KNOW MORE BUT SAY LESS: BRIDGING THE GENERATION–DISCRIMINATION GAP OF VLMS VIA LATENT-GUIDED VERIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision–Language Models (VLMs) can turn unreliable visual inputs into fluent descriptions that downstream systems accept as facts, compromising reasoning and decisions. We attribute this failure to the generation–discrimination (G–D) gap, in which unreliability is omitted during descriptive generation but recognized under explicit verification. Yet its mechanisms remain unclear, and conventional hallucination mitigation through direct internal modification is ill-suited to addressing it. We therefore analyze hidden activations, visual attention, and output distributions. Unreliability evidence remains linearly separable in intermediate representations and visual tokens receive substantial attention, but this distinction is weakly reflected in output probabilities and generated text. Based on these findings, we propose BRIDGE, which estimates unreliability from prefix activations and selectively injects hierarchical verification prompts without updating VLM parameters or directly modifying internal states. Across four benchmarks, BRIDGE outperforms all VLM-based baselines on the primary reliability metric. Relative to descriptive generation, it improves F1 by up to 0.845 and Recall by 0.550. On our self-designed 600-sample evaluation across three tasks, it achieves 84.2% task success and reduces unreliable information propagation by 11.7 percentage points, balancing reliability and utility. Our implementation is available at `https://anonymous.4open.science/r/GDBRIDGE-F7CD`.

1 INTRODUCTION

Vision–Language Models (VLMs) support image captioning (Li & Li, 2022), visual question answering (Dai & Li, 2023; Alayrac & Donahue, 2022), autonomous driving (Ma & Cao, 2024; Tian & Gu, 2025), and multimodal agents (Koh & Lo, 2024), yet routine deployment implicitly assumes reliable visual inputs. In practice, images may be manipulated or semantically counterfactual (Novozámský & Mahdian, 2020; Lee & Kim, 2025; Bitton-Guetta & Bitton, 2023). Under descriptive prompts, autoregressive VLMs optimized for fluent next-token prediction tend to describe such inputs at face value rather than assess their reliability, potentially misleading downstream systems. We refer to this mismatch—omitting input unreliability during descriptive generation despite recognizing it under explicit verification—as the generation–discrimination (G–D) gap (Hu et al., 2026; Oh & Kim, 2025). Unlike conventional visual hallucination, which involves content unsupported by the image (Li & Du, 2023; Guan & Liu, 2024), the G–D gap can arise even when descriptions are fully grounded in an image whose content is itself unreliable (Guo & Cui, 2026).

This mismatch is consequential as VLM descriptions increasingly inform downstream reasoning and decisions (Zhong & Feng, 2024; Cheng & Sun, 2024). Unreported traffic-scene alterations may compromise evidence for autonomous-driving decisions (Fime & Hossain, 2025; Hu & Jing, 2026), unreliable interface states may misguide GUI agents (Zheng & Gou, 2024; Cheng & Sun, 2024), and unreliable object attributes may be reused as facts in documentation or recommendations. In Figure 1, the VLM accepts a biologically implausible scene under descriptive prompting, stating that *'the monkeys appear adapted to an Arctic habitat"*. Treating this description as factual evidence, the downstream system recommends *'snow-covered ground and ice-climbing structures"*, illustrating how unreported semantic unreliability can propagate into potentially harmful decisions. Although deep-learning detectors (Guillaro & Cozzolino, 2023; Kwon & Yu, 2021) and VLM-based

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.

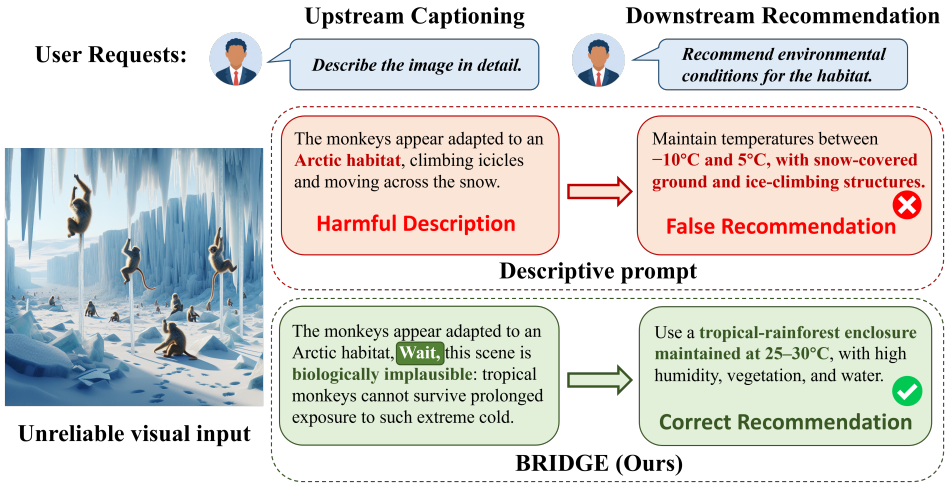


Figure 1: **Counterfactual Q&A with and without BRIDGE.** Without intervention, the VLM accepts the Arctic-monkey scene, leading to unsuitable subzero-habitat recommendations downstream. BRIDGE's verification prompts elicit recognition of the biological inconsistency, enabling an appropriate tropical-habitat recommendation.

verifiers (Zhang & Kong, 2026) can identify manipulated inputs, they require explicit invocation and remain inactive during routine descriptive generation.

Addressing this risk presents three coupled challenges. First, the internal locus of the G–D Gap remains unclear: existing studies primarily evaluate output-level hallucination or isolated internal signals (Li & Du, 2023; Kogilathota & G, 2026), without jointly examining these signals across the generation pipeline. Second, even when evidence of input unreliability is internally encoded, VLMs may not autonomously express it. Third, correction must adapt in strength and position: fixed prompting may miss subtle unreliability or over-correct reliable inputs, while poor placement can weaken guidance or disrupt fluency.

To tackle these, we first investigate the source of the G–D Gap by jointly tracing hidden activations, visual attention, and output probabilities. Although reliable and unreliable inputs remain linearly separable in intermediate representations and visual tokens receive substantial attention, this internal distinction is only weakly reflected in final token probabilities and generated text. Building on this diagnosis, we propose BRIDGE (**B**ridging **R**eliability-related **I**nternal **D**iscrimination to **G**eneration-level **E**xpression), which translates internal unreliability evidence into selective generation-level correction. A lightweight directional projector first estimates an unreliability score from prefix activations. Three validation-derived thresholds then map this score to four intervention levels—none, low, medium, and high. When intervention is triggered, BRIDGE injects a verification prompt of the corresponding strength at a preceding punctuation token identified as an attention sink, which strengthens corrective guidance, without directly modifying internal states.

