GENERATE, VERIFY, RETRACT: EXPLOITING THE GENERATION–VERIFICATION GAP IN VISION-LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language models (VLMs) can generate visually unsupported objects, which then become part of the autoregressive context and may influence subsequent generation. We uncover a substantial **generation–verification gap**: on paired generated object events, the same frozen VLM's explicit object-existence judgments detect hallucinations substantially better than generation-token probabilities (AUROC 0.8584–0.9022 vs. 0.6093–0.6155). This suggests that although a VLM may fail to generate only visually grounded content, it can often recognize an unsupported object once that object has been proposed. Motivated by this observation, we introduce **RETRACT**, a decoding framework that transfers the model's explicit verification capability into its generation process. RETRACT distills object-existence judgments from the frozen VLM into a lightweight verifier operating on candidate-specific internal representations. During decoding, each object candidate is verified after being processed but before being retained in the generation context, allowing unsupported candidates to be locally retracted before they influence subsequent generation. This design avoids repeated explicit verification queries and requires no modification to the VLM parameters. Across multiple VLM backbones and hallucination benchmarks, RETRACT consistently reduces object hallucinations while preserving generation quality, with only modest inference overhead. Code is available at `https://anonymous.4open.science/r/RETRACT/`.

1 INTRODUCTION

Vision-language models (VLMs) (Bai et al., 2025; Zhu et al., 2025; Gemma Team et al., 2025) can generate linguistically plausible descriptions containing objects unsupported by the image (Rohrbach et al., 2018; Li et al., 2023; Leng et al., 2024). Such discrepancies have been studied extensively through benchmarks such as CHAIR (Rohrbach et al., 2018) and AMBER (Wang et al., 2023). In autoregressive generation, each object mention serves both as a claim about the image and as context for subsequent tokens. An unsupported object can therefore influence later descriptions, motivating feedback that assesses object candidates before they guide subsequent generation.

The model itself provides a potential source of this feedback. Prior work shows that visual information can remain available in intermediate predictions even when it is not reflected in the final output (Wang et al., 2025; Li et al., 2025). Research on self-evaluation further distinguishes answer generation from correctness assessment (Kadavath et al., 2022), with a corresponding generation–verification gap observed in VLM object judgments (Chen et al., 2026). We examine this distinction at the level of generated object events (Figure 1). Across three backbones on 10,000 COCO (Lin et al., 2014) images, explicit object-existence verification consistently outperforms generation-token risk in paired comparisons (AUROC: 0.8584–0.9022 vs. 0.6093–0.6155; Appendix B.5). This advantage motivates using the model's own verification behavior to guide generation.

Turning this verification advantage into generation-time correction poses two challenges. First, the decoder must **identify which object candidates warrant intervention** using an efficient, candidate-specific grounding signal: generation probability provides limited discrimination, while explicit object-existence queries require additional model computation. Second, it must **translate rejection**

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.

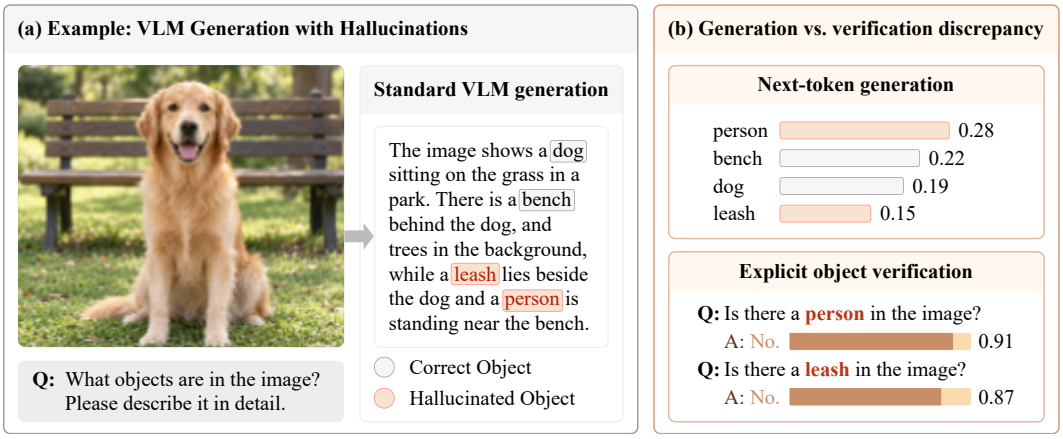

Figure 1: **Generation and verification can disagree.** A VLM may assign high generation probability to unsupported objects, while explicit verification evaluates their visual grounding.

into a local update of the generation process. Since a processed candidate contributes to the state used by subsequent tokens, correction must exclude its influence from future context and select a replacement while preserving the accepted prefix.

To address these challenges, we propose **RETRACT**, a decoding framework that converts a frozen VLM's own verification judgments into feedback for object-level correction. RETRACT uses explicit object-existence judgments from the same model as soft supervision (Hinton et al., 2015) for a lightweight linear verifier. During decoding, the verifier reads the representation obtained after an eligible object candidate has been processed by the backbone. This *post-commit* representation is conditioned on the image, retained prefix, and candidate itself, enabling the verifier to estimate the candidate's hallucination risk within the generation trajectory.

RETRACT makes this feedback actionable by **separating candidate processing from candidate retention**. Its *commit–check–retract* procedure first processes a candidate, then uses verification to determine whether it should remain in the generation context. Accepted candidates reuse the continuation logits from the same forward pass. Rejected candidates are excluded from subsequent attention through KV-cache masking, and replacements are selected under the retained prefix. This separation allows the model to assess a candidate after processing it and correct the decision before generation advances, without repeated verification queries or recomputation of the entire prefix.

Our contributions are summarized as follows:

- We characterize the generation–verification gap through paired object-event analysis across three VLM backbones and distill the model's own object-existence judgments into a lightweight verifier.
- We introduce post-commit verification with local retraction, separating candidate processing from candidate retention to identify and correct unsupported object candidates during autoregressive decoding.
- We demonstrate consistent reductions in object hallucination across multiple backbones and benchmarks, with systematic analyses of content coverage, inference cost, and key design choices.

2 Preliminaries

Given an image I and an instruction q, a vision-language model (VLM) p_θ defines the probability of an output sequence $y = (y_1, \ldots, y_T)$ through the standard autoregressive factorization (Vaswani et al., 2017; Holtzman et al., 2019)

$$p_\theta(y \mid I, q) = \prod_{t=1}^{T} p_\theta(y_t \mid I, q, y_{<t}), \tag{1}$$

where $y_t \in \mathcal{V}$ and $\mathcal{V}$ is the vocabulary. We use $y_{<t}$ to denote the retained prefix, $c_t = (I, q, y_{<t})$ the current context, and $l_t(v)$ the base model's logit for token v under c_t. We distinguish a candidate awaiting verification, $\hat{y}_t$, from the token ultimately retained, y_t. For a candidate v that maps to

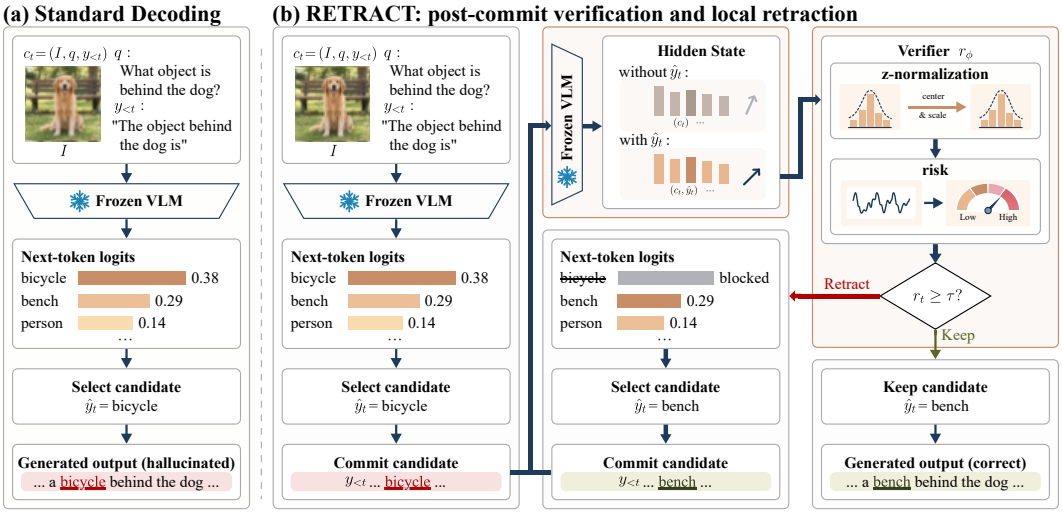

Figure 2: **Overview of RETRACT.** RETRACT separates candidate processing from candidate retention during autoregressive decoding. After a candidate is processed, a lightweight verifier provides grounding feedback by estimating object hallucination risk from its post-commit representation. High-risk candidates are locally retracted and replaced under the retained context, allowing verification decisions to influence subsequent generation.

an object mention, $o(v)$ denotes the corresponding object term; verification assesses whether its presence is supported by the image. The base model parameters θ remain frozen throughout.

Candidate selection is followed by a forward pass that computes a candidate-specific representation. Let $F_\theta^{(\ell)}$ return the final-position hidden state at layer ℓ. The pre-commit and post-commit states are

$$h_t^- = F_\theta^{(\ell)}(c_t), \qquad h_t^+(v) = F_\theta^{(\ell)}(c_t, v). \tag{2}$$

For a fixed c_t, all candidates share h_t^-, whereas $h_t^+(v)$ depends jointly on the image, prefix, and candidate. Processing v also produces the logits for subsequent generation. We call this forward pass a *commit*; whether to retain the candidate is decided after verification. Decoding reuses prefix keys and values through the KV cache (Shazeer, 2019), and causal attention ensures that earlier representations do not depend on the newly committed candidate (Vaswani et al., 2017).

3 Method

RETRACT integrates object-level verification into autoregressive decoding through a closed-loop candidate verification and local retraction framework. As illustrated in Figure 2, RETRACT first processes a candidate token under the frozen VLM, estimates its object hallucination risk from the resulting post-commit representation, and feeds the verification outcome back into decoding by selectively retracting unsupported candidates before they affect future generation. The verifier training, online decoding, and cache-level retraction procedures are summarized in Algorithms 1, 2, and 3 in Appendix A.

3.1 Self-Distilled Object Verification

RETRACT obtains verification supervision from the frozen VLM itself, building on model self-evaluation (Kadavath et al., 2022). Given an image I and an object term o, we query the model with an object-existence question $Q(o)$, a format used in object-hallucination evaluation (Li et al., 2023): "Is there a [object] in the image? Answer yes or no." The query only contains the image and object term, without the generated caption. Let $\mathcal{V}_{\text{yes}}$ and $\mathcal{V}_{\text{no}}$ denote the affirmative and negative answer-token sets. We define the object absence risk as:

$$\tilde{r}(I, o) = \frac{\sum_{v \in \mathcal{V}_{\text{no}}} p_\theta(v | I, Q(o))}{\sum_{v \in \mathcal{V}_{\text{yes}}} p_\theta(v | I, Q(o)) + \sum_{v \in \mathcal{V}_{\text{no}}} p_\theta(v | I, Q(o))}. \tag{3}$$

where $\widetilde{r}(I, o)$ denotes the model's estimated object absence risk. For each generated object event, we pair its post-commit representation h_i^+ with $\widetilde{r}_i$, obtaining $\mathcal{D} = \{(h_i^+, \widetilde{r}_i)\}_{i=1}^N$.

