masked policy terms are averaged over active response tokens, and the microbatch loss is scaled by its response-token count during the update. Thus the displayed per-sequence form is a compact description of the mask and does not imply an equal-weight average over responses. By focusing the gradient on the top-ρ high-entropy positions ($\rho = 0.2$), EGR concentrates learning on uncertain tokens.

4.4 TWO-PHASE VAG→EGR TRAINING

Grounding-First Protocol. VIEW trains for one epoch per phase (Algorithm 1).

Phase 1 (VAG) applies m_t to all response-token advantages, encouraging relative emphasis on tokens with stronger visual attention.

Phase 2 (EGR) resumes from the Phase 1 checkpoint, disables modulation, and focuses the gradient on the top-ρ high-entropy tokens where the model still hesitates.

Algorithm 1: VIEW: Two-phase VAG→EGR training

Require: Policy π_θ; ρ; $\beta_{\text{base}}, \beta_{\text{mod}}$

Phase 1: VAG

for each training step **do**
  Sample rollouts; compute $\{A_t\}$ and VisAttn_t
  Normalize to s_t; $m_t \leftarrow \beta_{\text{base}} + \beta_{\text{mod}} s_t$
  Update θ with $\mathcal{L}_{\text{PG}}(A_t m_t)$ on all tokens
end for
Save checkpoint $\theta^{(1)}$

Phase 2: EGR, resume from $\theta^{(1)}$

for each training step **do**
  Compute $\{A_t\}$ and $\{H_t\}$
  $\mathcal{M}_t^{\text{EGR}} \leftarrow \mathbf{1}[t \in \text{TopK}_\rho(\{H_t\})]$
  Update θ with $\mathcal{L}_{\text{EGR}}(A)$ on masked tokens
end for

Rationale for Staging. When both are applied together, the effective loss becomes:

$$\mathcal{L}^{\text{joint}} \propto \sum_t \mathcal{M}_t^{\text{EGR}} \cdot \mathcal{L}_t^{\text{clip}}\big(\theta;\, A_t \cdot m_t\big). \tag{12}$$

The EGR mask restricts attention-based weighting to a small subset ($\rho = 0.2$); its effect then depends on how much the scaling factors vary within that subset. In logged training batches, the variance of m_t inside the EGR-selected subset is lower than over the full sequence, so the scaling can act as a near-constant factor. We use this as a qualitative motivation for staging, not as a separately estimated effect size. In this case, Eq. (12) reduces to

$$\mathcal{L}^{\text{joint}} \approx \bar{m} \cdot \sum_t \mathcal{M}_t^{\text{EGR}} \cdot \mathcal{L}_t^{\text{clip}}\big(\theta;\, A_t\big), \tag{13}$$

where $\bar{m}$ is nearly constant. In this regime, attention-based weighting provides little differentiation among the selected tokens. Two-phase training avoids this by assigning each phase the token set it is designed for. VAG scales all tokens in Phase 1, and EGR keeps a small subset of tokens in Phase 2, *after* the model already grounds well in the image. The ordering ablation examines the grounding-first design (Section 5.4).

5 EXPERIMENTS

5.1 SETUP

Model and training. We use Qwen3-VL-8B-Thinking (Bai et al., 2025a) as the base MLLM and train VIEW with the DAPO (Yu et al., 2025) loop and a GRPO (Shao et al., 2024) advantage estimator. VIEW assigns token-level credit by rescaling the advantage with VAG and masking the policy gradient with EGR (Section 4). The Qwen3 runs share the rule-based correctness reward, $n = 5$ rollouts per prompt at temperature 1.0, a rollout batch of 256 prompts with a policy batch of 128, and a frozen vision tower with the language backbone fine-tuned by AdamW (learning rate $1{\times}10^{-6}$) in bf16. Each baseline retains its documented GRPO or DAPO loop; VIEW adds token credit on the DAPO-style loop. The exact clipping, filtering, and KL flags are listed in the reproducibility appendix. All methods are trained on the same data, a curated 36.6k subset of ViRL39K (Wang et al., 2025a) with rule-verifiable answers spanning mathematical, scientific, chart, and general visual reasoning. Each run takes 2 epochs (258 steps) on a single node with 8 NVIDIA A100 GPUs.

VIEW configuration. The visual attention ratio is read from the last $L = 4$ decoder layers and normalized within each sequence (Eq. (5)). VAG uses $\beta_{\text{base}} = \beta_{\text{mod}} = 0.5$, so the scaling factor lies in $m_t \in [0.5, 1.0]$, and EGR keeps the top $\rho = 0.2$ of tokens by entropy within each sequence. Phase 1 applies VAG to all response tokens for one epoch. Phase 2 resumes from that checkpoint for the second epoch with VAG disabled and EGR active.

Table 1: Main results. Best and second-best scores within each model group are **bold** and underlined. The Qwen3 same-backbone group uses the same base model and data; △ is the gain over the untrained base. Qwen2.5 baselines use public checkpoints, and MathVerse denotes MathVerse-VD.

	Hallucination			Visual Reasoning		General		
Model	POPE	HallusionB.	VideoHal.	MathVerse	MMK12	MME	MVBench	**Avg.**
Open-source MLLMs								
Qwen3-VL-30B-A3B-Thinking	89.03	**65.54**	76.65	69.42	79.05	88.25	69.68	76.80
Qwen3-VL-32B-Thinking	88.89	64.92	**77.74**	**69.67**	**80.35**	**89.55**	**70.16**	**77.33**
InternVL3.5-8B	89.46	58.99	71.70	46.83	68.70	87.07	68.39	70.16
InternVL3.5-14B	87.29	55.80	70.96	45.30	65.90	86.82	59.33	67.34
InternVL3.5-38B	**90.38**	46.50	75.26	50.25	71.50	87.24	60.06	68.74
RL Fine-tuning on Qwen2.5-VL-7B								
Qwen2.5-VL-7B-Instruct	88.08	43.93	72.83	41.62	52.30	86.18	50.47	62.20
Vision-R1	88.54	**55.89**	69.39	43.78	51.90	85.17	47.58	63.18
OpenVLThinker	83.90	49.51	70.43	45.05	54.65	81.84	42.82	61.17
VIEW (Qwen2.5-VL-7B)	**88.67**	45.26	**73.39**	**49.49**	**70.85**	**86.69**	**65.00**	**68.48**
Our Method (Qwen3-VL-8B-Thinking)								
Qwen3-VL-8B-Thinking	88.19	55.62	76.35	59.90	68.25	85.59	67.42	71.62
+ GRPO	88.48	57.84	76.43	66.62	69.00	87.99	68.92	73.61
+ PAPO-G	88.24	56.95	76.48	63.32	73.15	87.49	68.68	73.47
+ PAPO-D	88.52	55.80	76.30	63.07	**75.85**	88.16	68.61	73.76
+ **VIEW (ours)**	**88.80**	**58.10**	**76.61**	**67.13**	74.65	**88.50**	**69.05**	**74.69**
△ vs. Qwen3-VL-8B-Thinking	+0.61	+2.48	+0.26	+7.23	+6.40	+2.91	+1.63	+3.07