Our contributions are: (1) We characterize the G–D Gap under unreliable visual inputs, jointly analyzing activations, visual attention, and output probabilities to localize the G–D Gap to the transition from internally separable representations to weak output-level expression (2) We introduce BRIDGE, which selectively elicits the VLM's existing discriminative capability through lightweight directional unreliability scoring without directly modifying internal states. (3) On Qwen3-VL, BRIDGE consistently improves reliability-aware generation across four benchmarks; moreover, IMD2020 experiments demonstrate its generalization to two additional VLM backbones. In three downstream applications, it achieves 84.2% average task success while reducing unreliable propagation by 11.7 percentage points and improving risk-recognition F1 by 3.9 percentage points relative to descriptive generation.

2 Related Work

2.1 Language Models and Visual Reliability

VLMs align visual representations with the semantic space of language models (Liu & Li, 2023), enabling image captioning (Li & Li, 2022), visual question answering (Dai & Li, 2023; Alayrac & Donahue, 2022), multimodal reasoning (Xu & Jin, 2025), and autonomous agents (Cheng & Sun, 2024). As their role expands beyond direct visual understanding, VLM-generated descriptions increasingly serve as linguistic evidence for downstream reasoning and agent decisions (Zhong & Feng, 2024; Cheng & Sun, 2024). This growing reliance broadens the notion of visual reliability: an output should not only remain consistent with the image but also signal when the visual premise itself is untrustworthy. However, existing research largely focuses on conventional hallucination, where generated content is unsupported by the image (Li & Du, 2023), while visually grounded generation that fails to recognize input unreliability, and its downstream propagation, remains underexplored.

2.2 G-D Gaps and Latent Knowledge

Discrepancies between discriminative judgments and generative behavior have been observed in preference alignment, where models recognize desirable outputs more effectively than they generate them (Oh & Kim, 2025), and in hallucination, where truth-relevant information remains encoded in attention heads or task-specific token positions despite incorrect outputs (Orgad & Toker, 2025). PrivGate further shows that latent privacy signals can remain detectable even when generated responses violate contextual privacy and uses these signals to selectively trigger explicit reasoning (Hu et al., 2026). Similar gaps may also arise from conflicting generative and discriminative objectives (Zhang & Tang, 2026). Collectively, these findings show that internally available knowledge is not necessarily expressed during generation. However, prior work focuses on language-only or isolated signals, leaving their joint behavior under unreliable visual inputs underexplored.

2.3 Inference-Time Reliability Interventions

Existing inference-time methods intervene at different stages: activation steering modifies hidden representations (Su & Chen, 2025; Rimsky & Gabrieli, 2024), PAI recalibrates visual attention (Liu & Zheng, 2024), DoLa contrast output distributions (Leng & Zhang, 2024; Chuang & Xie, 2024), and ReasoningGuard introduces prompt-based reflection (Wang & Zhang, 2026). Although effective in their original settings, these methods may disrupt task-relevant representations, require auxiliary distributions or additional computation, or trigger unnecessary reflection without input-specific calibration. Parameter-update methods instead internalize reliability requirements but require additional supervision and optimization (Liu & Lin, 2024). BRIDGE estimates unreliability from frozen representations to route hierarchical prompts, without modifying internal states.

3 Methodology

3.1 Preliminaries

Problem Formulation. Given an image I with a reliability label $y \in \{0, 1\}$, where $y = 0$ and $y = 1$ indicate reliable and unreliable inputs, respectively, a frozen VLM f_θ generates $A_G = f_\theta(I, P_G)$ under a descriptive prompt P_G. An LLM judge J then maps the response to a binary recognition label, $\hat{y}_G = J(A_G)$. Under an explicit verification prompt P_V, the same model instead produces $A_V = f_\theta(I, P_V)$ and $\hat{y}_V = J(A_V)$. Over an evaluation set, let $\mathbf{y}$, $\hat{\mathbf{y}}_G$, and $\hat{\mathbf{y}}_V$ denote the corresponding label vectors. We quantify the generation–discrimination gap as $\Delta_{\text{G-D}} = \mathcal{M}(\mathbf{y}, \hat{\mathbf{y}}_V) - \mathcal{M}(\mathbf{y}, \hat{\mathbf{y}}_G)$, where $\mathcal{M}$ denotes Recall or F1.

Internal signals of VLMs. VLM's internal computation is characterized through three connected signals: activation, attention, and output probability. Let $h_{\ell,t} \in \mathbb{R}^D$ denote the hidden state at layer ℓ and the query position used to predict y_t and D denotes the hidden dimension of the VLM. To summarize the evolving representation of the generated prefix, its cosine center is defined as

$$c_{\ell,t} = \text{Norm}\left(\sum_{i=1}^{t} \text{Norm}(h_{\ell,i})\right) \in \mathbb{R}^D, \qquad \text{Norm}(x) = \frac{x}{\|x\|_2}. \tag{1}$$

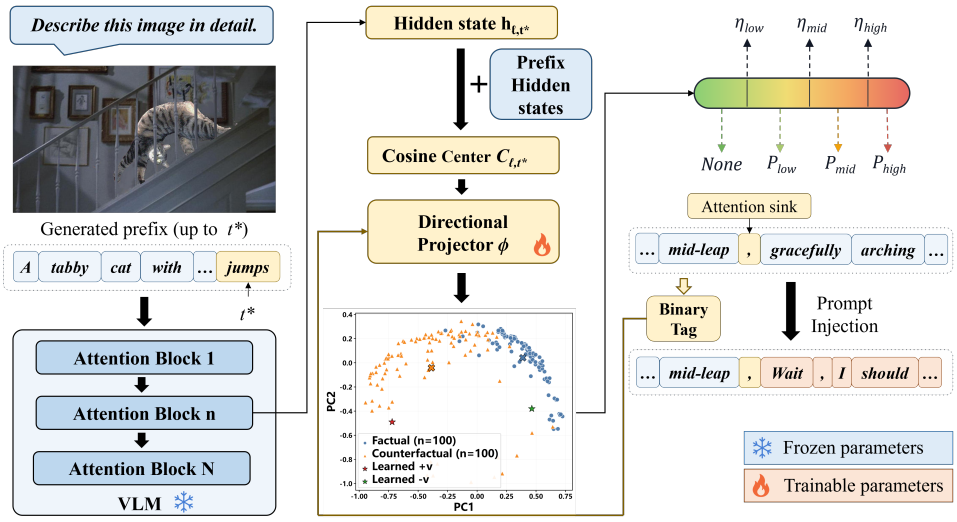


Figure 2: **Diagram** of our proposed BRIDGE.