Linear probes have been used to extract correctness signals from language-model representations (Orgad et al., 2025). We train a lightweight linear verifier over the frozen VLM representations:

$$z_i = \frac{h_i^+ - \mu}{s + \epsilon}, \qquad r_\phi(h_i^+) = \sigma(w^\top z_i + b_0), \tag{4}$$

where μ and s denote the feature statistics computed from training events and $\phi = (w, b_0)$ represents the verifier parameters. Only ϕ is optimized during training, while the VLM parameters remain frozen. The resulting verifier transfers the model's object-existence judgments into a compact feedback signal (Hinton et al., 2015) that can be queried during decoding.

3.2 Post-Commit Verification

Autoregressive decoding normally commits each predicted token immediately: once a token is appended to the prefix, all subsequent generation steps are conditioned on this decision (Vaswani et al., 2017). However, the probability assigned during token selection only reflects the model's preference among possible continuations and does not explicitly measure whether the selected object is visually supported (Leng et al., 2024). RETRACT therefore separates candidate processing from candidate retention: a candidate is first allowed to participate in the model computation, while the verification outcome is fed back to determine whether it should become part of the future generation context.

At decoding position t, RETRACT maintains the retained prefix $y_{<t}$ and current context $c_t = (I, q, y_{<t})$. The base model first proposes a candidate $\hat{y}_t = \arg\max_{v \in \mathcal{V}} l_t(v)$ according to its original decoding distribution.

Pre-commit detectors such as TruthPrInt (Duan et al., 2025) read a representation preceding the candidate token. In our notation, this state is shared by all candidates under the same retained prefix. Post-commit verification operates on the state after the candidate has participated in the forward computation. Specifically, the candidate is processed by the frozen VLM to obtain

$$h_t^+(\hat{y}_t) = F_\theta^{(\ell)}(c_t, \hat{y}_t), \tag{5}$$

where the representation jointly encodes the image evidence, retained prefix, and the candidate itself. This candidate-conditioned representation provides a more direct interface for estimating whether the newly introduced object is supported by the visual input.

For candidates corresponding to physical object terms, RETRACT applies the lightweight verifier introduced in Sec. 3.1 to compute $r_t = r_\phi(h_t^+(\hat{y}_t))$, where r_t represents the estimated object hallucination risk. The verification signal is fed back only to decide whether the current candidate should remain in the future context; it does not alter the underlying language model distribution. This design preserves the original generation ability while adding a selective feedback mechanism at object-level decisions.

When the per-position retraction budget has not been exhausted, the candidate is accepted if its estimated risk is below the threshold τ; otherwise, RETRACT locally retracts the candidate and performs reselection:

$$d_t = \begin{cases} \text{retract}, & r_t \geq \tau, \\ \text{keep}, & r_t < \tau. \end{cases} \tag{6}$$

Verification is restricted to newly introduced physical object terms that are fully expressed by the current token. Non-object tokens and previously verified objects bypass verification and continue normal decoding, ensuring that the feedback mechanism focuses computation on decisions with potential grounding errors.

3.3 Local Retraction and Reselection

Local retraction closes the verification loop by feeding the rejection decision back into the autoregressive state. A rejected candidate should not only be removed from the output sequence, but also excluded from the future context on which subsequent tokens are generated. RETRACT exploits

the causal structure of autoregressive models (Vaswani et al., 2017) to perform this correction locally. Since previous tokens are computed without access to future candidates, rejecting the current token does not require recomputing the retained prefix. Instead, RETRACT isolates the candidate's contribution by masking its corresponding KV-cache slot, i.e., setting its attention visibility to $m_{j_t} \leftarrow -\infty$.

This operation excludes the rejected candidate from subsequent attention while preserving the cached states of previously accepted tokens. Its physical KV slot remains allocated but permanently masked. RETRACT therefore corrects the current decision without recomputing the retained prefix.

After removing a candidate, RETRACT prevents the same unsupported object from being immediately selected again. Specifically, it maintains a temporary rejection set $\mathcal{B}$ and adds all token variants associated with the rejected object term: $\mathcal{B} \leftarrow \mathcal{B} \cup \mathcal{T}(o(\hat{y}_t))$, where $\mathcal{T}(o)$ denotes the tokenization variants of object o.

The replacement candidate is then selected from the original candidate distribution before the rejected object was committed:

$$\hat{y}_t \leftarrow \arg\max_{v \in \mathcal{V} \setminus \mathcal{B}} l_t(v). \tag{7}$$

The new candidate is processed by the same commit–check procedure, allowing RETRACT to repeatedly refine the current decision until a candidate is accepted or the retraction budget is reached. Once accepted, the candidate's continuation logits are directly reused and decoding proceeds to the next position. Therefore, local retraction turns verification into an actionable feedback loop while avoiding full-prefix regeneration.

4 Experiments

4.1 Experimental Setup

Models and benchmarks. We evaluate RETRACT on three representative vision-language models: Qwen2.5-VL-7B-Instruct (Bai et al., 2025), InternVL3-8B (Zhu et al., 2025), and Gemma-3-12B-IT (Gemma Team et al., 2025). We conduct evaluations on multiple hallucination benchmarks, including CHAIR (Rohrbach et al., 2018), AMBER (Wang et al., 2023), and OpenCHAIR (Ben-Kish et al., 2024). We report hallucination metrics together with benchmark-specific measures of content preservation and response quality.

Baselines and evaluation protocol. We compare RETRACT with Vanilla decoding, DeCo (Wang et al., 2025), VCD (Leng et al., 2024), VISTA (Li et al., 2025), DoLa (Chuang et al., 2024), ICD (Wang et al., 2024), LCD (Manevich & Tsarfaty, 2024), CausalLens (Ji et al., 2026), TruthPrInt (Duan et al., 2025), and MGAP (Zhuang et al., 2026). All methods are evaluated under matched backbone, input, prompt, and generation settings. RETRACT trains a lightweight verifier using self-generated object events and the frozen model's object-existence judgments. Detailed model configurations, baseline implementations, benchmark protocols, and verifier selection procedures are provided in Appendix B.

Generation and analysis. All experiments use frozen backbone models with BF16 inference and greedy decoding (Holtzman et al., 2019). For RETRACT, the generation budget counts candidate attempts, including candidates that are subsequently retracted. CHAIR and AMBER follow their benchmark evaluation protocols; OpenCHAIR uses the shared judge configuration detailed in Appendix B.3. Statistical analysis and additional experimental settings are described in Appendix B.

4.2 Main Results

Overall performance. Table 1 summarizes the results on CHAIR (Rohrbach et al., 2018) and AMBER (Wang et al., 2023). RETRACT achieves the best or second-best performance on the majority of evaluation metrics across different backbones, consistently reducing object hallucination while maintaining comparable output length. On CHAIR, RETRACT reduces CHAIR-i from 8.36

Table 1: **Object hallucination and content coverage on CHAIR and AMBER.** All hallucination and coverage metrics are reported in percentages. Avg. Len denotes the average whitespace-tokenized output length. TruthPrInt denotes our CC-SBU adaptation. Bold and underlined numbers indicate the best and second-best point estimates, respectively, for each hallucination or coverage metric within the same backbone.

	CHAIR (500 images)				AMBER (1,004 queries)			
Method	CHAIR-s↓	CHAIR-i↓	Recall↑	Avg. Len	CHAIR↓	Cover↑	Hal↓	Avg. Len
Qwen2.5-VL-7B								
Vanilla	31.80	8.36	66.90	151.54	5.61	64.51	30.48	85.70
VCD (Leng et al., 2024)	37.60	10.45	67.59	163.60	6.85	65.48	36.26	87.55
ICD (Wang et al., 2024)	33.00	8.70	67.09	176.72	6.04	**65.57**	34.86	92.96
DoLa (Chuang et al., 2024)	32.00	8.64	66.08	145.82	5.41	63.78	29.68	80.12
LCD (Manevich & Tsarfaty, 2024)	32.20	8.83	**67.90**	140.81	5.57	64.07	29.88	79.51
DeCo (Wang et al., 2025)	33.40	8.18	67.72	171.56	5.79	63.94	32.57	93.07
VISTA (Li et al., 2025)	30.00	7.67	66.58	171.02	5.14	65.14	29.18	96.76
CausalLens (Ji et al., 2026)	30.60	8.46	65.07	137.77	4.82	63.30	27.59	82.87
TruthPrInt (Duan et al., 2025)	29.60	7.66	65.64	149.90	5.42	64.40	30.38	85.83
MGAP (Zhuang et al., 2026)	32.80	9.03	66.65	153.47	5.61	64.65	29.48	84.53
RETRACT (ours)	**26.20**	**6.73**	64.13	152.10	**3.98**	63.89	**22.91**	85.97
InternVL3-8B								
Vanilla	29.20	8.66	66.90	97.18	7.39	71.57	61.45	212.56
VCD (Leng et al., 2024)	36.20	10.95	**68.60**	99.89	8.62	71.94	64.74	207.87
ICD (Wang et al., 2024)	29.00	9.22	66.96	97.69	6.69	63.26	52.39	207.82
DoLa (Chuang et al., 2024)	29.80	8.85	67.90	99.51	7.05	70.14	61.16	207.61
LCD (Manevich & Tsarfaty, 2024)	32.20	9.95	67.46	95.95	7.58	66.57	57.17	207.62
DeCo (Wang et al., 2025)	27.80	8.14	66.90	104.52	6.13	66.51	51.59	207.85
VISTA (Li et al., 2025)	29.80	8.14	67.97	121.81	6.01	67.87	49.00	212.33
CausalLens (Ji et al., 2026)	26.00	**7.29**	64.82	85.91	5.91	**72.14**	53.69	206.88
TruthPrInt (Duan et al., 2025)	27.60	8.14	66.39	97.10	7.29	71.57	61.06	211.81
MGAP (Zhuang et al., 2026)	27.60	8.51	65.95	97.10	7.30	72.01	61.85	213.20
RETRACT (ours)	**25.40**	7.65	65.89	98.06	**5.39**	69.19	**46.71**	203.22
Gemma-3-12B								
Vanilla	33.80	7.97	62.05	184.50	6.74	70.28	51.49	187.07
VCD (Leng et al., 2024)	38.00	9.43	62.74	183.65	8.16	**71.16**	59.96	186.68
ICD (Wang et al., 2024)	29.80	7.35	59.79	184.31	6.80	69.53	51.39	187.24
DoLa (Chuang et al., 2024)	32.80	7.86	61.42	178.39	6.69	69.33	52.49	180.74
LCD (Manevich & Tsarfaty, 2024)	39.00	9.24	**64.38**	185.17	7.25	70.42	55.68	183.42
DeCo (Wang et al., 2025)	33.20	7.87	61.23	185.20	6.76	69.92	53.59	187.76
VISTA (Li et al., 2025)	**27.60**	7.10	53.30	200.06	**5.71**	56.74	**38.55**	200.71
CausalLens (Ji et al., 2026)	33.20	7.84	60.86	184.49	6.35	68.28	50.30	186.00
TruthPrInt (Duan et al., 2025)	31.40	7.51	62.18	183.25	6.57	69.94	51.39	186.51
MGAP (Zhuang et al., 2026)	34.20	8.12	61.74	184.88	6.78	70.48	52.09	186.77
RETRACT (ours)	28.20	**6.73**	59.91	182.83	5.73	69.48	45.32	185.38

Table 2: **Open-vocabulary captioning on OpenCHAIR.** Each method generates 4,863 captions per backbone. OCH and HR denote object-level and caption-level hallucination rates, respectively, both reported as percentages; Len denotes the average whitespace-tokenized word count. All methods use the same judge configuration (Appendix B.3). Bold and underlined numbers indicate the best and second-best point estimates, respectively, for OCH and HR within each backbone.