5.2 Benchmarks and Baselines

Benchmarks. We evaluate on seven benchmarks in three families. POPE (Li et al., 2023), HallusionBench (Guan et al., 2024), and VideoHallucer (Wang et al., 2024) measure hallucination. MathVerse (Zhang et al., 2024b), evaluated on its vision-dominant split MathVerse-VD, and MMK12 (Meng et al., 2025) measure visual reasoning. MME (Fu et al., 2023) and MVBench (Li et al., 2024) measure general multimodal understanding. All of them use images as input except VideoHallucer and MVBench, which use videos. This mix evaluates hallucination reduction alongside visual reasoning and video understanding. Judged benchmarks follow their official protocols; in particular, HallusionBench is scored by the official GPT judge (gpt-4o). The sensitivity study additionally reports MMStar (Chen et al., 2024) as an extra diagnostic benchmark; it is not included in the seven-benchmark averages or the main comparison table.

Baselines. We compare against the untrained base model and three RL methods trained on the same data. GRPO (Shao et al., 2024) provides a reference with uniform token credit within each response. PAPO (Wang et al., 2025c) adds an implicit perception loss on top of GRPO (PAPO-G) or DAPO (PAPO-D), allowing comparison with a method that uses an additional perception objective. Plain DAPO is the training reference in the sensitivity and phase analyses, where it is reported separately from the main benchmark comparison.

5.3 Main Results

Table 1 groups results by backbone and comparison setting. In the *Our Method* group, RL methods share the Qwen3-VL-8B-Thinking backbone, training data, and budget, while retaining their documented optimizer, clipping, filtering, and KL settings (Section B). The Qwen2.5-VL group tests transfer using publicly released baselines, and the open-source group provides reference models at several sizes. The average is the unweighted mean of the seven benchmark percentages.

Results on Qwen3-VL-8B-Thinking. Among methods trained from the same base model, VIEW has the best average, improving Qwen3-VL-8B-Thinking by 3.1 points and GRPO by 1.1, and ranks first on six of seven benchmarks. Relative to the untrained base, gains are positive everywhere and largest on visual reasoning (MathVerse +7.2, MMK12 +6.4), followed by MME (+2.9) and HallusionBench (+2.5). Against GRPO, gains align with VIEW's targets: MMK12 (+5.7 to 74.7)

and MathVerse (+0.5) are consistent with EGR's reasoning refinement, while HallusionBench and POPE (+0.3 each) are consistent with VAG's grounding. VIEW's only loss is MMK12, where PAPO-D leads by 1.2 points. PAPO-D adds an implicit perception loss from masked-image passes, which may explain its edge on diagram-heavy K-12 science. VIEW closes most of the gap without this extra supervision and leads PAPO-D on the other six benchmarks and on average (74.69 versus 73.76), consistent with decoder attention serving as a grounding-sensitive proxy.

Comparison with other methods. We also train VIEW with the same token-credit recipe on Qwen2.5-VL-7B (Bai et al., 2025b); the other rows are publicly released reference checkpoints. VIEW raises the instruct average from 62.20 to 68.48 (+6.28) and ranks first on six of seven benchmarks. Gains concentrate on visual reasoning (MMK12 +18.6, MathVerse +7.9) and video understanding (MVBench +14.5), showing transfer across backbone generations and task families. HallusionBench is the exception: Vision-R1 (55.89) and OpenVLThinker (49.51) score higher. Both distill long reasoning traces from stronger models before RL, which may contribute to these scores. VIEW uses no distillation data yet improves HallusionBench (43.93 → 45.26) and retains gains elsewhere.

Comparison with larger open-source models. Despite far fewer parameters, 8B VIEW remains competitive with larger open-source MLLMs. Its average of 74.69 exceeds every InternVL3.5 row, including the 38B model (68.74), which it beats on six of seven benchmarks. Qwen3-VL-Thinking 30B-A3B and 32B remain ahead on average (76.80 and 77.33), but VIEW beats the 30B-A3B model on MME and nearly ties it on POPE and VideoHallucer.

5.4 Qualitative and Ablation Analysis

Figure 3 illustrates answer revisions. VIEW distinguishes a bench from a chair and compares the green circles without following the surrounding ring. In each case, it revisits the image at a high-entropy *wait* token.

Table 2 examines VAG normalization, EGR alone, and the two phase orders, separating component gains from the effect of staging.

Table 2: Core ablation on Qwen3-VL-8B-Thinking.

Variant	MME	MathVerse
Baseline	85.59	59.90
VAG (batch norm)	87.49	65.99
VAG (sequence norm)	87.57	65.36
EGR only (top-20%)	**88.54**	65.36
EGR→VAG (reversed)	87.07	64.85
VIEW (VAG→EGR)	88.50	**67.13**

Both VAG variants improve MathVerse over the baseline. Batch normalization gives the higher MathVerse score, while sequence normalization is slightly better on MME. Adding EGR after VAG raises MathVerse to 67.13, while MME is only 0.04 below the best EGR-only score. Reversing the phases lowers MME and MathVerse by 1.43 and 2.28 points relative to VIEW, supporting the grounding-first ordering (Section 4.4).

5.5 Hyperparameter Sensitivity and Training Dynamics

Table 3 varies the base/modulation split μ and entropy selection ratio ρ. The equal split performs best on both benchmarks. The ρ sweep uses DAPO with EGR and no VAG; $\rho = 0$ denotes DAPO with EGR disabled. Both scores peak at $\rho = 0.2$ among the tested values.

Table 3: Hyperparameter sensitivity on Qwen3-VL-8B-Thinking. Best results are **bold**; * marks the VIEW setting.

μ (base/mod)	POPE	MMStar	ρ	POPE	MMStar
0.3/0.7	88.16	72.67	0.0	88.17	72.67
0.5/0.5*	**88.80**	**73.07**	0.1	88.74	72.87
0.7/0.3	87.61	72.20	0.2*	**88.79**	**73.60**
1.0/0.0	88.70	72.33	0.3	88.38	71.93
			0.4	88.33	73.07

Across the four μ settings, scores span 1.19 points on POPE and 0.87 on MMStar. Removing modulation (1.0/0.0) leaves POPE near its peak but reduces MMStar by 0.74 points. In the EGR sweep, increasing ρ from 0.2 to 0.3 lowers MMStar from 73.60 to 71.93; selecting more tokens