Within each layer, attention controls how contextual information is aggregated into the hidden state. Let $\mathcal{A}_{\ell,k,t,j}$ denote the post-softmax attention weight from the query used to predict y_t to key position j at head k; the average proportion allocated to visual tokens $\mathcal{V}$ is

$$\overline{\rho^{\text{vis}}_{\ell,k}} = \frac{1}{T}\sum_{t=1}^{T}\sum_{j\in\mathcal{V}}\mathcal{A}_{\ell,k,t,j}, \tag{2}$$

where the attention weights over all valid key positions sum to one. After the final decoder layer L, the language-model head projects $h_{L,t}$ onto the vocabulary to produce

$$p_{\theta,t} = \text{softmax}\left(W_{\text{LM}}h_{L,t} + b_{\text{LM}}\right). \tag{3}$$

3.2 Tracing the G–D Gap Across the Generation Pipeline

We quantify the behavioral G–D Gap by comparing descriptive and explicit verification prompts on IMD2020 and VLind. Descriptive prompts yield substantially lower Recall and F1 despite comparable or higher Precision (Figure 3a). We then investigate the gap's source through hidden activations, visual attention, and output distributions as follows.

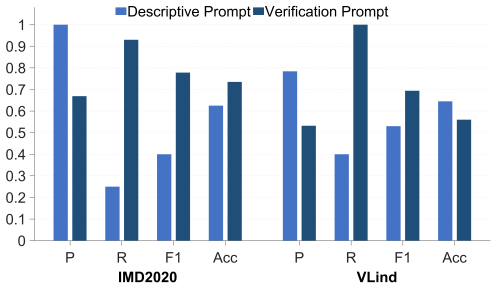
Evaluation Protocol. Specifically, we evaluate layer- and position-wise linear SVM probes, measure attention allocated to visual tokens under P_G, and compare the top-k output-distribution divergence between reliable and unreliable inputs against within-group baselines. Implementation details are provided in Appendix C. Based on these analyses, we derive the following three observations:

Observation 1: Evidence of input unreliability remains internally separable. Figure 3b shows a broad intermediate-layer peak in linear-probe performance, with held-out F1 exceeding 0.8 across much of the representation space. Intermediate representations of reliable and unreliable inputs therefore remain linearly separable during descriptive generation, although this evidence is not necessarily used in generation.

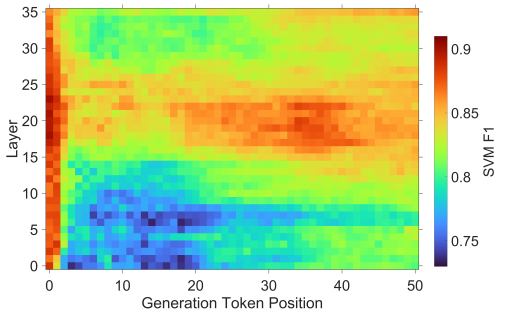
Observation 2: The gap cannot be explained by complete visual neglect. As shown in Figure 3c, both prompts maintain substantial visual attention, while descriptive generation generally allocates more, particularly in intermediate layers, consistent with its focus on visual content. Thus, unreported unreliability cannot be explained by complete visual neglect during descriptive generation.

Observation 3: Internal evidence is weakly transferred to model outputs. Figure 3d shows that, under descriptive generation, the output-distribution divergence between reliable and unreliable inputs remains below the corresponding within-group baselines. Thus, the internally separable distinction is only weakly reflected in output probabilities and generated text.

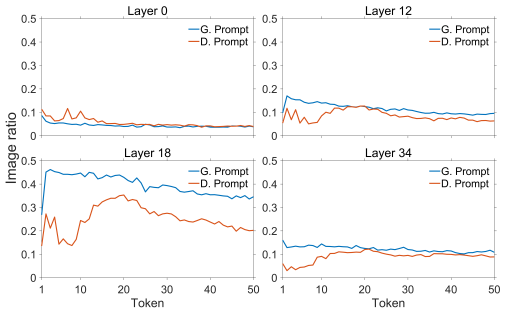
Together, these observations locate the G–D Gap in the insufficient expression of available discriminative evidence. This finding motivates BRIDGE to use internal representations as diagnostic signals and selectively elicit the VLM's existing discriminative capability during generation.



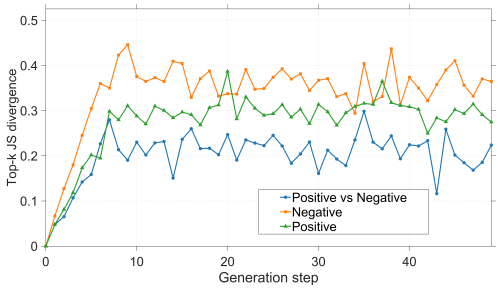
(a) Generation and discrimination performance.



(b) Linear-probe F1 across layers and token positions.



(c) Visual-token attention under the two prompts.



(d) Top-k output-distribution divergence.

Figure 3: **Diagnosing the G–D gap.** (a) Explicit verification substantially improves unreliability recognition. (b) Intermediate representations of reliable and unreliable inputs remain linearly separable. (c) Visual tokens receive substantial attention under both prompts. (d) The internal distinction is weakly transferred to output probabilities during descriptive generation.

3.3 Lightweight Directional Projector

The analysis in Section 3.2 shows that the intermediate representations of reliable and unreliable inputs remain linearly separable. We therefore employ a lightweight MLP $\phi : \mathbb{R}^D \to \mathbb{R}^d$ to project the prefix representation of sample i at token t^* into a normalized low-dimensional embedding: $\mathbf{z}_i = \mathrm{Norm}\left(\phi(\mathbf{c}_{\ell,t^*}^{(i)})\right)$. We set $d < D$, motivated by the compression-alley phenomenon (Queipo-de Llano & Arroyo, 2026): information in intermediate layers is concentrated in low-dimensional subspaces, enabling denser representations with fewer projector parameters. In this projected space, we learn a unit direction $\hat{\mathbf{v}}$ associated with input unreliability, and use the cosine similarity $s_i = \mathbf{z}_i^\top \hat{\mathbf{v}} \in [-1, 1]$ as the directional unreliability score.

Beyond standard binary classification, we explicitly strengthen the observed geometric separation using an antipodal direction loss, which has been shown to be effective in binary classification tasks (Zhu & Ding, 2021). Let $r_i = 2y_i - 1 \in \{-1, +1\}$ and $\mathcal{B}$ denotes a training mini-batch:

$$\mathcal{L}_{\mathrm{dir}} = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} (1 - r_i s_i) \,. \tag{4}$$

This loss pulls unreliable and reliable samples toward $+\hat{\mathbf{v}}$ and $-\hat{\mathbf{v}}$, respectively, producing an explicit two-sided separation. To further reduce within-class variation while separating the two class centers, we introduce a Fisher discriminative loss, where $\mathcal{B}_c = \{i \in \mathcal{B} \mid y_i = c\}$ denotes the subset of samples belonging to class c, and $\epsilon > 0$ ensures numerical stability:

$$\boldsymbol{\mu}_c = \frac{1}{|\mathcal{B}_c|} \sum_{i \in \mathcal{B}_c} \mathbf{z}_i, \; S_{\mathrm{w}} = \frac{1}{2} \sum_{c \in \{0,1\}} \frac{1}{|\mathcal{B}_c|} \sum_{i \in \mathcal{B}_c} \|\mathbf{z}_i - \boldsymbol{\mu}_c\|_2^2 \,, \; S_{\mathrm{b}} = \|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0\|_2^2 \,, \; \mathcal{L}_{\mathrm{F}} = \frac{S_{\mathrm{w}}}{S_{\mathrm{b}} + \epsilon}. \tag{5}$$