	Qwen2.5-VL-7B			InternVL3-8B			Gemma-3-12B		
Method	OCH↓	HR↓	Len	OCH↓	HR↓	Len	OCH↓	HR↓	Len
Vanilla	7.84	14.52	16.24	6.54	12.58	10.03	10.15	19.72	32.29
VCD (Leng et al., 2024)	9.66	17.21	16.45	9.46	17.33	10.47	10.81	21.43	36.96
ICD (Wang et al., 2024)	8.48	15.40	16.22	**5.75**	**9.71**	9.36	9.31	19.35	37.09
DoLa (Chuang et al., 2024)	8.10	13.90	15.39	9.18	13.94	9.90	10.18	17.48	29.25
LCD (Manevich & Tsarfaty, 2024)	9.33	16.55	15.65	8.53	15.77	10.01	10.55	21.22	36.30
DeCo (Wang et al., 2025)	7.44	13.61	16.21	6.66	13.24	11.18	10.22	18.51	32.19
VISTA (Li et al., 2025)	7.19	12.56	15.99	6.52	12.32	12.02	9.96	22.48	52.52
CausalLens (Ji et al., 2026)	7.84	14.66	16.12	6.33	12.13	10.00	9.80	17.87	32.14
TruthPrInt (Duan et al., 2025)	7.50	14.11	16.21	6.62	12.75	10.04	10.43	19.84	32.16
MGAP (Zhuang et al., 2026)	7.76	14.37	16.21	6.91	13.16	9.92	10.42	19.99	32.43
RETRACT (ours)	**6.80**	**12.48**	16.20	6.16	11.87	10.01	**8.98**	**17.17**	32.16

to 6.73 on Qwen2.5-VL-7B, from 8.66 to 7.65 on InternVL3-8B, and from 7.97 to 6.73 on Gemma-3-12B. Similar improvements are observed on AMBER, where the AMBER CHAIR score decreases from 5.61 to 3.98, from 7.39 to 5.39, and from 6.74 to 5.73 for the three backbones, respectively. Importantly, these improvements are not obtained by simply shortening generated descriptions. The average output length of RETRACT remains close to Vanilla across benchmarks, e.g., 152.10 vs. 151.54 words on Qwen CHAIR and 98.06 vs. 97.18 words on InternVL CHAIR, indicating that the improvement mainly comes from correcting unsupported object choices during generation rather than reducing the amount of generated content.

Generalization to open-ended generation. Beyond closed-set object hallucination benchmarks, we evaluate RETRACT on OpenCHAIR, which measures open-vocabulary object hallucination (Ben-Kish et al., 2024). As shown in Table 2, RETRACT achieves the best or second-best point estimates across the three backbones for both the object-level hallucination rate (OCH) and the caption-level hallucination rate (HR). For Qwen2.5-VL-7B, RETRACT reduces OCH from 7.84 to 6.80 and HR from 14.52 to 12.48. For InternVL3-8B and Gemma-3-12B, RETRACT achieves OCH scores of 6.16 and 8.98, respectively. These results show reduced open-vocabulary object hallucination relative to Vanilla under the same evaluation protocol.

4.3 Inference Efficiency

We measure end-to-end latency and peak allocated GPU memory on 96 fixed inputs from CHAIR, AMBER, and OpenCHAIR, using batch size 1 on NVIDIA RTX 5090 GPUs. As shown in Table 3, RETRACT changes latency relative to Vanilla by $+5.7\%$, $+0.7\%$, and -0.6% on Qwen2.5-VL-7B, InternVL3-8B, and Gemma-3-12B, respectively, while peak memory differs by at most 0.004 GiB.

RETRACT evaluates its lightweight verifier on representations produced during decoding and reuses the continuation logits of accepted candidates. Additional computation arises from verifier evaluations and rejected candidate forward passes, while decoding maintains a single trajectory without an auxiliary branch. Appendix C.1 provides the corresponding computational and memory analysis.

Table 3: **Online inference cost.** Latency (s/response) is averaged over 96 inputs per backbone and method; memory (GiB) is the maximum allocated GPU memory across the three dataset runs.

	Qwen2.5-VL-7B		InternVL3-8B		Gemma-3-12B	
Method	Latency↓	Memory↓	Latency↓	Memory↓	Latency↓	Memory↓
Vanilla	2.103	15.573	2.297	14.861	8.461	22.920
VCD (Leng et al., 2024)	4.115	15.893	4.331	15.000	16.121	23.249
ICD (Wang et al., 2024)	4.415	15.900	4.366	15.006	16.199	23.256
DoLa (Chuang et al., 2024)	2.013	15.789	2.324	14.959	8.785	23.070
LCD (Manevich & Tsarfaty, 2024)	3.568	15.689	4.287	14.912	16.108	23.065
DeCo (Wang et al., 2025)	2.756	15.797	2.976	14.962	10.536	23.078
VISTA (Li et al., 2025)	3.716	15.756	4.019	14.944	14.236	22.956
CausalLens (Ji et al., 2026)	2.191	15.678	2.392	14.905	8.917	22.979
TruthPrInt (Duan et al., 2025)	2.309	15.579	2.346	14.865	8.754	22.924
MGAP (Zhuang et al., 2026)	3.704	15.573	1.656	14.861	9.815	22.921
RETRACT (ours)	2.222	15.577	2.313	14.863	8.414	22.919

4.4 Hyperparameter Sensitivity

We examine how the risk threshold τ affects hallucination and object coverage on CHAIR500 (Rohrbach et al., 2018). Complementary analyses of verifier readout depth and retraction budget are provided in Appendix C.2.

Figure 3 shows that the response to τ varies across backbones. For Qwen2.5-VL-7B, increasing τ generally improves Recall while weakening hallucination reduction. InternVL3-8B exhibits a similar trade-off at low thresholds, with Recall stabilizing as τ increases. Gemma-3-12B changes less across thresholds, with CHAIR scores mostly near or above Vanilla. These results motivate backbone-specific threshold selection using both hallucination and Recall; the development-set selection procedure is described in Appendix B.4.

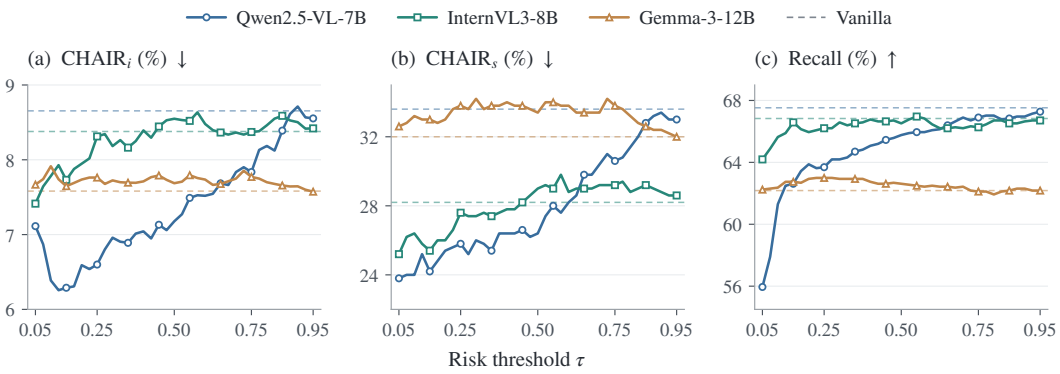

Figure 3: **Sensitivity to the risk threshold τ.** CHAIR-i, CHAIR-s, and object-category Recall on CHAIR500. Dashed lines denote the corresponding Vanilla baselines.

Table 4: **Ablation study of RETRACT components.** All variants use the same backbone, decoding protocol, and local retraction mechanism. Lower CHAIR-i/CHAIR-s indicates fewer hallucinations, while higher Recall indicates better content coverage.

Model	Variant	CHAIR-i↓	CHAIR-s↓	Recall↑	Retract/img
Qwen2.5-VL-7B					
Vanilla	Vanilla	8.65	33.60	67.53	0.00
Representation	Pre-commit	8.36	31.00	64.07	4.39
Signal	Token confidence	8.40	27.60	59.85	6.47
Signal	Predictive entropy	8.59	30.40	60.79	5.17
Signal	Random	9.72	32.47	60.29	4.06
KV	No KV masking	5.94	24.40	64.76	3.93
Full	**Post-commit (RETRACT, ours)**	**6.96**	**26.00**	64.19	3.94
InternVL3-8B					
Vanilla	Vanilla	8.89	30.20	67.28	0.00
Representation	Pre-commit	8.21	28.20	66.14	3.47
Signal	Token confidence	8.61	27.40	61.11	5.82
Signal	Predictive entropy	8.56	28.60	64.13	3.88
Signal	Random	9.87	30.60	62.20	3.23
KV	No KV masking	7.28	23.80	65.14	3.02
Full	**Post-commit (RETRACT, ours)**	**7.48**	**25.60**	66.14	3.17
Gemma-3-12B					
Vanilla	Vanilla	7.58	32.00	62.18	0.00
Representation	Pre-commit	7.50	32.20	59.85	1.90
Signal	Token confidence	7.24	31.00	60.29	3.41
Signal	Predictive entropy	7.16	31.20	61.36	2.52
Signal	Random	8.12	32.27	59.89	1.96
KV	No KV masking	6.42	26.60	60.86	1.89
Full	**Post-commit (RETRACT, ours)**	**6.62**	**27.60**	59.79	1.94

4.5 Ablation Study of Verification Mechanism

To investigate the contribution of different components in RETRACT, we conduct controlled ablation studies on three design choices: the representation position used for verification, the verification signal, and the treatment of rejected candidates in KV cache. All variants share the same backbone, input samples, prompts, decoding budget, and local retraction mechanism. Detailed experimental configurations are provided in Appendix C.3.

Representation position. We study whether the verification representation should be extracted before or after candidate commitment. Replacing the post-commit representation with the pre-commit state consistently increases hallucination rates across backbones. Pre-commit increases CHAIR-i by 1.40, 0.73, and 0.87 points on Qwen2.5-VL-7B, InternVL3-8B, and Gemma-3-12B, respectively.

This result supports our design choice that candidate-specific representations provide a more informative interface for object-level verification than shared prefix representations.

Verification signal. We compare the learned self-verification signal with uncertainty-based alternatives. Replacing the verifier with token confidence, predictive entropy, or random rejection consistently weakens hallucination reduction. Compared with random rejection, RETRACT reduces CHAIR-i by 2.76, 2.39, and 1.50 points on the three backbones, respectively. These results indicate that the benefit of RETRACT does not come solely from rejecting more candidates, but from using a verification signal aligned with object-level visual support.

KV masking. The no-masking variant keeps rejected KV states visible while applying the same token removal and reselection procedure. It yields lower CHAIR-i and CHAIR-s across all three backbones, with mixed changes in Recall. RETRACT uses KV masking to exclude rejected candidates from subsequent attention, keeping the effective context aligned with the retained sequence.