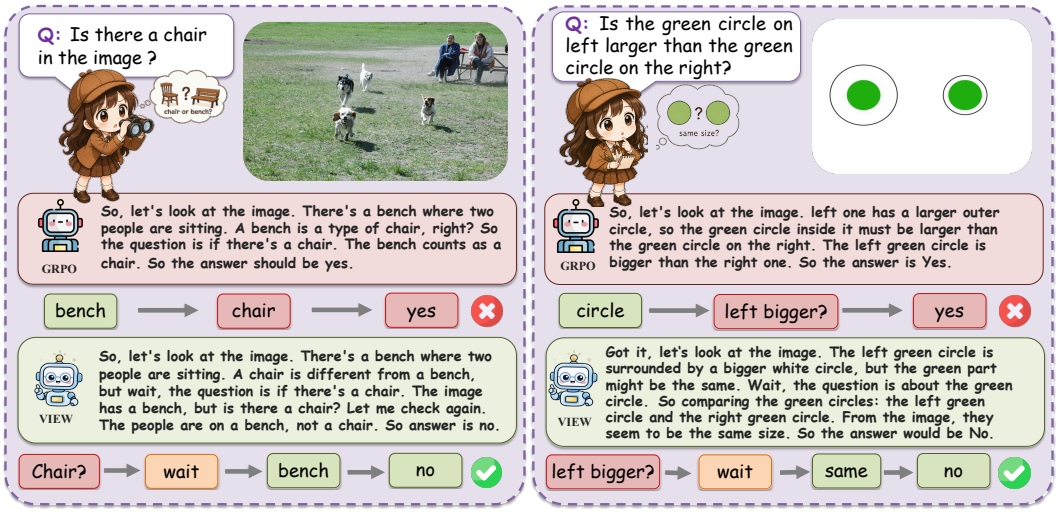

Figure 3: Qualitative comparison of GRPO and VIEW. VIEW revisits the image and reconsiders at a *wait* token, illustrating the behavior targeted by VAG and EGR.

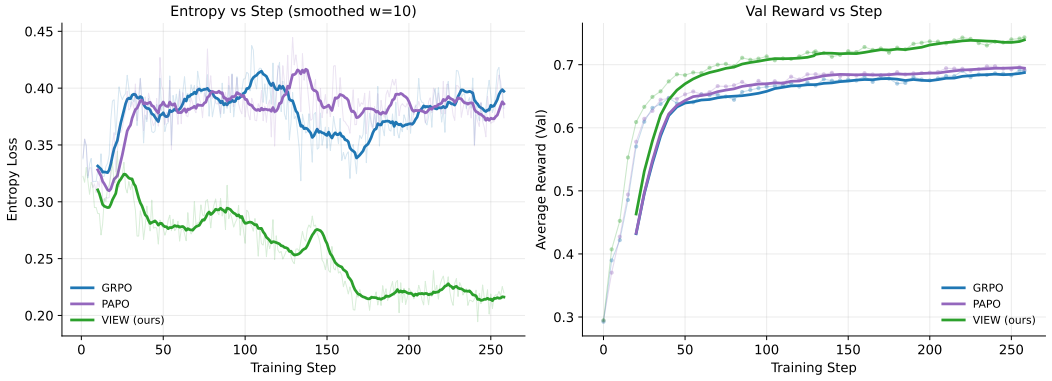

Figure 4: Training dynamics over 258 steps on Qwen3-VL-8B-Thinking. **Left:** actor entropy (smoothed, window 10). **Right:** mean validation reward. PAPO denotes PAPO-G.

does not consistently improve performance. These results favor concentrating updates on a small high-entropy subset in this setting.

Training dynamics. Figure 4 shows different training trajectories. The actor-entropy metric rises from about 0.34 to 0.38 for GRPO and PAPO, whereas VIEW's falls from about 0.32 to 0.22 across the VAG and EGR phases. VIEW's validation reward moves ahead early and ends near 0.74. The joint trend links lower entropy with better held-out reward, without establishing a causal relationship; logging details are given in Section C.

6 Conclusion

This paper starts from the view that not all tokens contribute equally to the error. Insufficient visual grounding and uncertain reasoning motivate two complementary token-level signals. VIEW turns this view into a token-level reinforcement learning framework that weights credit using two internal signals recomputed during training. Visual-Attention Grounding (VAG) rescales the RL advantage of each token by its visual attention, encouraging relative emphasis on tokens that use more visual information. Entropy-Guided Refinement (EGR) focuses policy gradients on the high-entropy tokens where the model remains uncertain about the next continuation.

AI USE STATEMENT

In accordance with ICLR 2027 policy, we report the use of AI models in this work. The AI models used for dataset construction and evaluation are described in the main paper. Beyond these methodological uses, AI assistance during manuscript preparation was strictly limited to enhancing linguistic quality, including improving sentence structure and ensuring grammatical correctness. The entirety of the scientific contributions in this paper is attributable solely to the authors. The final manuscript has been critically reviewed by all authors, who collectively assume full responsibility for its accuracy and integrity.

REFERENCES

Kazi Hasan Ibn Arif, Sajib Acharjee Dip, Khizar Hussain, Lang Zhang, and Chris Thomas. Fixing imbalanced attention to mitigate in-context hallucination of large vision-language models. *arXiv preprint arXiv:2501.12206*, 2025.

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, Yuanzhi Zhu, and Ke Zhu. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 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. *arXiv preprint arXiv:2502.13923*, 2025b.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. Are we on the right way for evaluating large vision-language models? *arXiv preprint arXiv:2403.20330*, 2024.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. MME: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, Dinesh Manocha, and Tianyi Zhou. Hallusionbench: An advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. *arXiv preprint arXiv:2310.14566*, 2024.

Qidong Huang, Xiaoyi Dong, Pan Zhang, Bin Wang, Conghui He, Jiaqi Wang, Dahua Lin, Weiming Zhang, and Nenghai Yu. OPERA: Alleviating hallucination in multi-modal large language models via over-trust penalty and retrospection-allocation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Siyuan Huang, Xiaoye Qu, Yafu Li, Yun Luo, Zefeng He, Daizong Liu, and Yu Cheng. Spotlight on token perception for multimodal reinforcement learning. *arXiv preprint arXiv:2510.09285*, 2025.

Sicong Leng, Hang Zhang, Guanzheng Chen, Xin Li, Shijian Lu, Chunyan Miao, and Lidong Bing. Mitigating object hallucinations in large vision-language models through visual contrastive decoding. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13872–13882. IEEE, 2024.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, Limin Wang, and Yu Qiao. MVBench: A comprehensive multi-modal video understanding benchmark. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

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

Fanqing Meng, Lingxiao Du, Zongkai Liu, Zhixiang Zhou, Quanfeng Lu, Daocheng Fu, Botian Shi, Wenhai Wang, Junjun He, Kaipeng Zhang, et al. Mm-eureka: Exploring visual aha moment with rule-based large-scale reinforcement learning. *arXiv preprint arXiv:2503.07365*, 3, 2025.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Yongkang Li, Yu Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Han Sun, Qin Li, Peixin Wang, and Min Zhang. Mitigating object hallucination in LVLMs via attention imbalance rectification. *arXiv preprint arXiv:2603.24058*, 2026.