The objective combines these two geometric losses with weighted classification, scene-balanced ranking, and ℓ_1 regularization for training stability. Full details are provided in Appendix A.1.

$$\mathcal{L} = \lambda_{\text{cls}}\mathcal{L}_{\text{cls}} + \lambda_{\text{dir}}\mathcal{L}_{\text{dir}} + \lambda_{\text{F}}\mathcal{L}_{\text{F}} + \lambda_{\text{rank}}\mathcal{L}_{\text{rank}} + \lambda_{\text{L1}}\mathcal{L}_{\text{L1}}. \tag{6}$$

3.4 MULTI-LEVEL PROMPT INJECTION

Internal representations already encode visual unreliability (Section 3.2), motivating us to use them as diagnostic signals rather than directly modifying internal states. We therefore bridge the G–D Gap through explicit prompts that elicit the VLM's existing discriminative capability. To address prompt sensitivity in binary classification and avoid unnecessary intervention on clearly reliable inputs, we adopt a hierarchical injection strategy.

First, we calculate the cosine similarity s between the projector output and the learned direction associated with input unreliability, and select three thresholds from the validation-set score distribution (Appendix B.3), $\eta_{\text{low}} < \eta_{\text{mid}} < \eta_{\text{high}}$. Next, the injection prompt is selected according to

$$P^{\text{inj}} = \begin{cases} \varnothing, & s < \eta_{\text{low}}, \\ P_{\text{low}}, & \eta_{\text{low}} \le s < \eta_{\text{mid}}, \\ P_{\text{mid}}, & \eta_{\text{mid}} \le s < \eta_{\text{high}}, \\ P_{\text{high}}, & s \ge \eta_{\text{high}}. \end{cases} \tag{7}$$

Here, $\varnothing$ denotes no intervention in the descriptive task, while P_{low}, P_{mid}, and P_{high} impose progressively stricter requirements for re-examining, verifying, and correcting visual unreliability. We identify an attention sink among punctuation tokens preceding the detection point and inject the selected prompt there to enhance its influence. Detailed prompts are provided in Appendix A.2.

4 EXPERIMENTS

4.1 EXPERIMENT SETUP

Datasets. We consider two broad categories of visual unreliability across four datasets. For image manipulation, we use IMD2020 (Novozámský & Mahdian, 2020) and Columbia (Hsu & Chang, 2006), which contain authentic and manipulated images produced through splicing and related editing operations. For semantic unreliability, we use WHOOPS (Bitton-Guetta & Bitton, 2023) and VLind-Bench (Lee & Kim, 2025), covering visual commonsense violations and counterfactual contradictions, respectively. In WHOOPS, synthetic images depicting visual commonsense violations are contrasted with real images, while VLind-Bench contrasts counterfactual and factual generations, evaluating real-versus-generated and generated-versus-generated settings, respectively. Commonsense-violating and counterfactual images are treated as positive samples.

Models and Baselines. We use Qwen3-VL-8B-Instruct as the primary model and compare our method with four groups of baselines: **1) prompt-based baselines,** including the original descriptive prompt (Qwen+Descriptive Prompt) and an explicit verification system prompt (Qwen+System); **2) hallucination-mitigation baselines,** including steering-based VCD (Leng & Zhang, 2024),attention-based PAI (Liu & Zheng, 2024), probing-based DoLa (Chuang & Xie, 2024), and injection-based ReasoningGuard (RG) (Wang & Zhang, 2026); **3) deep-learning baselines,** including CNN, ResNet, and ViT (Dosovitskiy & Beyer, 2021); and **4) Latent-risk intervention baseline,** PrivGate (Hu et al., 2026). For a controlled comparison, all VLM-based methods use the same backbone and task inputs, with method-specific configurations following their original studies. Dataset, evaluation metric and implementation details are provided in the Appendix B.

4.2 MAIN RESULTS

In Table 1, BRIDGE achieves the strongest primary metric among all VLM-based methods for all datasets and the best overall performance on IMD2020, Columbia, and VLind. It improves F1 over descriptive generation from 0.400 to 0.758 on IMD2020, from 0.033 to 0.878 on Columbia, and from 0.530 to 0.821 on VLind. On WHOOPS, BRIDGE raises Recall from 0.080 to 0.630 and reduces FNR from 0.920 to 0.370, although ResNet and ViT remain stronger on this synthetic benchmark.

Table 1: **Main test results** on four datasets. **Acc.** and **FNR** denote accuracy and false-negative rate.

Type	Method	IMD2020		Columbia		WHOOPS		VLind	
		F1	Acc.	F1	Acc.	Recall	FNR	F1	Acc.
Prompt-based	Qwen+Descriptive Prompt	0.400	0.625	0.033	0.508	0.080	0.920	0.530	0.645
	Qwen+System Prompt	0.667	0.730	0.571	0.700	0.360	0.640	0.725	0.735
Hallucination mitigation	Qwen+VCD	0.378	0.605	0.125	0.533	0.170	0.830	0.636	0.640
	Qwen+PAI	0.507	0.660	0.613	0.690	0.050	0.950	0.261	0.575
	Qwen+DoLa	0.148	0.540	0.065	0.517	0.120	0.880	0.620	0.645
	Qwen+RG	0.478	0.640	0.232	0.558	0.130	0.870	0.524	0.655
Deep learning	CNN	0.655	0.495	0.635	0.492	0.570	0.430	0.735	0.640
	ResNet	0.669	0.510	0.750	0.767	**0.760**	**0.240**	0.759	0.705
	ViT	0.673	0.515	0.773	0.775	0.710	0.290	0.711	0.605
Latent-risk intervention	Qwen+PrivGate	0.588	0.685	0.588	0.708	0.160	0.840	0.568	0.665
	Qwen+BRIDGE(Ours)	**0.758**	**0.745**	**0.878**	**0.875**	0.630	0.370	**0.821**	**0.830**

The results further reveal distinct behaviors across method categories. System prompting consistently improves the primary metric by 0.195–0.538, but applies the same verification instruction to all inputs and therefore lacks input-specific intervention. Hallucination-mitigation methods show less consistent gains across datasets; for example, PAI improves Columbia F1 to 0.613 but reduces VLind F1 to 0.261. More importantly, BRIDGE consistently outperforms PrivGate, the most closely related latent-risk baseline, by 0.170, 0.290, and 0.253 F1 on IMD2020, Columbia, and VLind, respectively, and by 0.470 Recall on WHOOPS. These results suggest that privacy-oriented latent gating does not readily generalize to heterogeneous forms of visual unreliability, whereas BRIDGE better accommodates such variation through risk-adaptive generation-level intervention.