5 Related Work

Decoding and representation interventions. Inference-time methods mitigate hallucinations by modifying output distributions or internal representations. VCD contrasts predictions under original and perturbed images (Leng et al., 2024), ICD uses instruction perturbations (Wang et al., 2024), and LCD contrasts multimodal and language-only predictions (Manevich & Tsarfaty, 2024). DeCo dynamically incorporates intermediate-layer predictions into final logits (Wang et al., 2025), while VISTA combines visual representation steering with intermediate-layer logit augmentation (Li et al., 2025).

Self-verification and signal distillation. Kadavath et al. (2022) study language models' self-evaluation, while Orgad et al. (2025) show that correctness signals vary across answer-token representations. OSCAR combines tree search and self-verification to construct preference data for model optimization (Chen et al., 2026). PSRD uses phrase-level self-evaluation to train a CLIP-based reward model for phase-wise decoding (Zhang et al., 2026). RETRACT distills a frozen VLM's object-existence judgments into a linear verifier over its internal states.

Verification-guided online correction. TruthPrInt uses reference-derived labels to train a detector on pre-commit states and backtracks to low-confidence positions (Duan et al., 2025). REVERSE jointly trains generation and phrase-level verification for online retrospective resampling (Wu et al., 2025). UniProbe combines heterogeneous internal signals for token-level detection and supports resampling through its streaming variant (Samuel et al., 2026). RETRACT evaluates candidate-specific post-commit representations and uses local retraction and reselection to control candidate retention.

6 Conclusion

We presented RETRACT, a decoding framework that converts a frozen VLM's object-presence judgments into generation-time verification feedback. Through self-distillation, a lightweight linear verifier learns to assess object candidates from their post-commit representations. Local retraction and reselection then determine whether these candidates remain in the subsequent context. By separating candidate processing from candidate retention, RETRACT integrates object-level verification directly into autoregressive generation.

Experiments across three VLM backbones and multiple benchmarks show that RETRACT consistently reduces object hallucinations, while measured latency and peak GPU memory remain close to standard decoding. Ablation studies further support the effectiveness of candidate-specific representations and object-level verification signals. These findings demonstrate the practical value of using the model's own verification judgments to guide local generation decisions. Future work may extend this framework to multi-token object mentions, attributes, and relations for correcting more complex visual claims.

AI USE STATEMENT

During the research and preparation of this manuscript, we used GPT-6 Astra to assist with literature search and synthesis, manuscript organization and language refinement, translation of selected text, code debugging, and troubleshooting the execution environment. Specifically, the model helped identify relevant literature, improve textual clarity and coherence, analyze program errors, and suggest code fixes and environment configurations. The authors are responsible for reviewing and verifying the adopted AI-assisted content against original sources, experimental records, and program execution results, and take full responsibility for the final manuscript, references, code implementation, experimental results, and research conclusions.

ETHICS STATEMENT

This work aims to improve the reliability of visual descriptions by reducing unsupported object mentions. Our experiments use existing public benchmarks and pretrained vision-language models. Since verification relies on the backbone's own judgments, residual hallucinations and biases inherited from the underlying models may persist. Downstream applications should account for these limitations and include appropriate validation and human oversight where incorrect descriptions could affect individuals.

REPRODUCIBILITY STATEMENT

The method is described in Section 3, with pseudocode for verifier training and decoding provided in Appendix A. Appendix B documents the model configurations, data partitions, prompts, training procedures, hyperparameter selection, and evaluation protocols. The paired generation–verification analysis and its statistical procedures are detailed in Appendix B.5. Our source code is openly available at `https://anonymous.4open.science/r/RETRACT/` to support independent reproduction and further research.

REFERENCES

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, 2025. URL `https://arxiv.org/abs/2502.13923`.

Assaf Ben-Kish, Moran Yanuka, Morris Alper, Raja Giryes, and Hadar Averbuch-Elor. Mitigating open-vocabulary caption hallucinations. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 22680–22698, 2024.

Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. Concreteness ratings for 40 thousand generally known english word lemmas. *Behavior research methods*, 46(3):904–911, 2014.

Minghui Chen, Chenxu Yang, Hengjie Zhu, Dayan Wu, Zheng Lin, and Qingyi Si. Online self-calibration against hallucination in vision-language models. *arXiv preprint arXiv:2605.00323*, 2026.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R Glass, and Pengcheng He. Dola: Decoding by contrasting layers improves factuality in large language models. In *International Conference on Learning Representations*, volume 2024, pp. 54158–54183, 2024.

Jinhao Duan, Fei Kong, Hao Cheng, James Diffenderfer, Bhavya Kailkhura, Lichao Sun, Xiaofeng Zhu, Xiaoshuang Shi, and Kaidi Xu. Truthprint: Mitigating large vision-language models object hallucination via latent truthful-guided pre-intervention. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7372–7382. IEEE, 2025.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Peter J Green. Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives. *Journal of the Royal Statistical Society: Series B (Methodological)*, 46(2):149–170, 1984.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.

Junyang Ji, Qifan Liu, Wenming Yang, and Zhihai He. Causallens: Sensitivity-guided multi-head causal intervention for hallucination mitigation in large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4199–4209, 2026.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. Language models (mostly) know what they know, 2022. URL https://arxiv.org/abs/2207.05221.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 292–305, 2023.

Zhuowei Li, Haizhou Shi, Yunhe Gao, Di Liu, Zhenting Wang, Yuxiao Chen, Ting Liu, Long Zhao, Hao Wang, and Dimitris N. Metaxas. The hidden life of tokens: Reducing hallucination of large vision-language models via visual information steering. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 35799–35819. PMLR, 13–19 Jul 2025.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Edward Loper and Steven Bird. Nltk: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics*, pp. 63–70, 2002.

Avshalom Manevich and Reut Tsarfaty. Mitigating hallucinations in large vision-language models (lvlms) via language-contrastive decoding (lcd). In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 6008–6022, 2024.

George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11): 39–41, 1995.

Hadas Orgad, Michael Toker, Zorik Gekhman, Roi Reichart, Idan Szpektor, Hadas Kotek, and Yonatan Belinkov. Llms know more than they show: On the intrinsic representation of llm hallucinations. In *International Conference on Learning Representations*, volume 2025, pp. 66880–66913, 2025.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. Object hallucination in image captioning. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 4035–4045, 2018.

Dvir Samuel, Guy Bar-Shalom, Fabrizio Frasca, Ethan Fetaya, Yftah Ziser, Gal Chechik, and Haggai Maron. Uniprobe: A learnable token-level hallucination detector for large vlms using multi-structural internal representations. *arXiv preprint arXiv:2608.10835*, 2026.

Noam Shazeer. Fast transformer decoding: One write-head is all you need. *arXiv preprint arXiv:1911.02150*, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Chenxi Wang, Xiang Chen, Ningyu Zhang, Bozhong Tian, Haoming Xu, Shumin Deng, and Huajun Chen. Mllm can see? dynamic correction decoding for hallucination mitigation. In *International Conference on Learning Representations*, volume 2025, pp. 13712–13736, 2025.

Junyang Wang, Yuhang Wang, Guohai Xu, Jing Zhang, Yukai Gu, Haitao Jia, Jiaqi Wang, Haiyang Xu, Ming Yan, Ji Zhang, et al. Amber: An llm-free multi-dimensional benchmark for mllms hallucination evaluation. *arXiv preprint arXiv:2311.07397*, 2023.

Xintong Wang, Jingheng Pan, Liang Ding, and Chris Biemann. Mitigating hallucinations in large vision-language models with instruction contrastive decoding. In *Findings of the association for computational linguistics: ACL 2024*, pp. 15840–15853, 2024.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pp. 38–45, 2020.

Tsung-Han (Patrick) Wu, Heekyung Lee, Jiaxin Ge, Joseph Gonzalez, Trevor Darrell, and David Chan. Generate, but verify: Reducing hallucination in vision-language models with retrospective resampling. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 65749–65777. Curran Associates, Inc., 2025. doi: 10.52202/085713-2206.

Yu Zhang, Chuyang Sun, Kehai Chen, Xuefeng Bai, Yang Xiang, and Min Zhang. Mitigating multimodal hallucination via phase-wise self-reward. *arXiv preprint arXiv:2604.17982*, 2026.

Deyao Zhu, jun chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 18378–18394, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/50623630a2372839c078474efa6c0cb8-Paper-Conference.pdf.

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. URL https://arxiv.org/abs/2504.10479.

Yingxuan Zhuang, Jingxiao Yang, Miao Pan, Cheng Tan, Yuxiang Cai, Siwei Tan, Chen Zhi, Xuhong Zhang, Jianwei Yin, and Jintao Chen. Mitigating manifold departure: Uncertainty-aware subspace rectification for trustworthy mllm decoding. *arXiv preprint arXiv:2606.09859*, 2026.

A RETRACT PSEUDOCODE

This appendix summarizes the three components of RETRACT: offline self-distillation (Hinton et al., 2015) of the verifier, online commit–check–retract decoding, and local retraction of rejected candidates. The notation follows Sec. 3. Model-specific verifier training, layer selection, and threshold selection are detailed in Appendix B.

Algorithm 1 Offline Self-Distillation of the RETRACT Verifier

Require: Frozen VLM p_θ, training images $\mathcal{I}_{\text{train}}$, generation instruction q, readout layer ℓ
Require: Answer-token sets $\mathcal{V}_{\text{yes}}$ and $\mathcal{V}_{\text{no}}$, regularization coefficient λ, numerical constant ϵ
Ensure: Feature statistics (μ, s) and verifier parameters $\phi = (w, b_0)$

Initialize event set $\mathcal{D} \leftarrow \emptyset$
for all $I \in \mathcal{I}_{\text{train}}$ **do**
  $\mathcal{E} \leftarrow \textsc{CollectEligibleEvents}(p_\theta, I, q, \ell)$
  for all $(o_i, h_i^+) \in \mathcal{E}$ **do**
    $a_i \leftarrow \sum_{u \in \mathcal{V}_{\text{yes}}} p_\theta(u \mid I, Q(o_i))$
    $b_i \leftarrow \sum_{u \in \mathcal{V}_{\text{no}}} p_\theta(u \mid I, Q(o_i))$
    $\widetilde{r}_i \leftarrow b_i/(a_i + b_i)$
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(h_i^+, \widetilde{r}_i)\}$
  end for
end for
Compute feature statistics (μ, s) from $\{h_i^+ : (h_i^+, \widetilde{r}_i) \in \mathcal{D}\}$
Standardize each event as $z_i \leftarrow (h_i^+ - \mu)/(s + \epsilon)$
Fit $\phi = (w, b_0)$
return (μ, s, ϕ)

Algorithm 2 RETRACT Commit–Check–Retract Decoding

Require: Frozen VLM p_θ, image I, generation instruction q
Require: Verifier (ℓ, μ, s, ϕ), numerical constant ϵ, risk threshold τ, per-position retraction budget M, attempt budget $B_{\max}$
Ensure: Retained output sequence y