Zhiqing Sun, Sheng Shen, Shengcao Cao, Haotian Liu, Chunyuan Li, Yikang Shen, Chuang Gan, Liangyan Gui, Yu-Xiong Wang, Yiming Yang, et al. Aligning large multimodal models with factually augmented rlhf. In *Findings of the association for computational linguistics: ACL 2024*, pp. 13088–13110, 2024.

Haozhe Wang, Chao Qu, Zuming Huang, Wei Chu, Fangzhen Lin, and Wenhu Chen. VL-Rethinker: Incentivizing self-reflection of vision-language models with reinforcement learning. *arXiv preprint arXiv:2504.08837*, 2025a.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xiong-Hui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for LLM reasoning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025b. arXiv:2506.01939.

Yuxuan Wang, Yueqian Wang, Dongyan Zhao, Cihang Xie, and Zilong Zheng. VideoHallucer: Evaluating intrinsic and extrinsic hallucinations in large video-language models. *arXiv preprint arXiv:2406.16338*, 2024.

Zhenhailong Wang, Xuehang Guo, Sofia Stoica, Haiyang Xu, Hongru Wang, Hyeonjeong Ha, Xiusi Chen, Yangyi Chen, Ming Yan, Fei Huang, and Heng Ji. Perception-aware policy optimization for multimodal reasoning. *arXiv preprint arXiv:2507.06448*, 2025c.

Tong Xiao, Xin Xu, Zhenya Huang, Hongyu Gao, Quan Liu, Qi Liu, and Enhong Chen. Perception-R1: Advancing multimodal reasoning capabilities of MLLMs via visual perception reward. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2506.07218.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Weinan Dai, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Tianyu Yu, Yuan Yao, Haoye Zhang, Taiwen He, Yifeng Han, Ganqu Cui, Jinyi Hu, Zhiyuan Liu, Hai-Tao Zheng, Maosong Sun, and Tat-Seng Chua. RLHF-V: Towards trustworthy MLLMs via behavior alignment from fine-grained correctional human feedback. *arXiv preprint arXiv:2312.00849*, 2023.

Chi Zhang, Haibo Qiu, Qiming Zhang, Yufei Xu, Zhixiong Zeng, Siqi Yang, Peng Shi, Lin Ma, and Jing Zhang. Perceptual-evidence anchored reinforced learning for multimodal reasoning. *arXiv preprint arXiv:2511.18437*, 2025.

Jiacheng Zhang, Yang Jiao, Shaoxiang Chen, Na Zhao, Zhiyu Tan, Hao Li, Xingjun Ma, and Jingjing Chen. Eventhallusion: Diagnosing event hallucinations in video llms. *arXiv preprint arXiv:2409.16597*, 2024a.

Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Peng Gao, and Hongsheng Li. MathVerse: Does your multi-modal LLM truly see the diagrams in visual math problems? In *European Conference on Computer Vision (ECCV)*, 2024b.

A Discussion

Why Does Staging Work? The interaction between visual attention and predictive entropy motivates staging. Staging preserves complementary signals by letting VAG cover all response tokens in Phase 1 before EGR narrows the update to high-entropy tokens (Section 4.4).

Extensions. The same staging principle can be explored on longer videos, adaptive versions of the grid-searched hyperparameters, smoothly varying phase schedules, and larger multimodal models.

B Reproducibility Details

Controlled training setup. The controlled Qwen3-VL-8B-Thinking experiments use the same curated 36.6k subset of ViRL39K (33,213 training and 3,368 held-out examples), rule-based correctness reward, and optimization budget for VIEW and the GRPO/DAPO baselines. Each prompt produces five rollouts at temperature 1.0. The rollout and policy batches contain 256 prompts and 128 prompts, respectively. We freeze the vision tower and fine-tune the language backbone in bf16 with AdamW at a learning rate of 1×10^{-6}. Training uses eight NVIDIA A100 GPUs for two epochs (258 steps). The first epoch trains VAG; EGR resumes from its checkpoint in the second.

Token-signal configuration. VAG averages visual attention over all heads in the final four decoder layers, then applies sequence-internal min–max normalization. We use $\beta_{\text{base}} = \beta_{\text{mod}} = 0.5$, which gives $m_t \in [0.5, 1.0]$. EGR ranks response tokens by next-token predictive entropy, keeps the top 20% within each sequence (with at least one selected token), and normalizes the loss by the number of selected tokens. VAG is disabled in the EGR phase, as specified in Algorithm 1.

Evaluation protocol and comparison scope. We follow each benchmark's official protocol. MathVerse uses the vision-dominant split and its official GPT-based evaluation (788 examples). MMK12 uses the official GPT-based evaluation with DeepSeek-V3 as judge and the reported overall score (2,000 examples). HallusionBench reports `Overall_acc` from the official judge with `gpt-4o-2024-08-06`. The Qwen3-VL-8B-Thinking block in Table 1 is the controlled comparison. The Qwen2.5-VL block evaluates transfer to a second backbone, while the larger open-source models are reference points rather than training-controlled baselines. The displayed average is the unweighted arithmetic mean of the seven benchmark percentages. For the reported columns, the selected fields are POPE accuracy, HallusionBench `Overall_acc`, VideoHallucer overall accuracy, MathVerse overall, MMK12 overall, MME question accuracy, and MVBench overall score.

B.1 Evaluation and Reporting Details

Each reported benchmark row is tied to its model, benchmark split and sample count, generation configuration, and parser or judge identifier. The final model state is used unless a row is explicitly marked as a validation-reward "best" state. Failed generations and judge-invalid outputs remain in the benchmark's official denominator. Benchmark values are reported as point estimates under this protocol.

B.2 Exact Data and Optimization Details

The controlled data contains 33,213 training examples and 3,368 held-out examples (36,581 in total). Training and validation use these fixed splits; prompts are shuffled with seed 1. We cap prompts at 4,096 tokens and responses at 2,048 tokens, filter overlong prompts, sample videos at 2 fps, and constrain image resolution to 262,144–4,194,304 pixels. The rollout batch contains 256 prompts (mini-rollout batch 64) and the policy batch contains 128 prompts. Each prompt produces five rollouts

at temperature 1.0, top-$p = 1.0$, with one PPO epoch per update. The actor is trained in bf16 with AdamW (learning rate 10^{-6}, betas $(0.9, 0.999)$, weight decay 0.01, no warm-up, and maximum gradient norm 1.0), while the vision tower is frozen. Dynamic batching, padding-free attention, and the same split, shuffle seed, and loader settings are used for the controlled runs. Rollout sampling is stochastic, and DAPO-style filtering is applied to each rollout group under the fixed split.