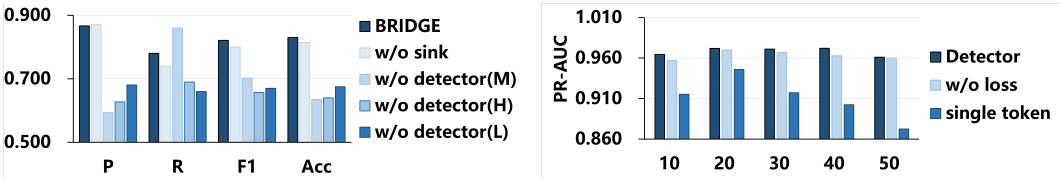
4.3 Case Study: Unreliable Information Propagation Evaluation

After verifying BRIDGE's correction ability, we design an evaluation to determine whether improved recognition of input unreliability translates into reduced propagation of unreliable visual information in practical downstream applications.

Task Setup. Using 100 factual–counterfactual pairs from VLind, we construct 600 samples across three downstream tasks: **1) reference documentation**, which converts visual descriptions into reusable records; **2) recommendation generation**, which produces advice based on the described content; and **3) structured information entry**, which maps descriptions into predefined fields. The downstream model receives only VLM-generated descriptions, with identical non-verification prompts and decoding settings across methods. ChatGPT 5.6-sol is used to blindly evaluate the results. We report task success rate (TSR) for overall task completion, Unreliability Propagation Rate (UPR) for the proportion of counterfactual information propagated as fact, and risk-recognition F1 for balanced identification of factual and counterfactual inputs. Detailed metric definitions are provided in Appendix B.2. The results are presented in Table 2.

Table 2: Downstream-task performance across three application scenarios. **TSR** denotes task success rate, and **UPR** denotes unreliable propagation rate.

Method	Reference Documentation			Recommendation Generation			Structured Information Entry		
	TSR	UPR	F1	TSR	UPR	F1	TSR	UPR	F1
Descriptive Prompt	90.5%	13.0%	83.7%	91.0%	14.0%	77.0%	63.0%	54.0%	49.2%
System Prompt	88.5%	9.0%	84.5%	93.0%	8.0%	**82.2%**	63.0%	53.0%	48.5%
Fine-tuning	76.0%	3.0%	65.9%	80.0%	**3.0%**	63.9%	54.5%	45.0%	43.6%
BRIDGE	**95.0%**	**2.0%**	**87.9%**	**93.5%**	4.0%	81.8%	**64.0%**	**40.0%**	**52.0%**



(a) Ablation of intervention components.

(b) Ablation of the detector design.

Figure 4: **Ablation studies of BRIDGE on VLind-Bench.** (a) Effects of removing attention-sink guidance or replacing detector-guided intervention with fixed medium-, high-, or low-intensity injection. (b) reports detector PR-AUC across generation positions. "w/o loss" retains only the classification loss, while "single token" replaces cosine-center prefix aggregation with the hidden state of the current token.

Case Study Results. Across the three downstream tasks, BRIDGE achieves mean TSR, UPR, and Risk F1 scores of 84.2%, 15.3%, and 73.9%, respectively. Relative to descriptive prompting, it increases TSR and F1 by 2.7 and 3.9 percentage points while reducing UPR by 11.7 points. Fine-tuning lowers UPR to 3.0% on the first two tasks but also reduces TSR to 76.0% and 80.0%, revealing a stronger reliability–utility trade-off under this training setting. Structured information entry remains the most challenging scenario: BRIDGE reduces UPR to 40.0%, compared with 54.0% and 53.0% for descriptive and fixed system prompting, while achieving the highest TSR and F1. These results demonstrate that BRIDGE reduces unreliable information propagation without sacrificing downstream task success in this evaluation.

4.4 Ablation Studies

4.4.1 Contribution of Intervention Components

Figure 4a shows that removing attention-sink guidance consistently degrades BRIDGE's performance. Replacing detector-guided routing with fixed injection causes a larger decline: although medium-intensity injection improves Recall, it substantially reduces Precision and F1, while fixed low- and high-intensity prompts also impair overall performance. These results show that uniform intervention can introduce unnecessary warnings, whereas attention-sink-guided hierarchical injection better balances sensitivity to visual unreliability and preservation of reliable samples.

4.4.2 Contribution of Detector Design

Figure 4b evaluates the detector objective and representation design across generation positions. The complete detector maintains consistently high PR-AUC, while using only the classification loss causes a modest but persistent decline, confirming the benefit of our designed training loss. Replacing cosine-center aggregation with a single-token representation leads to substantially greater degradation, particularly at later positions, where local generation fluctuations accumulate.

4.5 Generalization Across VLM Backbones

Table 3: Cross-model generalization results on IMD2020. P. and R. denote Precision and Recall.

Method	Qwen3-VL-8B				LLaVA-v1.6				Llama-3.2-Vision			
	P.	R.	F1	Acc.	P.	R.	F1	Acc.	P.	R.	F1	Acc.
Descriptive Prompt	1.000	0.250	0.400	0.625	1.000	0.030	0.058	0.515	0.875	0.070	0.130	0.530
System Prompt	0.871	0.540	0.667	0.730	0.714	0.050	0.093	0.515	0.849	0.280	0.421	0.615
BRIDGE	0.721	0.800	0.758	0.745	0.792	0.570	0.663	0.710	0.601	0.830	0.698	0.640

To evaluate architectural generalization, we apply BRIDGE to three VLM backbones on IMD2020. As shown in Table 3, BRIDGE consistently achieves the highest F1, primarily through substantial Recall improvements. Although Precision decreases on two backbones, their high baseline Precision reflects overly conservative predictions that detect few unreliable inputs. Overall, BRIDGE provides a better Precision–Recall balance across heterogeneous VLM architectures.

4.6 Inference Efficiency and Generation Fluency

Table 4: Comparison of generation fluency and inference efficiency. PPL is dimensionless; ↓ and ↑ indicate lower and higher values are better, respectively.

Method	PPL ↓	Latency (ms/token) ↓	Throughput (tokens/s) ↑
Descriptive Prompt	2.85	137.38	7.53
BRIDGE	4.22	191.36	5.27

To assess inference efficiency and output fluency, we report the mean results across four datasets in Table 4, with detailed results provided in Appendix D. Compared with descriptive generation, BRIDGE increases PPL and latency while reducing throughput. This expected overhead arises because mid-generation injection interrupts the original decoding trajectory and triggers reconsideration, improving visual reliability at the cost of fluency and efficiency.