$(K, m, l_1) \leftarrow \textsc{Prefill}(p_\theta, I, q)$
$y \leftarrow [\,], \mathcal{S} \leftarrow \emptyset, \mathcal{B} \leftarrow \emptyset$
$t \leftarrow 1, n_t \leftarrow 0, A \leftarrow 0$
while $A < B_{\max}$ **do**
  $\hat{y}_t \leftarrow \arg\max_{v \in \mathcal{V} \setminus \mathcal{B}} l_t(v)$
  $A \leftarrow A + 1$
  if $\hat{y}_t$ is an EOS token **then**
    break
  end if
  if $A = B_{\max}$ **then**
    $y \leftarrow y \| [\hat{y}_t]$
    break ▷ Terminal candidate is not checked
  end if
  $(h_t^+, l^{\text{next}}, j_t, K, m) \leftarrow \textsc{Commit}(p_\theta, \hat{y}_t, t, K, m, \ell)$
  $o \leftarrow o(\hat{y}_t)$ if $\hat{y}_t$ maps to an object term; otherwise $o \leftarrow \perp$
  complete $\leftarrow (o \neq \perp) \wedge \textsc{IsCompleteObject}(\hat{y}_t, l^{\text{next}})$
  eligible $\leftarrow (o \neq \perp) \wedge \text{Physical}(o) \wedge \text{complete} \wedge (o \notin \mathcal{S})$
  if eligible **then**
    $z_t \leftarrow (h_t^+ - \mu)/(s + \epsilon)$
    $r_t \leftarrow \sigma(w^\top z_t + b_0)$
    if $r_t \geq \tau \wedge n_t < M$ **then**
      $(K, m) \leftarrow \textsc{LocalRetract}(K, m, j_t)$
      Discard l^{next}
      $\mathcal{B} \leftarrow \mathcal{B} \cup \mathcal{T}(o)$
      $n_t \leftarrow n_t + 1$
      continue ▷ Reselect from the saved l_t
    end if
  end if
  $y_t \leftarrow \hat{y}_t$
  $y \leftarrow y \| [y_t]$
  if $o \neq \perp \wedge$ complete **then**
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{o\}$
  end if
  $l_{t+1} \leftarrow l^{\text{next}}$ ▷ Reuse the accepted candidate forward pass
  $t \leftarrow t + 1$
  $n_t \leftarrow 0$
  $\mathcal{B} \leftarrow \emptyset$ ▷ Reset the rejection set for the next position
end while
return y

Algorithm 3 Local Retraction of a Rejected Candidate

Require: Physical KV cache K, visibility mask m, rejected physical slot j_t
Ensure: Cache state in which the rejected candidate is excluded from the effective context

procedure $\textsc{LocalRetract}(K, m, j_t)$
  Set $m_{j_t} \leftarrow -\infty$ for the current sample in every attention layer and head
  Keep all previously allocated KV entries in K
  Keep all other cache slots and visibility values unchanged
  Keep the rejected physical slot j_t allocated but permanently masked
  return (K, m)
end procedure

Table 5: Data partitions and inputs. The object-existence prompt is used offline to construct RETRACT's training targets.

Partition	Size	Input / purpose
COCO verifier training	2,000 images	Captions and object-existence questions for verifier fitting
COCO development	400 images	Caption generation for intervention selection
COCO / CHAIR test	500 images	`Please describe this image in detail.`
AMBER generative	1,004 inputs	Released generative queries, without modification
OpenCHAIR test	4,863 images	`Provide a one-sentence caption for the provided image.`

B Experimental Details

This appendix details the backbone configurations, baseline implementations, evaluation protocols, and verifier training used in our experiments. Our main quality evaluation covers CHAIR (Rohrbach et al., 2018), AMBER (Wang et al., 2023), and OpenCHAIR (Ben-Kish et al., 2024), with hallucination and content-coverage metrics reported together.

B.1 Basic Experimental Configuration

Backbones and environment. We evaluate RETRACT on Qwen2.5-VL-7B-Instruct (Bai et al., 2025), InternVL3-8B (Zhu et al., 2025), and Gemma-3-12B-IT (Gemma Team et al., 2025). Experiments are conducted on NVIDIA GeForce RTX 5090 GPUs with 32 GB memory, using Python 3.11.16, PyTorch 2.9.1+cu128 (Paszke et al., 2019), CUDA 12.8, and Transformers 4.56.2 (Wolf et al., 2020). All models use their released instruction-tuned weights, tokenizers, image processors, and chat templates, and inference uses BF16 precision with frozen backbone parameters. Qwen2.5-VL uses a fixed 672×672 input with 576 visual tokens; InternVL3 uses one 448×448 tile with 256 visual tokens; Gemma-3 uses its processor at 896×896 with 256 visual tokens. Each method retains the corresponding backbone's positional encoding and attention-mask conventions.

Data partitions. The RETRACT training, development, and CHAIR test splits contain 2,000, 400, and 500 COCO val2014 (Lin et al., 2014) images, respectively, and are disjoint by image ID. We fix the test and development partitions with seed 42, then sample the training images from the remaining pool with seed 1234. AMBER (Wang et al., 2023) and OpenCHAIR (Ben-Kish et al., 2024) are used only for evaluation. Table 5 summarizes the data and prompts for verifier development and the main quality evaluation.

COCO (Lin et al., 2014) training and development captions use the same detailed-description prompt as CHAIR. The offline verification prompt is `Is there a {word} in the image? Answer yes or no.` (Li et al., 2023), where `{word}` is the generated object's surface form.

Generation settings. Main quality experiments use greedy decoding (Holtzman et al., 2019), `max_new_tokens=256`, and random seed 1234, stopping at the model's end-of-sequence marker or the generation budget. For RETRACT, the budget counts candidate-selection attempts, including rejected candidates; we therefore report output length alongside quality metrics. Within each backbone and benchmark, methods share the evaluation images and order, task prompts, input preprocessing, and scorer. Method-specific contrastive branches and interventions are applied on top of these common inputs. Generation batch sizes are retained in the run configurations; online efficiency is measured separately at batch size one on the RTX 5090.

B.2 Baseline Configurations

We compare RETRACT with Vanilla decoding, DeCo (Wang et al., 2025), VCD (Leng et al., 2024), VISTA (Li et al., 2025), DoLa (Chuang et al., 2024), ICD (Wang et al., 2024), LCD (Manevich & Tsarfaty, 2024), CausalLens (Ji et al., 2026), TruthPrInt (Duan et al., 2025), and MGAP (Zhuang et al., 2026). We implement the published decoding mechanisms in the three backbone interfaces, following the original papers with the settings specified below. Table 6 gives the values actually used; architecture-dependent layer choices and local adaptations are described explicitly.

Vanilla and DeCo. Vanilla uses standard greedy decoding (Holtzman et al., 2019), selecting the highest-logit token without an intervention. For DeCo, we use its greedy correction procedure (Wang

Table 6: Baseline hyperparameters. Layer entries retain each implementation's indexing convention, explained in the text.

Method	Configuration
Vanilla	Greedy decoding from the original model logits
DeCo (Wang et al., 2025)	$\alpha = 0.6$, candidate top-$p = 0.9$, top-$k = 20$; Qwen/InternVL hidden-state entries 20–27, Gemma 20–28
VCD (Leng et al., 2024)	$\alpha = 1.0$, plausibility threshold $\beta = 0.1$, diffusion noise step 500
VISTA (Li et al., 2025)	VSV strength 0.05; SLA weight 0.3; inclusive hidden-state windows 20–26 for Qwen/InternVL and 34–45 for Gemma
DoLa (Chuang et al., 2024)	Fixed early decoder-block index 12 for Qwen/InternVL and 21 for Gemma; $\alpha = 0.5$, $\beta = 0.1$
ICD (Wang et al., 2024)	$\alpha = 1.0$, $\beta = 0.1$; fixed negative-role instruction prefix
LCD (Manevich & Tsarfaty, 2024)	$\alpha = 1.0$, $\beta = 0.1$; text-only contrastive branch
CausalLens (Ji et al., 2026)	$\lambda_{\text{causal}} = 0.15$; $10 <$ `layer_idx` < 20
TruthPrInt (CC-SBU) (Duan et al., 2025)	MLP $d \to 128 \to 64 \to 1$; Adam (Kingma & Ba, 2014), learning rate 10^{-3}, batch 512, 30 epochs; two backtracking attempts
MGAP (Zhuang et al., 2026)	$N = 50$, PCA rank $K = 5$, $\alpha = 0.75$, gate sensitivity $\lambda = 0.5$; paper-form update

et al., 2025), setting $\alpha = 0.6$, `threshold_top_p=0.9`, and `threshold_top_k=20`. The final-layer distribution defines a candidate set; the highest candidate probability across the selected intermediate layers determines the anchor layer and its confidence. We add the confidence-weighted anchor logits to the final logits and mask tokens outside the candidate set. Intermediate states pass through the backbone's final normalization and shared vocabulary head. The candidate layers are hidden-state tuple entries 20–27 for the two 28-layer backbones and 20–28 for the 48-layer Gemma backbone, where entry zero denotes the embedding output. Here top-p and top-k control candidate filtering, while token selection remains greedy.

VCD. Following VCD (Leng et al., 2024), we set the contrastive weight to $\alpha = 1.0$, plausibility threshold $\beta = 0.1$, and noise step 500. The original and noise-corrupted images define two branches that receive the same generated text prefix and maintain separate KV caches (Shazeer, 2019). Writing their logits as z and z^-, the decoder uses (Leng et al., 2024)

$$\begin{aligned} \widetilde{z}(v) &= (1+\alpha)z(v) - \alpha z^-(v), \\ \mathcal{V}_\beta &= \{v : p(v) \geq \beta \max_u p(u)\}, \end{aligned} \tag{8}$$

and takes the maximum over $\mathcal{V}_\beta$. The local image-noise utility uses a 1,000-step scaled-linear variance schedule from 0.00085 to 0.012, applied to the preprocessed visual tensor; the noise-step argument 500 selects array entry 499.

ICD and LCD. Following the instruction-contrast mechanism of ICD (Wang et al., 2024), we construct a second branch by prepending `You are a confused assistant who tends to ignore the image content and hallucinate freely.` to the task prompt, retaining the image in both branches. Following the language-contrast mechanism of LCD (Manevich & Tsarfaty, 2024), we contrast the multimodal branch with the same task prompt encoded without the image and its image markers. Our shared implementation uses Eq. 8 with $\alpha = 1.0$ and $\beta = 0.1$ for both methods, with separate branch caches and the same generated tokens fed to each branch. These are fixed contrastive adaptations: the ICD prefix and logit parameterization are those given here, and LCD uses a constant contrastive coefficient.

DoLa. We use a fixed-layer adaptation of DoLa (Chuang et al., 2024), contrasting the final-layer logits with an earlier normalized state projected through the same vocabulary head. The implemented score is $(1+\alpha)z_{\text{final}} - \alpha z_{\text{early}}$, with $\alpha = 0.5$ and the same $\beta = 0.1$ plausibility filter. The early decoder-block indices are 12 for Qwen/InternVL and 21 for Gemma, with zero-based indexing; these correspond to hidden-state tuple entries 13 and 22. The deeper Gemma model uses the corresponding relative-depth setting. This configuration fixes the early layer throughout generation.

VISTA. We enable both visual steering vectors (VSV) and self-logits augmentation (SLA) as described in VISTA (Li et al., 2025). For each image, multimodal and text-only prompt passes provide a layer-wise visual direction; the decoder applies the gated, norm-preserving steering operation to the MLP outputs. We use VSV strength 0.05 and SLA mixing weight 0.3 for our backbones. SLA averages the intermediate vocabulary logits over the inclusive windows 20–26 for Qwen/InternVL and 34–45 for Gemma, then mixes this average with the final logits using weights 0.3 and 0.7. The steering strength and layer windows are the backbone-adapted settings used in our runs.