The sequence-level reward is $0.9r_{\text{accuracy}} + 0.1r_{\text{format}}$. The format score is one only when the complete response contains a closed `<think>` block followed by optional whitespace and a closed `<answer>` block. For accuracy, the parser first extracts the `<answer>` span, then falls back to text after the closing think tag or the full response; it recognizes short yes/no, multiple-choice, ratio, numeric, and free-form answers. The normalized candidate is compared with the ground truth using the MathRuler grader and a deterministic normalized-string fallback. This parser produces one response-level score for the rule-based reward.

For DAPO-style online filtering, the five rollouts for each prompt are kept as a group. The filter computes the group's mean `overall` reward and retains the complete group only when that mean lies strictly between 0.01 and 0.99; additional groups are sampled until the requested rollout batch is filled. This fixed reward-interval filter is neither a token label nor benchmark-dependent.

B.3 Exact VAG and EGR Update Protocol

For the final staged configuration, visual attention uses the last four decoder layers, all heads, and sequence-level normalization with $\beta_{\text{base}} = \beta_{\text{mod}} = 0.5$. The raw attention ratio is detached from the policy gradient; VAG multiplies the sequence-level advantage by m_t. Both signals are computed after a complete response has been sampled. Sequence min–max and per-response entropy TopK therefore construct offline actor-update weights; neither signal is fed back into autoregressive decoding. The sequence-level normalization is applied after each complete response is available.

EGR entropy is recomputed from current-policy logits in each actor-update microbatch, rather than read from the sampler's metadata. The entropy tensor is detached before TopK selection, the mask is recomputed for every update, and it is zero outside the response span. The response span includes both reasoning and answer tokens; prompt, image/video, padding, and post-EOS positions are excluded. For a response with T valid tokens, $k = \max(1, \lceil 0.2T \rceil)$ positions are selected independently within that response. The implementation's deterministic TopK tie behavior is retained. The per-response TopK choice is followed by a token-average masked loss over the update microbatch, rather than an equal-weight mean over responses. The mask carries no gradient and no auxiliary entropy loss is added. Because TopK compares all valid response positions, its selection is formed after the response is available and does not alter the sampled text.

The attention extraction uses the model's image/video placeholder IDs and a causal, non-padding key mask. Its denominator is the total mass on visible keys, and the ratio is computed before sequence-level normalization. During training, the implementation alignment reads the attention query at the generated token position $q(t)$, while the logits used for the predictive entropy target y_t come from $q(t-1)$. The validation appendix therefore reports both $q(t)$ and the strict prediction-aligned $q(t-1)$ score. The attention pass uses a prefix KV cache and response chunks during training; inference uses the unchanged policy without an additional module.

B.4 Statistical Scope

The benchmark rows follow the fixed optimization-seed protocol. The confidence intervals in Table 4 quantify uncertainty from the content-span diagnostic by resampling complete answers. The Qwen3 block shares the backbone, data split, and two-epoch budget across the reported methods, while baseline implementations retain their documented GRPO/DAPO clipping, filtering, and KL settings.

The Qwen2.5 and larger-model rows use their respective transfer or reference configurations.

For clarity, the DAPO-style VIEW configuration uses asymmetric clipping $(\epsilon_{\text{low}}, \epsilon_{\text{high}}) = (0.20, 0.28)$ with a negative-advantage dual clip of 10, online group-mean filtering on `overall` in $(0.01, 0.99)$, and a disabled reference-policy KL branch. The vanilla GRPO comparison uses its own documented settings: upper clip 0.30, dual clip 3, KL coefficient 0.01, and no online filtering. These flags are part of the baseline definitions and are not silently treated as token-credit ablations.

The implementation adds 10^{-6} to the group-score standard deviation when forming the GRPO advantage; a zero-variance group therefore contributes a zero normalized advantage. This numerical stabilizer is omitted from the compact equation in the main text.

C Additional Analysis

C.1 Hyperparameter Sensitivity

Table 3 varies VIEW's two hyperparameters. For the modulation split μ, all four settings stay within about one point on both POPE and MMStar, so VIEW is robust to how the advantage is split between the base and modulated terms, and the equal split $0.5/0.5$ is best on both columns. Tilting the split toward the base term $(0.7/0.3)$ reduces the attention-based modulation effect and gives the lowest scores, while disabling modulation entirely $(1.0/0.0)$ reduces VIEW to plain DAPO followed by EGR. For the entropy top ratio ρ, we sweep DAPO + EGR without VAG so the ratio is the only moving part. Both POPE and MMStar peak at $\rho = 0.2$ in this sweep, identifying a small high-entropy subset as the most effective token budget among the tested settings.

C.2 Training Dynamics

Figure 4 tracks actor entropy (left) and validation reward (right) during training. The curves use the logged sampled-token surprisal/actor metric and validation reward, apply the plotting script's 10-step moving average, and retain only logged points. The step and cumulative-policy-token axes are aligned to the common two-epoch budget before smoothing; no interpolation is used to fill missing logged points. GRPO and PAPO drift toward higher entropy $(0.34 \rightarrow 0.38)$ and stay there. VIEW moves the other way: entropy declines through the VAG phase and falls faster once EGR concentrates updates on the forking tokens, ending at 0.22. The right panel shows that this drop is accompanied by improved validation reward. VIEW's validation reward rises above both baselines early and stays highest to the end (≈ 0.74), alongside the lower entropy achieved by staged token selection.

C.3 Qualitative Case Study: Benchmark Decision Forks

The qualitative examples below are selected from matched Qwen3-VL-8B evaluation traces after the same two-epoch training budget (evaluation temperature 0.6, seed 42). They demonstrate answer revisions. The four examples are deliberately different from the bench/chair and circle-size examples shown in the main-text qualitative figure: the first pair tests fine-grained object grounding, while the second tests ordering and relations.

These examples illustrate how VIEW changes the training credit assigned around uncertain response tokens and complement the quantitative ablation and content-span validation.

C.4 What VAG Reweighting Means for the Gradient

For a fixed response and an unclipped local view, the VAG contribution at token t has the form

$$\nabla_\theta \mathcal{L}_t^{\mathrm{VAG}} \;=\; m_t A_t \, \nabla_\theta \log \pi_\theta(y_t \mid y_{<t}, \mathbf{I}, \mathbf{x}). \tag{14}$$

Thus a larger m_t amplifies a positive-advantage update and also amplifies the suppression of a negative-advantage token. VAG is a proxy-based reweighting of a sequence-level credit signal, with the clipped objective and dynamic sampling preserving this sign interpretation.

C.5 Phase Accounting and the DAPO Control

The staged run has two epochs: the first contains the VAG updates and the second continues from the Phase 1 model state with EGR. The $\rho = 0$ column in Table 3 is the plain DAPO-plus-data pipeline with EGR disabled, and is included as a training-base control even though the main table emphasizes GRPO and PAPO variants. EGR-only and reverse-order rows keep the same nominal two-epoch budget; their difference from VIEW is the token signal or phase order. This isolates the curriculum effect in the reported comparison.