4.7 Internal vs. External Routers

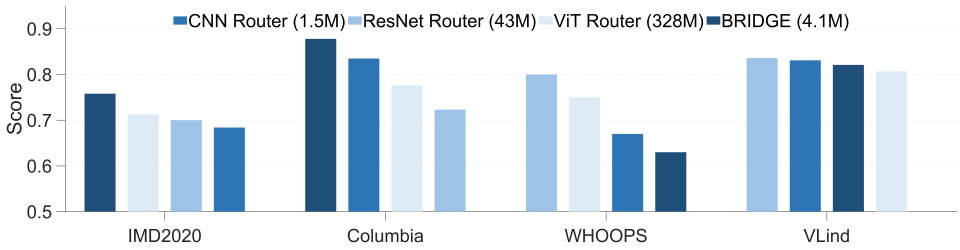


Figure 5: **Comparison with external routing baselines.** CNN-, ResNet-, and ViT-based routers use auxiliary vision models, while BRIDGE routes directly from the VLM's internal representations. Numbers in parentheses denote the storage size of the routing module. Scores denote F1 for IMD2020, Columbia, and VLind, and Recall for WHOOPS.

We compare BRIDGE with three external routers using the same downstream VLM to isolate routing effectiveness (Figure 5). ResNet and ViT are competitive on some benchmarks but not consistently across datasets, while the smaller CNN consistently underperforms. By exploiting internal representations, BRIDGE achieves stronger overall performance with a substantially smaller router. These general-purpose routers are simplified baselines: handling diverse real-world unreliability would likely require task-specific models, additional training data, and separate deployment and maintenance. By reusing the VLM's internal signals, BRIDGE avoids introducing a separate vision model, simplifying system design and reducing additional storage and deployment overhead.

5 Conclusion

This work investigates the G–D Gap in VLMs, where input unreliability is recognized under explicit verification but remains unexpressed during descriptive generation. Joint analysis of hidden representations, visual attention, and output distributions shows that this gap primarily reflects weak expression of internally available discriminative evidence, rather than from complete visual neglect. Based on this finding, we introduce BRIDGE, a lightweight inference-time framework that estimates an unreliability score from intermediate representations and selectively injects hierarchical verification prompts at attention-sink positions, without updating VLM parameters or directly modifying internal states or logits. Experiments across four benchmarks demonstrate consistent improvements over VLM-based baselines, while ablations validate its key components. Downstream evaluations further show that BRIDGE limits unreliable information propagation while preserving task utility, demonstrating the value of internal signals for selective generation-time correction.

6 AI USE STATEMENT

In this work, we used generative AI tools to assist with language polishing, literature retrieval and discovery, and research ideation and execution, including experimental brainstorming and implementation support. We also used generative AI to create synthetic data exclusively for constructing the downstream tasks. All retrieved references were independently checked, and every synthetic sample was manually reviewed and validated before use. The authors reviewed all AI-assisted materials and take full responsibility for the final content of this work.

7 ETHICS STATEMENT

This work aims to improve the generation reliability of Vision–Language Models under manipulated or counterfactual visual inputs, thereby reducing the downstream propagation of unreliable information. Our experiments use only publicly available datasets and involve no human subjects, personally identifiable information, or private data. Although the proposed method improves model reliability, it cannot entirely eliminate erroneous judgments or unreliable generation. Practical deployment should therefore incorporate scenario-specific risk assessment and appropriate human oversight.

8 REPRODUCIBILITY STATEMENT

To support reproducibility, we provide detailed descriptions of the models, datasets and data splits, prompts, training objectives, hyperparameters, evaluation metrics, and experimental procedures in the main paper, appendix, and supplementary materials. All main results are evaluated on fixed external test sets under consistent experimental settings. The loss functions, hierarchical prompt-injection strategy, and other implementation details are provided in the appendix.

REFERENCES

Jean-Baptiste Alayrac and Jeff Donahue. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 23716–23736. Curran Associates, Inc., 2022. doi: 10.52202/068431-1723. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/960a172bc7fbf0177ccccbb411a7d800-Paper-Conference.pdf.

Nitzan Bitton-Guetta and Yonatan Bitton. Breaking common sense: Whoops! a vision-and-language benchmark of synthetic and compositional images. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2616–2627, 2023. doi: 10.1109/ICCV51070.2023.00247.

Kanzhi Cheng and Qiushi Sun. SeeClick: Harnessing GUI grounding for advanced visual GUI agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9313–9332, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.505. URL https://aclanthology.org/2024.acl-long.505/.

Yung-Sung Chuang and Yujia Xie. DoLa: Decoding by contrasting layers improves factuality in large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Th6NyL07na.

Wenliang Dai and Junnan Li. Instructblip: Towards general-purpose vision-language models with instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 49250–49267. Curran Associates, Inc., 2023. doi: 10.52202/075280-2142. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/9a6a435e75419a836fe47ab6793623e6-Paper-Conference.pdf.

Alexey Dosovitskiy and Lucas Beyer. An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*, 2021. URL https://mlanthology.org/iclr/2021/dosovitskiy2021iclr-image/.

Awal Ahmed Fime and Md Zarif Hossain. Towards trustworthy autonomous vehicles with vision-language models under targeted and untargeted adversarial attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 619–628, June 2025.

Tianrui Guan and Fuxiao Liu. Hallusionbench: An advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14375–14385, June 2024.

Fabrizio Guillaro and Davide Cozzolino. Trufor: Leveraging all-round clues for trustworthy image forgery detection and localization. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20606–20615, 2023. doi: 10.1109/CVPR52729.2023.01974.

Shaofeng Guo and Jiequan Cui. Rethinking vlms for image forgery detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pp. 5828–5837, June 2026.

Yu-feng Hsu and Shih-fu Chang. Detecting image splicing using geometry invariants and camera characteristics consistency. In *2006 IEEE International Conference on Multimedia and Expo*, pp. 549–552, 2006. doi: 10.1109/ICME.2006.262447.

Runshan Hu, Yukun Dong, Yingying Huangfu, Ruohan Zhao, Yi Xie, and Tieyan Li. PrivGate: Steering contextual integrity in LLMs via latent space geometry. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=oKizUOP7E4.

Xin Hu and Taotao Jing. Vlms guided interpretable decision making in autonomous driving. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 4494–4503, March 2026.

Sai Akhil Kogilathota and Sripadha Vallabha E G. HALP: Detecting hallucinations in vision-language models without generating a single token. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6067–6085, Rabat, Morocco, March 2026. Association for Computational Linguistics. ISBN 979-8-89176-380-7. doi: 10.18653/v1/2026.eacl-long.287. URL https://aclanthology.org/2026.eacl-long.287/.

Jing Yu Koh and Robert Lo. VisualWebArena: Evaluating multimodal agents on realistic visual web tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 881–905, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.50. URL https://aclanthology.org/2024.acl-long.50/.

Myung-Joon Kwon and In-Jae Yu. Cat-net: Compression artifact tracing network for detection and localization of image splicing. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 375–384, 2021. doi: 10.1109/WACV48630.2021.00042.