CausalLens. We use the visual/language attention decomposition and sensitivity normalization of CausalLens (Ji et al., 2026), with intervention strength $\lambda_{\text{causal}} = 0.15$. The intervention applies to zero-based decoder-block indices 11–19 on all three backbones, following the strict condition $10 < \texttt{layer_idx} < 20$. Visual spans are determined from each backbone's image-token layout. The visual/language mixing weight is computed dynamically from their representation energies.

TruthPrInt adaptation. Following the supervision and classifier design of TruthPrInt (Duan et al., 2025), we use 3,439 CC-SBU-Align (Zhu et al., 2024) image–reference pairs. Generated object mentions receive reference-based labels through the released CHAIR/COCO synonym mapping (Rohrbach et al., 2018). For each backbone, we fit a $d \to 128 \to 64 \to 1$ ReLU MLP on pre-token hidden states, using binary cross-entropy, Adam (Kingma & Ba, 2014) with learning rate 10^{-3}, batch size 512, and 30 epochs. Features are centered by the training mean and individually L_2-normalized; the image-disjoint fitting/validation split is 80/20 with seed 42, after balancing the two event classes. The classifier reads layer 14 for Qwen/InternVL and layer 24 for Gemma. At each epoch, its decision threshold is the 99th percentile of non-hallucinated validation scores; we retain the epoch with the highest true-positive rate, breaking ties by the lower false-positive rate. The resulting raw-logit thresholds are 1.8535, 3.4593, and 2.2661, respectively. Inference uses the `previous_bt` sentence-level controller, two backtracking attempts, and 64 saved candidates per position; the batched adaptation retains the final branch when its retry budget is exhausted. We retrain the classifier per backbone, and denote this configuration as TruthPrInt (CC-SBU).

MGAP configuration. We follow the subspace construction and update of MGAP (Zhuang et al., 2026), using $N = 50$ distinct calibration prompts with a black null image and PCA rank $K = 5$. We fix $\alpha = 0.75$ and gate sensitivity $\lambda = 0.5$ from the settings discussed in the paper, and reuse each backbone's calibrated subspace across evaluation benchmarks. The update uses no additional cosine cutoff or norm restoration. Calibration is performed offline with frozen model weights.

B.3 Main Benchmark Protocols

CHAIR. We generate one caption for each of the 500 held-out COCO (Lin et al., 2014) images and score the complete caption following the CHAIR evaluation protocol (Rohrbach et al., 2018). The scorer maps singular forms, synonyms, and multiword expressions to the COCO object categories (Lin et al., 2014). For image i, the reference set G_i is the union of categories in the instance annotations and human reference captions. Let m_i be the number of extracted object mentions, h_i the number unsupported by G_i, and U_i the set of distinct generated categories. We report

$$\begin{aligned} \text{CHAIR-i} &= \frac{\sum_i h_i}{\sum_i m_i}, \\ \text{CHAIR-s} &= \frac{1}{N}\sum_i \mathbf{1}[h_i > 0], \\ \text{Recall} &= \frac{\sum_i |U_i \cap G_i|}{\sum_i |G_i|}. \end{aligned} \tag{9}$$

CHAIR-i counts repeated object mentions, whereas CHAIR-s measures the proportion of captions containing at least one hallucinated object. Recall is our supplementary category-coverage measure on the same ground-truth union. Lower CHAIR scores and higher recall are preferred.

AMBER. We evaluate all 1,004 released generative inputs, retaining the benchmark's queries and annotations. Scoring follows the AMBER evaluation protocol (Wang et al., 2023), with NLTK (Loper & Bird, 2002) noun extraction and WordNet (Miller, 1995) lemmatization, followed

by the benchmark association vocabulary, safe-word rules, and semantic matching. Semantic matching uses spaCy 3.7.5 with `en_core_web_lg` 3.7.1 and similarity threshold 0.8. Let m_i and h_i denote the scorer's total and hallucinated noun-mention counts, and let c_i and g_i denote covered and annotated reference-object counts. The reported metrics retain the released accumulator convention:

$$\begin{aligned} \text{CHAIR}_{\text{AMBER}} &= \frac{\sum_i h_i}{\sum_i m_i + \epsilon}, \\ \text{Cover} &= \frac{\sum_i c_i}{\sum_i g_i + \epsilon}, \\ \text{Hal} &= 1 - \frac{\sum_i \mathbf{1}[h_i = 0]}{N + \epsilon}, \end{aligned} \tag{10}$$

where $\epsilon = 10^{-3}$. Thus CHAIR measures object-level hallucination, Cover measures reference-object coverage, and Hal measures caption-level hallucination. The first and third are minimized and Cover is maximized. We aggregate the released counts over the entire dataset before computing the ratios.

OpenCHAIR. We use all 4,863 synthetic images and their associated source captions from OpenCHAIR (Ben-Kish et al., 2024), in the fixed order `image_id=0,...,4862`. The generation prompt requests one sentence, as listed in Table 5. Following the published extraction procedure (Ben-Kish et al., 2024), we lowercase each generated caption, extract `NOUN` lemmas with spaCy `en_core_web_sm`, and retain nouns whose Brysbaert concreteness score (Brysbaert et al., 2014) is strictly greater than 4.5. The terms `painting`, `drawing`, `photo`, `picture`, `portrait`, and `photograph` receive the label `ignore`. For each remaining object mention, a text judge receives the source caption used to generate the image and the following prompt:

```
Here are a few descriptions of an image:
{reference_caption}.
Does the image contain the following object: {object}?
Answer yes/no/unsure.
The answer is:
```

The judge receives no image or candidate caption beyond the extracted object. We use GLM-5.3-Flash with `reasoning_effort=low`, temperature 0, and a maximum of 256 judge-output tokens for every method. This replaces the benchmark's Llama-2-70B-chat judge (Ben-Kish et al., 2024); the same judge configuration is used for all methods. We denote the object-level hallucination rate by OCH and the caption-level hallucination rate by HR throughout the paper. We parse an explicit leading `yes`, `no`, or `unsure`, allowing simple Markdown formatting and an optional `The answer is:` prefix; subsequent explanations do not change the label. Incomplete or unparseable judgments are retried before aggregation.

Let Y_i and H_i be the numbers of `yes` and `no` object mentions in caption i. We report

$$\begin{aligned} \text{OCH} &= \frac{\sum_i H_i}{\sum_i (Y_i + H_i)}, \\ \text{HR} &= \frac{1}{4863} \sum_i \mathbf{1}[H_i > 0]. \end{aligned} \tag{11}$$

OCH measures the fraction of judged object mentions labeled `no`; `unsure` and `ignore` are excluded from its denominator. HR measures the fraction of captions containing at least one object mention labeled `no`, retaining all 4,863 captions in its denominator. Repeated mentions retain their multiplicity, including when identical reference–object queries reuse a cached judgment. All methods are scored with the same judge and parser, and results require complete captions and judgments for the full image list.

Length and statistical reporting. For all three benchmarks, average caption length is the arithmetic mean of the number of whitespace-delimited words in the complete generated response, excluding the input prompt. We report hallucination and coverage ratios as percentages, and method differences as percentage points. OpenCHAIR results in the main table are point estimates.

B.4 RETRACT Training and Selection

Verifier training. For each backbone, RETRACT collects the first complete single-token physical-object mentions from its training captions and records their post-commit hidden states. An independent object-existence question (Li et al., 2023) to the same frozen model provides a soft risk target (Hinton et al., 2015) by normalizing the probability mass assigned to yes/no answer tokens. Training features are standardized using the training-event mean and standard deviation, with numerical stabilizer 10^{-6}. We fit a linear logistic verifier in FP32 using Newton–IRLS (Green, 1984), L_2 regularization coefficient 300, and an unpenalized intercept. Fitting runs for at most 30 iterations and stops early when the maximum absolute parameter update is below 10^{-6}. Only the verifier is optimized; the backbone remains frozen.

Model selection. Layer ℓ denotes the output of decoder block ℓ, corresponding to zero-based block index $\ell - 1$. Qwen uses layer 20 from the development candidate set $\{8, 14, 20, 24, 28\}$, and InternVL uses layer 20. For Gemma, we compare layers $\{24, 32, 40\}$ on an image-disjoint 80/20 internal split of the training images, select layer 24 by cross-entropy against the teacher's soft targets (Hinton et al., 2015), and refit on all training images. Qwen uses the fixed development threshold $\tau = 0.3$. InternVL searches $\{0.2, 0.3, 0.4, 0.5, 0.6\}$ and Gemma additionally considers 0.7 on the separate 400-image development set. We retain settings whose category recall falls by at most two percentage points relative to Vanilla, then select the lowest object-hallucination rate; these development proxies use deduplicated COCO instance categories. Both models select $\tau = 0.2$. The final (ℓ, τ) settings are therefore $(20, 0.3)$, $(20, 0.2)$, and $(24, 0.2)$ for Qwen, InternVL, and Gemma, respectively, with at most $M = 2$ retractions per logical position. These configurations are fixed before test evaluation.

B.5 Paired Analysis of Generation and Verification

We compare generation-token risk with explicit object-existence verification on object events generated by Qwen2.5-VL-7B, InternVL3-8B, and Gemma-3-12B. Both scores are obtained from the same frozen backbone and evaluated against the same reference labels within each model.

Data and sampling. The analysis uses a shared set of 10,000 images from COCO train2014 (Lin et al., 2014). We retain the 2,000 images from our initial analysis and uniformly sample 8,000 additional images from those with at least one instance annotation, excluding both the retained images and a separate set of 10,000 images used in earlier exploration. Candidate image IDs are sorted before sampling with seed 20260927. The resulting ordered image list is fixed before inference and shared across all three backbones. This experiment is a descriptive extension of the initial analysis: all captions and scores are recomputed on the complete 10,000-image set. The analysis images are separate from the COCO val2014 partitions used for RETRACT training, development, and CHAIR evaluation.

Generation settings. All models use BF16 precision, greedy decoding, and a maximum of 256 generated tokens. Qwen2.5-VL-7B uses 672×672 images with 576 visual tokens; InternVL3-8B uses a single 448×448 image tile with 256 visual tokens; and Gemma-3-12B uses 896×896 images with 256 visual tokens. Generation batch sizes are 32, 128, and 16, respectively, and verification batch sizes are capped at 64, 64, and 8. Captions that reach the generation limit remain in the analysis, with eligible events extracted from the generated prefix.

Object events and reference labels. For each image, we retain the first complete single-token mention of each physical object term that can be mapped to a COCO category. A subsequent token is used to confirm the word boundary, and end-of-sequence tokens are handled using each model's tokenizer. Distinct terms mapped to the same category remain separate events. An event is labeled unsupported if its mapped category is absent from the image's instance annotations; unsupported events constitute the positive class for AUROC and AUPRC. These labels use instance annotations alone. The evaluation therefore differs from the standard CHAIR protocol (Rohrbach et al., 2018), which combines instance annotations with reference captions and counts repeated mentions. The event sets are model-specific, while generation and verification scores are paired on exactly the same events within each model.