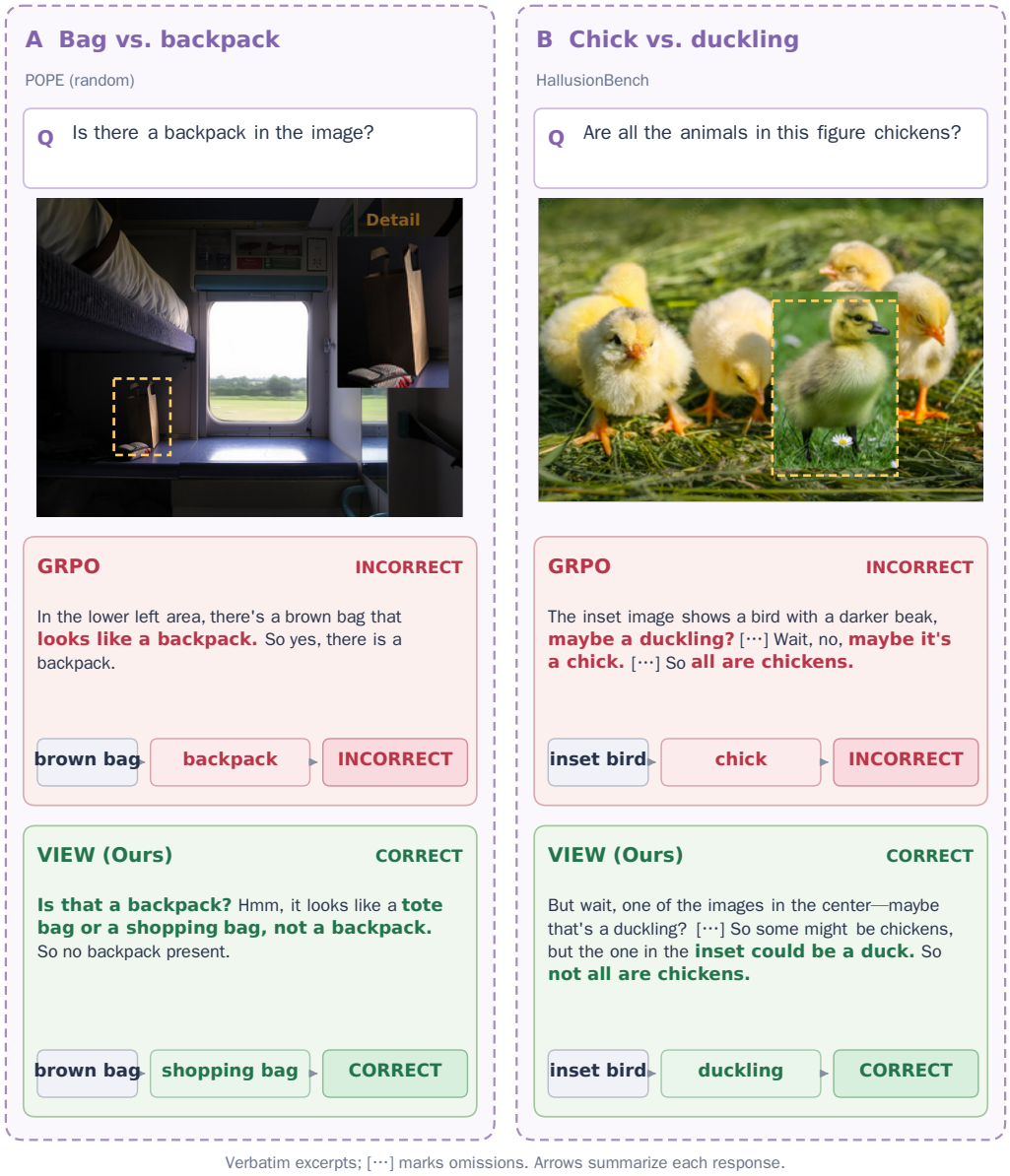

Figure 5: Qualitative examples (A–B) selected from matched Qwen3-VL-8B evaluation traces after the same two-epoch training budget (evaluation temperature 0.6, seed 42). In (A), the POPE item has a "no" gold label: GRPO calls the visible brown bag a backpack, while VIEW calls it a tote/shopping bag and predicts "no". In (B), the HallusionBench annotation identifies one duck in the figure; GRPO predicts that all animals are chickens, whereas VIEW predicts "no". The panels show the original benchmark images and contiguous reasoning excerpts ([. . .] marks omitted text); the arrows are author-written summaries.

D Content-Span Validation of the Visual-Attention Proxy

This appendix tests the central interpretation behind VAG: a response token that is grounded in the image should receive more visual attention than a token that states an unsupported or contradicted fact. The labels below are attention-independent, model-assisted annotations.