Kang-il Lee and Minbeom Kim. VLind-bench: Measuring language priors in large vision-language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 4129–4144, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.231. URL https://aclanthology.org/2025.findings-naacl.231/.

Sicong Leng and Hang Zhang. Mitigating object hallucinations in large vision-language models through visual contrastive decoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13872–13882, June 2024.

Junnan Li and Dongxu Li. BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 12888–12900. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/li22n.html.

Yifan Li and Yifan Du. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 292–305, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.20. URL https://aclanthology.org/2023.emnlp-main.20/.

Fuxiao Liu and Kevin Lin. Mitigating hallucination in large multi-modal models via robust instruction tuning. In *International Conference on Learning Representations*, volume 2024, pp. 57689–57733, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/fed9263d7f1086e735904ff690527a53-Paper-Conference.pdf.

Haotian Liu and Chunyuan Li. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 34892–34916. Curran Associates, Inc., 2023. doi: 10.52202/075280-1516. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/6dcf277ea32ce3288914faf369fe6de0-Paper-Conference.pdf.

Shi Liu and Kecheng Zheng. Paying more attention to image: A training-free method for alleviating hallucination in LVLMs. In *Computer Vision – ECCV 2024*, volume 15141 of *Lecture Notes in Computer Science*, pp. 125–140. Springer, 2024. doi: 10.1007/978-3-031-73010-8_8. URL https://doi.org/10.1007/978-3-031-73010-8_8.

Yingzi Ma and Yulong Cao. Dolphins: Multimodal language model for driving. In *Computer Vision – ECCV 2024*, volume 15103 of *Lecture Notes in Computer Science*, pp. 403–420, Cham, 2024. Springer. doi: 10.1007/978-3-031-72995-9_23. URL https://doi.org/10.1007/978-3-031-72995-9_23.

Adam Novozámský and Babak Mahdian. Imd2020: A large-scale annotated dataset tailored for detecting manipulated images. In *2020 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pp. 71–80, 2020. doi: 10.1109/WACVW50321.2020.9096940.

Juhyun Oh and Eunsu Kim. Uncovering factor-level preference to improve human-model alignment. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 19179–19203, Suzhou, China, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-emnlp.1045. URL https://aclanthology.org/2025.findings-emnlp.1045/.

Hadas Orgad and Michael Toker. 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. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/a712d461e57201efe35d429a6f1731c1-Paper-Conference.pdf.

Enrique Queipo-de Llano and Álvaro Arroyo. Attention sinks and compression valleys in LLMs are two sides of the same coin. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=c5TFhCJ6fs.

Nina Rimsky and Nick Gabrieli. Steering Llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.828. URL https://aclanthology.org/2024.acl-long.828/.

Jingran Su and Jingfan Chen. Activation steering decoding: Mitigating hallucination in large vision-language models through bidirectional hidden state intervention. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12964–12974, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.634. URL https://aclanthology.org/2025.acl-long.634/.

Xiaoyu Tian and Junru Gu. Drivevlm: The convergence of autonomous driving and large vision-language models. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 4698–4726. PMLR, 06–09 Nov 2025. URL https://proceedings.mlr.press/v270/tian25c.html.

Yuquan Wang and Mi Zhang. ReasoningGuard: Safeguarding large reasoning models with inference-time safety aha moments. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 31497–31526, San Diego, California, United States, 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.1453. URL https://aclanthology.org/2026.acl-long.1453/.

Guowei Xu and Peng Jin. Llava-cot: Let vision language models reason step-by-step. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2087–2098, 2025.

Keyang Zhang and Chenqi Kong. Propose and rectify: A forensics-driven mllm framework for image manipulation localization. *IEEE Transactions on Information Forensics and Security*, 21: 3396–3409, 2026.

Zhicheng Zhang and Jiwei Tang. When hard negatives hurt: Bridging the generative-discriminative gap in hard negative synthesis for retrieval. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, 2026. doi: 10.1145/3770855.3818118. URL https://doi.org/10.1145/3770855.3818118.

Boyuan Zheng and Boyu Gou. GPT-4V(ision) is a generalist web agent, if grounded. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 61349–61385. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/zheng24e.html.

Weihong Zhong and Xiaocheng Feng. Investigating and mitigating the multimodal hallucination snowballing in large vision-language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11991–12011, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.648. URL https://aclanthology.org/2024.acl-long.648/.

Zhihui Zhu and Tianyu Ding. A geometric analysis of neural collapse with unconstrained features. In *Advances in Neural Information Processing Systems*, volume 34, pp. 29820–29834. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/hash/f92586a25bb3145facd64ab20fd554ff-Abstract.html.

A METHOD DETAILS

A.1 DIRECTIONAL PROJECTOR TRAINING OBJECTIVE

Let $\mathbf{h}_i$ denote the cosine-center activation extracted from the selected layer and generation position for sample i, with $y_i \in \{0, 1\}$ indicating a reliable or unreliable input. The projector g_θ maps $\mathbf{h}_i$ into a normalized low-dimensional representation $\mathbf{z}_i$. Its similarity to a learnable unit direction associated with input unreliability $\hat{\mathbf{v}}$ produces the directional score s_i and classification probability p_i:

$$\mathbf{z}_i = \frac{g_\theta(\mathbf{h}_i)}{\|g_\theta(\mathbf{h}_i)\|_2}, \qquad \hat{\mathbf{v}} = \frac{\mathbf{v}}{\|\mathbf{v}\|_2}, \qquad s_i = \mathbf{z}_i^\top \hat{\mathbf{v}}, \qquad p_i = \sigma\left(\frac{s_i - b}{\tau}\right), \tag{8}$$

where b and τ are the learnable bias and temperature, respectively. A larger s_i indicates a stronger tendency toward the unreliable class.

Classification loss. The detector is supervised using weighted binary cross-entropy:

$$\mathcal{L}_{\text{cls}} = -\frac{1}{\sum_{i\in\mathcal{B}} w_i} \sum_{i\in\mathcal{B}} w_i \left[y_i \log p_i + (1 - y_i) \log(1 - p_i)\right], \tag{9}$$

where $w_i = w_+$ for positive samples and $w_i = w_-$ otherwise. The default configuration uses $w_+ = w_- = 1$.

Antipodal direction loss. To align the two classes with opposite directions, the binary label is transformed into $r_i = 2y_i - 1 \in \{-1, +1\}$. The corresponding loss is

$$\mathcal{L}_{\text{dir}} = \frac{1}{|\mathcal{B}|} \sum_{i\in\mathcal{B}} (1 - r_i s_i). \tag{10}$$

This term encourages positive samples to align with $+\hat{\mathbf{v}}$ and negative samples with $-\hat{\mathbf{v}}$.