Table 7: **Paired comparison of generation and verification risk.** All models use the same 10,000 COCO images. Unsupported events are the positive class. Brackets report 95% image-bootstrap confidence intervals; ΔAUROC denotes verification minus generation within each model.

Model	Events	Unsupported	Generation AUROC	Verification AUROC	Paired ΔAUROC
Qwen2.5-VL-7B	22,085	4,002	0.6115 [0.6006, 0.6214]	0.8584 [0.8517, 0.8654]	+0.2470 [+0.2353, +0.2589]
InternVL3-8B	21,040	3,403	0.6093 [0.5981, 0.6205]	0.9022 [0.8964, 0.9077]	+0.2928 [+0.2811, +0.3044]
Gemma-3-12B	22,736	4,074	0.6155 [0.6060, 0.6249]	0.8717 [0.8655, 0.8776]	+0.2562 [+0.2446, +0.2668]

Table 8: **Event coverage, generation limits, and supplementary AUPRC.** Image counts refer to the shared 10,000-image set. Limit hits count captions that reach 256 generated tokens; eligible events from these captions are retained.

Model	Images with events	Images without events	Generation limit hits	Generation AUPRC	Verification AUPRC
Qwen2.5-VL-7B	8,857	1,143	1,227	0.2793	0.6465
InternVL3-8B	8,830	1,170	72	0.2623	0.6947
Gemma-3-12B	9,314	686	9,997	0.2646	0.6442

Paired risk scores. Generation-token risk is derived from the probability assigned to the emitted object token at its actual generation position, with lower probability indicating higher risk. Probabilities are recorded directly from generation logits and aligned with token IDs and positions. Explicit verification uses the question "Is there a [object] in the image? Answer yes or no." Each query receives only the image and object term. At the first answer position, we aggregate the probability mass of all affirmative and negative token variants and compute the absence risk using Equation 3. Both signals are evaluated as continuous scores without selecting a decision threshold.

Metrics and uncertainty. The primary endpoint is the within-model difference between verification AUROC and generation AUROC. Each AUROC is computed over all eligible events for that model. We estimate uncertainty using 2,000 paired image-bootstrap replicates with seed 1947. Each replicate samples 10,000 image IDs with replacement from the complete image list, including images with no eligible events, and retains all events associated with each sampled image. The same sampled image multiplicities are applied to both risk scores, and their AUROC difference is recomputed in each replicate. Confidence intervals are the 2.5th and 97.5th percentiles of the bootstrap distribution. AUPRC is reported as a supplementary metric.

Results. Table 7 shows that explicit verification improves AUROC by 0.2470–0.2928 across the three backbones, with all paired confidence intervals lying above zero. Verification also achieves higher AUPRC for every backbone (Table 8). These results establish a consistent advantage of explicit object-existence judgments over generation-token probability for identifying unsupported object events.

Generation coverage. Table 8 reports the number of images contributing eligible events and the frequency of reaching the generation limit. Gemma-3-12B reaches the 256-token limit on 9,997 of 10,000 images. The reported metrics characterize eligible object events observed within this fixed generation budget, including those from truncated captions.

Implementation checks. For each backbone, we verify that the read-only probability logger preserves the native generation token sequence. Independent verification queries use full forward passes without sharing prefix KV caches across questions. We audit image identities, reference labels, event reconstruction, and score alignment. AUROC point estimates are independently recomputed using the average-rank formulation, and the weighted bootstrap implementation is checked against explicit sample repetition.

C Additional Analysis

C.1 Inference Cost Analysis

This section analyzes the computational and memory overhead introduced by RETRACT and compares it with representative inference-time hallucination mitigation methods. Different from the empirical latency measurements reported in Section 4.3, which evaluate complete runtime under fixed hardware and software configurations, this analysis focuses on the structural source of additional computation and memory during autoregressive decoding.

The goal of this analysis is to understand how different methods modify the standard generation trajectory. Existing inference-time methods mainly introduce overhead through three mechanisms: (1) additional decoding branches under alternative visual or textual conditions (Leng et al., 2024; Wang et al., 2024; Manevich & Tsarfaty, 2024), (2) additional intermediate-layer computation or representation processing (Wang et al., 2025; Li et al., 2025), and (3) extra states required for later correction (Duan et al., 2025). RETRACT follows a different design: it performs object-level verification on candidate representations already produced during normal decoding and introduces additional computation only when a candidate requires retraction.

All analysis assumes autoregressive generation with KV-cache inference (Shazeer, 2019). We separate overhead into incremental computation and persistent memory requirements. The derivation characterizes the structural cost of each method rather than predicting exact wall-clock latency, which is measured empirically in Section 4.3.

Computational model. We first define the computational cost of standard autoregressive decoding. Let L denote the number of decoder layers, d the hidden dimension, m the intermediate dimension of the MLP block, and V the vocabulary size. The attention configuration is represented by the number of query heads n_q, key-value heads n_{kv}, and head dimension h. We define $a = n_q h$.

Under KV-cache inference, the computation of a new token consists of layer transformations, attention over cached states, and vocabulary projection (Vaswani et al., 2017; Shazeer, 2019). The layer transformation cost is

$$F = L[4d(a + n_{kv}h) + 6dm]. \tag{12}$$

where the first term corresponds to attention projections and the second term corresponds to the MLP computation. The vocabulary projection cost is

$$H = 2dV. \tag{13}$$

Given cached sequence lengths s_l at each layer, the incremental decoding cost of Vanilla generation is

$$D(\mathbf{s}) = F + 4a \sum_{l=1}^{L} s_l + H. \tag{14}$$

The model configurations used in the analysis are identical to those used in the main experiments. We use the loaded configurations of the released backbones (Bai et al., 2025; Zhu et al., 2025; Gemma Team et al., 2025) rather than approximated architectural values.

Vanilla decoding and additional computation. The above formulation provides the reference cost for comparing different decoding strategies. For methods that modify the decoding trajectory, we measure only the additional computation relative to Vanilla decoding.

For methods introducing an additional generation branch, such as ICD and LCD (Wang et al., 2024; Manevich & Tsarfaty, 2024), the additional computation can be approximated as

$$\Delta C_{\text{branch}} \approx D(\mathbf{s}). \tag{15}$$

For methods requiring additional intermediate-layer vocabulary predictions, such as DeCo and DoLa (Wang et al., 2025; Chuang et al., 2024), where K hidden states are projected into vocabulary distributions, the additional computation becomes

$$\Delta C_{\text{layer}} \approx KH. \tag{16}$$

These formulations provide a unified view of the overhead introduced by different classes of inference-time interventions.

RETRACT cost decomposition. RETRACT maintains a single decoding trajectory and performs verification on candidate representations generated during normal decoding. The candidate representation is obtained after the candidate token participates in the forward computation, allowing accepted candidates to directly continue generation without additional decoding branches.

The additional computation of RETRACT consists of verifier evaluation and rejected candidate attempts. Let N_{check} denote the number of verified candidates and R denote the number of rejected candidates. The additional computation is formulated as

$$C_{\text{RETRACT}} = O(N_{\text{check}}d) + \sum_{r=1}^{R} C_{\text{step}}^{(r)}. \tag{17}$$

The first term corresponds to the lightweight linear verifier, while the second term corresponds to candidates requiring local re-selection.

When candidate verification can be completed after the k-th decoder layer, rejected candidates do not require executing the remaining layers. The corresponding early-stop computation is

$$E_k(\mathbf{s}) = k[4d(a + n_{kv}h) + 6dm] + 4a\sum_{l=1}^{k} s_l. \tag{18}$$

The normalized additional computation can therefore be approximated as

$$\frac{\Delta C_{\text{RETRACT}}}{T} \simeq \rho E_k + 2d\kappa, \tag{19}$$

where $\rho = R/T$ denotes the rejected candidate ratio and $\kappa = N_{\text{check}}/T$ denotes the verified candidate ratio.

This decomposition explains the efficiency property of RETRACT: the verifier introduces only lightweight computation, while additional decoding computation is concentrated on positions where candidate rejection occurs rather than being applied to every generated token.

Comparison with alternative decoding strategies. We analyze representative inference-time hallucination mitigation methods under the same computational framework.

Contrastive decoding methods. VCD, ICD, and LCD introduce additional conditional generation branches. VCD maintains a perturbed visual branch (Leng et al., 2024), while ICD (Wang et al., 2024) and LCD (Manevich & Tsarfaty, 2024) introduce alternative textual or language conditions. Since each branch requires additional forward computation and KV-cache storage, their overhead grows with the number of decoding trajectories.

Intermediate-layer correction methods. DeCo and DoLa reuse a single generation trajectory but require additional intermediate representations. DeCo obtains additional token distributions from multiple hidden layers (Wang et al., 2025), while DoLa contrasts final-layer predictions with early-layer predictions (Chuang et al., 2024). Their computational overhead depends on the number of intermediate states involved.

Representation intervention methods. VISTA (Li et al., 2025) and CausalLens (Ji et al., 2026) modify hidden representations or attention pathways during inference. Although their additional

arithmetic operations may remain limited, they introduce additional state processing and intervention operations.

Rollback-based correction methods. TruthPrInt performs correction through rollback and candidate storage (Duan et al., 2025). Therefore, its memory requirement depends on the amount of historical information retained for future recovery.

Compared with these approaches, RETRACT avoids maintaining an additional decoding trajectory and does not require persistent rollback states. Its overhead is mainly determined by the frequency of verification and local retraction, which is consistent with the empirical efficiency results reported in Section 4.3.

Numerical instantiation. The theoretical estimates are instantiated using the same backbone configurations and decoding trajectories as the main experiments. The purpose of this analysis is to estimate the relative structural overhead introduced by each method under a unified decoding process.

For RETRACT, the verification frequency and retraction frequency are obtained from fixed decoding trajectories. Let T_i denote the number of accepted forward steps for sample i, R_i denote the number of retracted candidates, and N_i denote the number of verified candidates. We compute the aggregated verification ratio and retraction ratio as

$$\hat{\rho} = \frac{\sum_i R_i}{\sum_i T_i}, \qquad \hat{\kappa} = \frac{\sum_i N_i}{\sum_i T_i}. \tag{20}$$

The statistics are collected using the same 96 fixed inputs for each backbone and method. These trajectories are only used for estimating inference workload and are independent of the quality evaluation benchmarks.

The resulting event statistics are summarized in Table 9. RETRACT verifies only a subset of generated positions and performs retraction on a smaller number of candidates, which leads to a limited additional computation compared with full additional decoding branches.

Table 9: **Decoding event statistics used for theoretical cost estimation.** T denotes accepted forward steps, R denotes retracted candidates, and N denotes verified candidates.

Model	Method	$\sum T_i$	$\sum R_i$	$\sum N_i$
Qwen2.5-VL-7B	**RETRACT (ours)**	6500	196	538
Qwen2.5-VL-7B	TruthPrInt (Duan et al., 2025)	6121	170	109
InternVL3-8B	**RETRACT (ours)**	4080	134	401
InternVL3-8B	TruthPrInt (Duan et al., 2025)	4163	0	100
Gemma-3-12B	**RETRACT (ours)**	8884	94	564
Gemma-3-12B	TruthPrInt (Duan et al., 2025)	8958	44	118

Using these statistics, the normalized additional computation of RETRACT can be estimated by substituting the observed event ratios into the cost decomposition above. For example, for Qwen2.5-VL-7B, the estimated additional computation is

$$\frac{\Delta C_{\mathrm{RETRACT}}}{T} \simeq 0.28994 \ \mathrm{GFLOPs/token}. \tag{21}$$

The corresponding values for InternVL3-8B and Gemma-3-12B are 0.31580 and 0.11809 GFLOPs/token, respectively. Relative to Vanilla decoding, these values correspond to approximately 1.99%, 2.17%, and 0.49% additional theoretical computation.