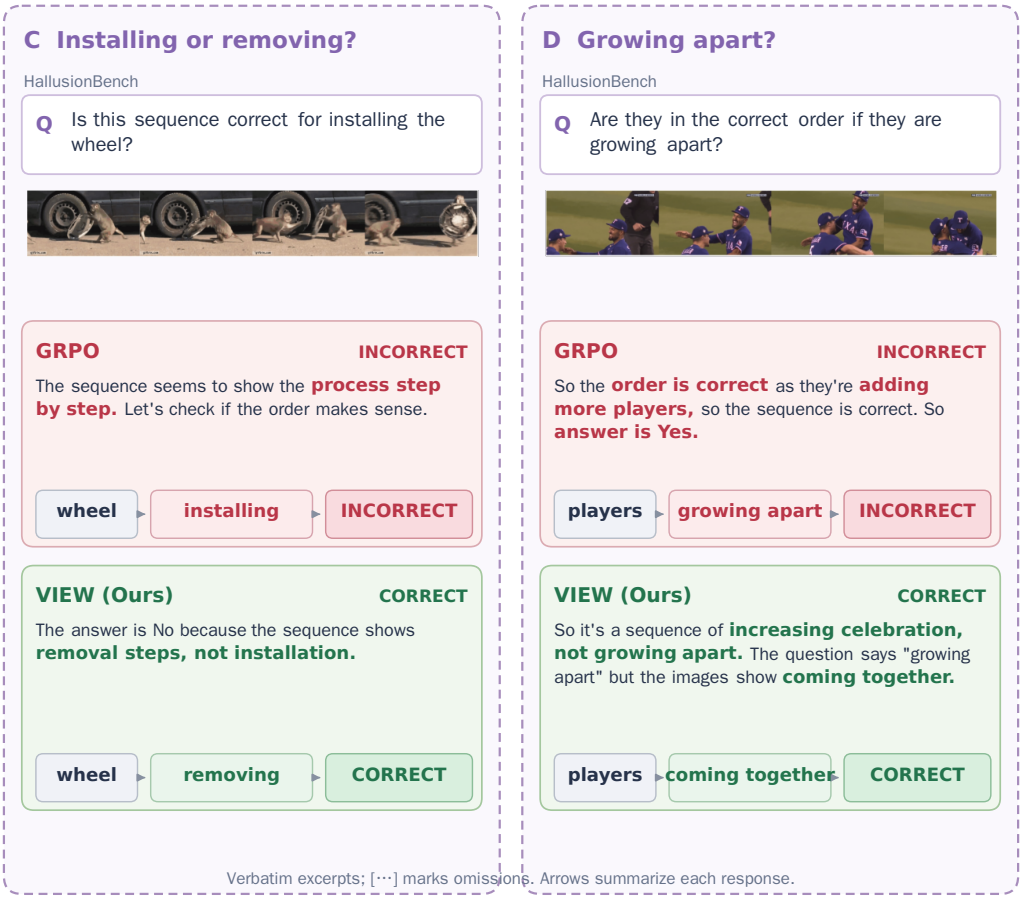

Figure 6: Additional qualitative examples (C–D) from the same saved evaluation traces. In (C), the HallusionBench sequence depicts wheel removal rather than wheel installation; GRPO accepts the order, whereas VIEW rejects it. In (D), the players come together in a celebration rather than grow apart; GRPO again accepts the order, while VIEW identifies the relational mismatch. The sequence strips are the original benchmark inputs; the abbreviated questions retain the tested relation. Status labels indicate correctness; the excerpts illustrate the examples only.

D.1 What is measured?

Let $A_{q,j}^{(\ell,h)}$ denote the decoder attention from query position q to key position j in layer ℓ and head h, and let $\mathcal{V}$ be the set of image-token positions. Let $\mathcal{K}_t$ be the causally visible, non-padding key positions for the query used to score response position t. Define $\bar{A}_{q,j} = (LH)^{-1} \sum_{\ell \in \mathcal{L}} \sum_{h=1}^{H} A_{q,j}^{(\ell,h)}$. For a response position t, the raw visual attention ratio is

$$\mathrm{VisAttn}_t = \frac{\sum_{j \in \mathcal{V} \cap \mathcal{K}_t} \bar{A}_{q(t),j}}{\sum_{j \in \mathcal{K}_t} \bar{A}_{q(t),j} + \epsilon}, \qquad |\mathcal{L}| = L = 4. \tag{15}$$

The quantity is a mass ratio, so it is in $[0, 1]$ before any sequence-level normalization. The denominator makes the masking convention explicit: only causally visible, non-padding keys are included. We use the final four decoder layers and average all heads. Importantly, this validation stores the *raw* ratio in Eq. (15); it does not apply the min–max normalization used inside VAG. This prevents a normalization operation from creating an artificial ranking within a response.

There are two query alignments. The implementation alignment uses $q(t)$, the query at the response-token position. The strict causal alignment uses $q(t-1)$, because the hidden state at position $t-1$

Table 4: Token-level content-span validation on 96 MMHal examples per model. AUROC and AUPRC intervals are 95% sample-cluster bootstrap confidence intervals; quintile rates are point estimates. The AUPRC treats unsupported or contradicted tokens as the positive class.

Alignment	Base AUROC	VIEW AUROC	VIEW unsupported-token AUPRC	VIEW unsupported-token rate: lowest / upper four quintiles
Query t	$0.500\ [0.436, 0.565]$	$0.632\ [0.490, 0.750]$	$0.306\ [0.090, 0.565]$	26.8% / 10.9%
Query $t-1$	$0.488\ [0.425, 0.558]$	$0.626\ [0.486, 0.748]$	$0.300\ [0.090, 0.556]$	26.8% / 10.9%

is the state that predicts the next token y_t. Reporting both makes the result robust to the indexing convention used when extracting decoder attention.

D.2 From answers to supported and unsupported spans

We evaluate the Qwen2.5-VL-7B-Instruct base model and its VIEW-trained counterpart on the same 96 MMHal image–question examples and retain their generated answers. Gemini 3.7 Flash is used only as a blind span annotator: it receives the image, question, and answer, but not the model identity, attention values, token probabilities, or reference labels. It decomposes an answer into short, exact character quotes and assigns one of four labels: `supported`, `contradicted`, `not_visible`, or `non_visual`. Exact-quote validation rejects an annotation when its quote cannot be located in the answer. The annotator is prompted to return JSON with an answer-level assessment and atomic claims containing an exact quote, occurrence index, claim type, label, confidence, and a short visual-evidence field. Requests use temperature zero; malformed JSON or invalid quotes are retried and then excluded rather than silently repaired.

For every valid quote, character offsets are mapped to the response-token offsets returned by the model tokenizer. The binary target used for the analysis is therefore

$$z = \begin{cases} 1, & \text{supported by the image,} \\ 0, & \text{contradicted or not visible in the image.} \end{cases} \tag{16}$$

Non-visual statements are excluded because their truth does not require visual evidence. The main analysis is token-level: every labeled response token inherits the label of its containing quote. We also report claim-level means as a conservative aggregation; the latter can dilute a signal that is localized to only the factual part of a long sentence. The annotator produced 994 claims in total, including 892 visual claims used for the binary analysis; after overlap checks, 7,882 response tokens were available for the token-level analysis.

The binary target pools `contradicted` and `not_visible` as "unsupported or contradicted." The label counts are 385/39/5/51 for the base model and 404/47/12/51 for the VIEW model (supported/contradicted/ not_visible/non_visual).

D.3 Statistics and interpretation

For a signal s_t, AUROC measures whether supported tokens rank above unsupported or contradicted tokens. For the unsupported-token AUPRC, that class is the positive class and the score is $-\mathrm{VisAttn}_t$, so that a larger score always means a stronger unsupported-token prediction. We additionally report the point-biserial correlation, Cliff's delta, and the unsupported-token rate in visual- attention quintiles. All confidence intervals use a 2,000-repeat bootstrap that resamples complete answers (not individual tokens). This clustering is necessary because many tokens from one answer share the same image, question, and annotation decision. For Base–VIEW deltas, bootstrap samples match the image–question item across both models; within-model metrics resample complete answers.