These estimates explain the empirical latency behavior reported in Section 4.3: RETRACT introduces additional computation only for verification and selected retraction events, rather than executing an additional decoding branch for every generated token.

Memory analysis. We analyze the persistent memory introduced during decoding separately from computational cost. For a single sequence with uniform KV dimensions across layers, storing one

additional cached position requires

$$M_{\mathrm{KV}} = 2Ln_{\mathrm{kv}}hq, \tag{22}$$

where q denotes the number of bytes per element and the factor of two accounts for keys and values (Shazeer, 2019). Under BF16 inference, this corresponds to 56 KiB per position for Qwen2.5-VL-7B and InternVL3-8B, and 384 KiB for Gemma-3-12B before local-window saturation.

Branch-based methods such as VCD (Leng et al., 2024), ICD (Wang et al., 2024), and LCD (Manevich & Tsarfaty, 2024) maintain additional KV trajectories for alternative generation conditions. Their persistent KV storage therefore grows with the cached lengths of these branches.

RETRACT maintains a single decoding trajectory. Accepted candidates reuse the existing cache, while rejected candidates are excluded from subsequent attention through masking. Their physical KV entries remain allocated, as specified in Algorithm 3. For an uncompressed dynamic cache that retains all allocated positions, R rejected candidate attempts contribute the following additional KV storage at a fixed retained-prefix length:

$$\Delta M_{\mathrm{KV}} = RM_{\mathrm{KV}} = 2RLn_{\mathrm{kv}}hq. \tag{23}$$

This expression measures KV tensor storage under the assumptions above. Actual memory allocation also depends on cache preallocation and local-window retention policies.

The folded linear verifier requires $O(d)$ parameters, and the saved candidate distribution requires $O(V)$ storage. These terms are independent of sequence length, whereas masked KV entries and visibility metadata depend on the decoding history.

Our TruthPrInt adaptation stores hidden states and candidate information at rollback positions for subsequent recovery (Duan et al., 2025). Its additional persistent storage depends on the number and size of retained recovery states. Table 10 summarizes the additional states associated with the evaluated inference strategies.

Table 10: **Additional persistent memory states required by different inference strategies.** Fixed-size method parameters and temporary runtime buffers are excluded.

Method	Token-dependent additional memory
VCD (Leng et al., 2024)	Additional KV trajectory
ICD (Wang et al., 2024)	Additional KV trajectory
LCD (Manevich & Tsarfaty, 2024)	Additional KV trajectory
DeCo (Wang et al., 2025)	No additional persistent KV state
DoLa (Chuang et al., 2024)	No additional persistent KV state
VISTA (Li et al., 2025)	No additional persistent KV state
CausalLens (Ji et al., 2026)	No additional persistent KV state
TruthPrInt (Duan et al., 2025)	Rollback history states
RETRACT (ours)	Masked KV entries and visibility metadata

RETRACT's overhead comprises lightweight verification, additional forward passes for rejected candidates, and storage for their masked KV entries. The peak allocated GPU memory reported in Table 3 measures the combined runtime footprint, including model weights, cache allocation, and temporary buffers. Under the evaluated generation settings, this footprint remains close to Vanilla decoding.

C.2 Sensitivity to Readout Depth and Retraction Budget

We complement the threshold analysis in Section 4.4 by examining the verifier readout layer ℓ and the per-position retraction budget M on CHAIR500 (Rohrbach et al., 2018).

Verifier readout depth. For each decoder layer, we train an independent linear verifier using the same object events, soft verification targets, and normalization and fitting procedures. The threshold-selection procedure, decoding controller, and retraction budget remain unchanged across layers. Verifier training and default parameter selection are described in Appendix B.4.

Figure 4 shows non-monotonic variation in hallucination rates across decoder depth. Multiple readout layers reduce hallucination relative to Vanilla, with no consistent advantage from using the deepest layer. Recall varies less across layers but remains below Vanilla for all three backbones. Thus, readout depth influences the balance between hallucination reduction and object coverage.

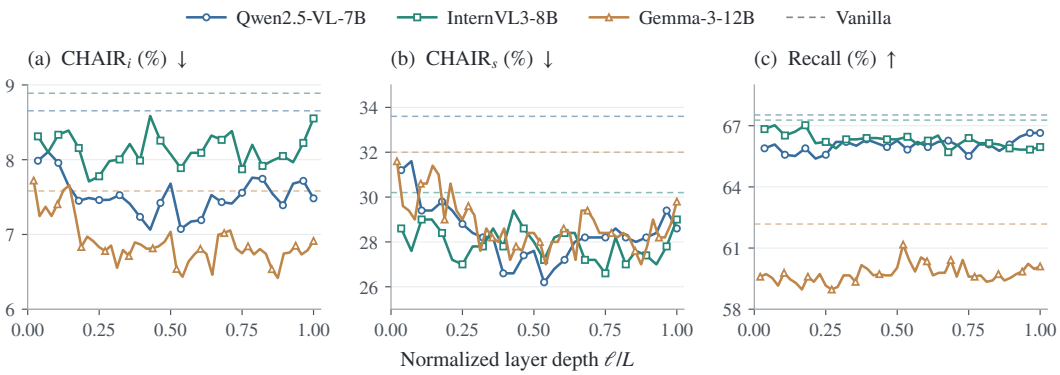

Figure 4: **Sensitivity to verifier readout depth.** CHAIR-i, CHAIR-s, and object-category Recall on CHAIR500 versus normalized layer depth ℓ/L, where L is the number of decoder layers. Each point uses a separately trained verifier. Dashed lines denote the corresponding Vanilla baselines.

Retraction budget. We evaluate $M \in \{0, 1, 2, 4\}$ while fixing the verifier, risk threshold, readout layer, and generation settings. Here, M limits the number of retractions at each logical decoding position; $M = 0$ yields Vanilla outputs.

Table 11: **Sensitivity to the retraction budget** M on CHAIR500. CHAIR-i, CHAIR-s, and Recall are reported in percentages. Retract/img denotes the average number of rejected candidates per image.

Model	M	CHAIR-i↓	CHAIR-s↓	Recall↑	Retract/img
Qwen2.5-VL-7B	0	8.65	33.60	67.53	0.00
	1	6.88	26.00	64.57	3.38
	2	6.94	26.00	64.25	3.96
	4	6.66	25.80	63.88	4.24
InternVL3-8B	0	8.89	30.20	67.28	0.00
	1	7.81	27.00	65.83	2.74
	2	7.48	25.60	66.21	3.17
	4	7.31	24.80	65.83	3.40
Gemma-3-12B	0	7.58	32.00	62.18	0.00
	1	6.85	29.40	60.54	1.65
	2	6.61	27.80	60.23	1.97
	4	6.62	28.00	60.35	2.12

Table 11 shows that allowing one retraction per position already reduces both CHAIR metrics across all three backbones. Increasing M from 2 to 4 further reduces hallucination on Qwen2.5-VL-7B and InternVL3-8B, accompanied by lower Recall and more retractions, while Gemma-3-12B shows no further hallucination reduction. The default budget $M = 2$ retains most of the hallucination reduction achieved with $M = 4$, with fewer retractions on every backbone.

Inference batch configuration. The sensitivity analyses and component ablations use different inference batch configurations. Within each parameter sweep, the inference batch size is fixed, and the Vanilla baseline is evaluated under the same configuration. Results at the default parameter value are obtained from the corresponding sweep runs. The batch sizes, verifier checkpoints, and fixed decoding parameters for each sweep are recorded in the experiment configurations.

C.3 Ablation Study of Verification Mechanism

This section provides detailed settings for the ablation experiments in Section 4.5. We analyze three components of RETRACT: (1) the representation position used by the verifier, (2) the source of verification signals, and (3) the handling of rejected candidates in KV cache.

All ablation experiments use the complete CHAIR500 benchmark (Rohrbach et al., 2018) with the same ordered image list. Within each backbone, we keep model parameters, image preprocessing, prompts, greedy decoding (Holtzman et al., 2019), generation budget, verifier training procedure, and retraction controller unchanged unless explicitly specified.

Representation position ablation. The representation ablation compares two verification states: pre-commit and post-commit representations.

The pre-commit variant uses the hidden state before the candidate token is written into the sequence:

$$h_t^{pre} = h_{t-1}. \tag{24}$$

The post-commit variant uses the candidate-specific hidden representation after the candidate participates in model computation:

$$h_t^{post} = F_\theta(c_t, \hat{y}_t). \tag{25}$$

Both variants share the same training events, soft verification targets, linear verifier architecture, threshold selection procedure, and online correction mechanism. The only difference is the representation provided to the verifier.

The motivation is that pre-commit states mainly encode the shared generation context, while post-commit states additionally incorporate the candidate object together with visual evidence and linguistic context.

Verification signal ablation. To evaluate the contribution of the learned verification signal, we replace the RETRACT verifier with three alternative signals.

Token confidence measures uncertainty from the selected candidate probability:

$$r_{\text{conf}} = 1 - p(\hat{y}_t | I, y_{<t}). \tag{26}$$

Predictive entropy measures uncertainty over the complete next-token distribution:

$$r_{\text{entropy}} = -\sum_v p(v) \log p(v). \tag{27}$$

Random rejection samples rejection decisions according to a fixed probability calibrated on development trajectories.

All alternative signals use the same candidate filtering rule, token disabling mechanism, retraction budget, and context restoration procedure. Therefore, the comparison isolates the quality of the verification signal.

KV masking ablation. The KV masking ablation removes the context filtering operation applied after candidate rejection.

In the full RETRACT implementation, rejected candidates are removed from the effective attention context by masking their KV entries. This prevents future tokens from attending to unsupported objects that have already been rejected.

The no-masking variant keeps the rejected candidate KV states visible while preserving all other correction operations, including token removal, logical position recovery, and candidate re-selection.

This experiment evaluates whether local correction requires only replacing the generated token or additionally restoring a consistent generation context.

Calibration and statistics. For confidence, entropy, and random baselines, intervention rates are calibrated using development trajectories. The calibration target follows the reference intervention

frequency of the corresponding RETRACT configuration. Test samples are not used for threshold selection.

The Qwen2.5-VL-7B, InternVL3-8B, and Gemma-3-12B configurations use thresholds $\tau = 0.30$, $\tau = 0.20$, and $\tau = 0.20$, respectively. The maximum number of retractions per logical position is fixed to $M = 2$.

All reported metrics are computed on the complete CHAIR500 benchmark without sample filtering. Random rejection results are averaged over three fixed seeds.

Inference batch configuration. The main comparison in Table 1 and the ablation study in Table 4 use different inference batch sizes. Within each backbone, all ablation variants use the same inference batch size. Vanilla and full RETRACT are evaluated separately under the respective batch configurations. Batch-dependent numerical variation in BF16 inference can affect greedy decoding trajectories and the resulting evaluation scores. We therefore assess component effects relative to the full RETRACT result within Table 4. The inference batch sizes used for each run are recorded in the corresponding experiment configurations.