For completeness, the full token-level metric vector is as follows (means and medians are listed as supported/unsupported; the last three entries are the standardized logistic coefficient, point-biserial correlation, and Cliff's delta). Base at query t is $0.142/0.127$ versus $0.139/0.127$, 0.041, 0.012 $[-0.053, 0.070]$, and 0.001 $[-0.128, 0.131]$; at query $t-1$ it is $0.145/0.130$ versus $0.144/0.133$, 0.008, 0.002 $[-0.061, 0.065]$, and -0.025 $[-0.150, 0.116]$. VIEW at query t is $0.142/0.127$ versus $0.110/0.086$, 0.435, 0.132 $[-0.021, 0.290]$, and 0.264 $[-0.019, 0.500]$; at query $t-1$ it is $0.146/0.131$ versus $0.115/0.090$, 0.410, 0.126 $[-0.025, 0.279]$, and 0.252 $[-0.028, 0.496]$. The intervals are answer-cluster bootstrap intervals; means, medians, and logistic coefficients are point estimates in the released analysis, while clustered intervals are reported for AUROC, AUPRC,

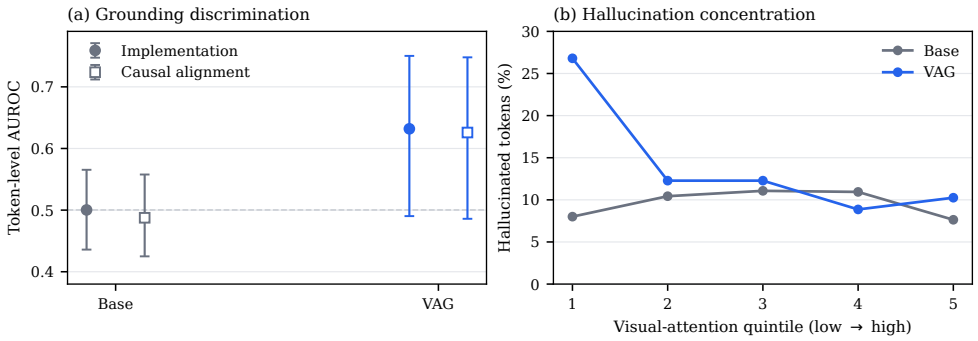

Figure 7: Content-span validation of raw visual attention. (a) VIEW shows a directional token-level association with grounding under both query alignments. (b) In VIEW, the lowest visual-attention quintile contains more unsupported or contradicted tokens than the remaining four quintiles.

correlation, and Cliff's delta. The unsupported-token prevalence is $557/3951 = 14.1\%$ for VIEW, which is the random-score reference for its AUPRC.

D.4 Results

At token level, Base attention is 0.500, while VIEW reaches AUROC 0.632 for the implementation alignment and 0.626 for the strict causal alignment. The lowest VIEW attention quintile has an unsupported-token rate of 26.8%, compared with 10.9% in the upper four quintiles. This pattern supports using visual attention as a grounding-sensitive credit signal.

D.5 Blind Human Audit of the Automatic Labels

We additionally sampled spans for a blind human audit. The auditor saw only the image, question, answer, and quoted span; the model identity and Gemini decision were hidden. In an initial 40-row development audit, 37 rows received a binary human judgment and 29 agreed with the Gemini-derived label (78.4%; Cohen's $\kappa = 0.557$). We then drew a later 40-row batch after revising the annotation interface. Thirty-eight rows in this batch received a binary judgment, and 35 agreed with Gemini, giving $35/38 = 92.1\%$ binary agreement. The non-binary rows are excluded from the binary denominator.

D.6 Additional Validation Checks

The following checks document the alignment, annotation, and statistical protocols used for this diagnostic.

D.6.1 Alternative explanations and alignment checks

The response-relative-position negative control has support-oriented AUROC 0.409 for VIEW under the implementation alignment (and 0.429 when the two models are pooled), whereas raw visual attention gives 0.632 and 0.579, respectively. Reversing the position orientation gives AUROC $1 - 0.409 = 0.591$ for VIEW, while visual attention remains the stronger grounding-sensitive signal.

The result is also stable to the query-indexing convention. Replacing the query at position t with the strictly causal query at position $t - 1$ changes VIEW token AUROC from 0.632 to 0.626 and unsupported-token AUPRC from 0.306 to 0.300. These two measurements use identical answer tokens and labels; only the attention query changes.

The extraction uses the model's causal mask. The prompt prefix is cached and response queries are evaluated in chunks, so future response tokens cannot enter the visual score. The final query is discarded for the predictive log-probability and entropy baselines because it has no next-token target; the implementation visual score retains the final $q(t)$ attention query. For a labeled response token y_t,

the entropy and log-probability baselines use the prediction-aligned query $q(t-1)$, while the two visual measurements use $q(t)$ and $q(t-1)$, respectively. This makes the indexing distinction explicit rather than treating the two extraction conventions as identical.

D.6.2 Annotation, token assignment, and statistical unit

Gemini receives only the image, question, and candidate answer; model identity, reference answers, attention values, and token scores are withheld. Every accepted span must be an exact contiguous substring before character offsets are mapped to tokenizer offsets. Of 994 annotated claims, 892 visual claims enter the binary analysis. Overlapping claims create 27 extra assignments, and 14 tokens with conflicting labels are removed before scoring. The resulting binary target is "supported" versus "unsupported or contradicted" at the content-span level.

All confidence intervals resample complete answers, preserving the shared image, question, generation, and annotation decision. The VIEW sample has 3,951 labeled tokens across 96 answers, of which 29 contain at least one unsupported or contradicted token. The experiment labels content spans such as entities, attributes, and relations.

The blind human audit had an initial 40-row development batch with 37 binary judgments, with 29 agreements (78.4%; Cohen's $\kappa = 0.557$). The later 40-row batch had 38 binary judgments and 35 agreements (92.1%); non-binary rows are excluded from the denominator.

D.6.3 Training Interpretation

The final four decoder layers and all-head average match the training configuration. The strict $q(t-1)$ analysis reports query alignment, and the μ and ρ sensitivity curves provide the grid-search evidence used to set the reported configuration.

D.6.4 Benchmark and Reporting Details

The content-span analysis uses exact-quote validation, deterministic decoding settings, answer-clustered bootstrap resampling, and the benchmark's reported train/test split. These protocol choices are recorded together with the parser, judge, generation parameters, and tokenization settings.

D.7 Scope and Reproducibility

This diagnostic uses 96 MMHal examples for each of a Qwen2.5-VL-7B-Instruct base model and its VIEW-trained counterpart (192 generated answers in total). The two query alignments, attention-position control, and content-span labels provide complementary views of the proposed token-level credit signal.

For reproduction, key protocol choices include the final four decoder layers and all-head average, raw pre-normalization visual-attention ratios, exact-quote span mapping, and answer-clustered 2,000-repeat bootstrap. The 96 MMHal items were selected deterministically. Generation settings, the annotation prompt, and invalid-output handling are part of the protocol and are reported here for reproducibility.

E Formula Reference

Table 5: Summary of key equations.

#	Equation	Description
(4)	$\mathrm{VisAttn}_t$	Visual attention ratio
(5)	s_t	Sequence-internal min-max normalization
(6)	$m_t = \beta_{\text{base}} + \beta_{\text{mod}} \cdot s_t$	Modulation factor
(7)	$\tilde{A}_t = A_t \cdot m_t$	VA-modulated advantage
(8)	H_t	Token-level predictive entropy
(9)	$\mathcal{S}_{\text{EGR}}$	Forking-token selection (top-ρ)
(10)	$\mathcal{M}_t^{\text{EGR}}$	High-entropy (EGR) mask
(11)	$\mathcal{L}_{\text{EGR}}$	EGR-selective policy gradient