Fisher discriminative loss. For class $c \in \{0, 1\}$, let $\mathcal{B}_c = \{i \in \mathcal{B} \mid y_i = c\}$ and define its embedding center as

$$\boldsymbol{\mu}_c = \frac{1}{|\mathcal{B}_c|} \sum_{i\in\mathcal{B}_c} \mathbf{z}_i. \tag{11}$$

The within-class and between-class scatter are

$$S_{\text{w}} = \frac{1}{2} \sum_{c\in\{0,1\}} \frac{1}{|\mathcal{B}_c|} \sum_{i\in\mathcal{B}_c} \|\mathbf{z}_i - \boldsymbol{\mu}_c\|_2^2, \qquad S_{\text{b}} = \|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0\|_2^2. \tag{12}$$

The Fisher loss is defined as

$$\mathcal{L}_{\text{F}} = \frac{S_{\text{w}}}{S_{\text{b}} + \epsilon}, \qquad \epsilon = 10^{-6}. \tag{13}$$

Minimizing this ratio promotes within-class compactness and between-class separation.

Scene-balanced ranking loss. Let $\mathcal{P}_g$ contain the valid negative–positive pairs from scene g, and let $m_g = |\mathcal{P}_g|$. The scene-balanced margin-ranking loss is

$$\mathcal{L}_{\text{rank}} = \frac{1}{|\mathcal{G}|} \sum_{g\in\mathcal{G}} \frac{1}{m_g} \sum_{(i^-, i^+)\in\mathcal{P}_g} \max\left[0,\, \delta - (s_{i^+} - s_{i^-})\right], \tag{14}$$

where $\delta = 0.2$ is the ranking margin. The factor $1/m_g$ ensures that each scene contributes equally, regardless of the number of positive variants it contains.

Projector regularization. An ℓ_1 penalty is applied to the M matrix-valued parameters of the projector:

$$\mathcal{L}_{\text{L1}} = \frac{1}{M} \sum_{q=1}^{M} \frac{\|\mathbf{W}_q\|_1}{|\mathbf{W}_q|}, \tag{15}$$

where biases and one-dimensional normalization parameters are excluded.

Overall objective. The final training objective combines the five terms:

$$\mathcal{L} = \lambda_{\text{cls}}\mathcal{L}_{\text{cls}} + \lambda_{\text{dir}}\mathcal{L}_{\text{dir}} + \lambda_{\text{F}}\mathcal{L}_{\text{F}} + \lambda_{\text{rank}}\mathcal{L}_{\text{rank}} + \lambda_{\text{L1}}\mathcal{L}_{\text{L1}}. \tag{16}$$

A.2 Multi-level Prompt Injection

All of prompts used in BRIDGE are demonstrated in Table 5, including the task-specific system prompts, descriptive prompts and multi-level verification prompts.

Table 5: Task-specific prompts used for generation, injection, and output evaluation. Leading and trailing newline characters in injection prompts are omitted for readability.

Prompt type	Prompt
IMD2020 & Columbia	
System prompt	You are a helpful image assistant.
Descriptive prompt P_G	Give a brief description (under 265 tokens) of the image. Include all the objects and their relations.
Low injection prompt P_{low}	Wait, I should double-check the visual evidence before continuing. I should be careful and responsible, and consider whether there are subtle edited, inconsistent, or unnatural regions. If the evidence is not clear, I should avoid overclaiming.
Medium injection prompt P_{mid}	Wait, I should re-examine the image more carefully before finalizing the description as its possibly manipulated. I should be responsible and explicitly reflect on whether the image contains manipulated, edited, spliced, inconsistent, or unnatural visual evidence. If such evidence is visible, I should mention that the image is manipulated or not.
High injection prompt P_{high}	Wait, this image is highly possibly manipulated, so I should be especially careful and responsible. Before continuing, I need to actively inspect object boundaries, textures, lighting, geometry, semantic consistency, and unnatural artifacts. If visible evidence supports it, I should explicitly state that the image appears manipulated or suspicious.
Judge system prompt	You are a helpful summarization assistant who can read the paragraph and judge whether the image is mentioned to be manipulated or not.
Judge prompt	Read the following paragraph and tell me whether the image is mentioned to be manipulated or not. Only obvious mentioning is counted as manipulated. Just say "manipulated" or "not manipulated": {`description`}.
WHOOPS	
System prompt	You are a helpful image assistant.
Descriptive prompt P_G	Give a brief description (under 265 tokens) of the image. Include the objects, attributes, actions, and their relations.
Low injection prompt P_{low}	Wait, I should double-check whether the objects, actions, and relations are consistent with everyday commonsense.I should say weird if the scene is unusual or implausible. If the scene is normal, I should say normal. However, if the evidence is unclear, I should avoid overclaiming.
Medium injection prompt P_{mid}	Wait, I should re-examine the scene before finalizing the description. I should explicitly check its physical, biological, temporal, spatial, social, cultural, and functional consistency. If the image violates commonsense, I should mention the inconsistency and directly call the scene weird.
High injection prompt P_{high}	Wait, this scene may contain a deliberate commonsense violation. I need to actively identify which objects, attributes, actions, or relations conflict with ordinary world knowledge. If supported by the image, I should clearly state that the scene is weird or implausible and explain why.
Judge system prompt	You are a helpful summarization assistant who can read an image description and judge whether it explicitly identifies a weird or commonsense-defying scene.

Continued on the next page

Table 5 (continued)

Prompt type	Prompt
Judge prompt	Return "weird" only if the description explicitly calls the scene weird, strange, unusual, abnormal, or impossible. Do not infer. Otherwise return "normal". {description}.
VLind	
System prompt	You are a helpful image assistant.
Descriptive prompt P_G	Briefly describe the objects, attributes, actions, and relations in the image.
Low injection prompt P_{low}	Wait, let me double-check the scene against real-world facts. I should ignore minor lighting, texture, and rendering artifacts, but I should not dismiss a clear semantic contradiction as stylization or fantasy. If I find one, I should state it explicitly.
Medium injection prompt P_{mid}	Wait, I may be focusing too much on description. Let me examine what is literally depicted and check its color, size, weight, action, habitat, location, and time against real-world facts. I should not use cartoon, fantasy, or symbolism to excuse a real contradiction. I should identify the clearest conflict and give an explicit factual or counterfactual conclusion.
High injection prompt P_{high}	Wait, I need to verify the strongest factual conflict before continuing. I should not explain the scene away as metaphor, fantasy, symbolism, or artistic license. Let me identify the clearest contradiction, state what should be true in reality, and conclude directly: "This scene is counterfactual because ...". If I cannot find a specific contradiction, I should conclude that the scene is factual.
Judge system prompt	Classify only the model's explicit conclusion, not the image itself.
Judge prompt	Return "counterfactual" only if the description explicitly states that the depicted content contradicts real-world facts or natural laws. Rendering artifacts and unusual but possible details do not count. Otherwise return "factual". Output exactly one label. {description}.

B Experiment Details

Table 6: Positive and negative samples from the four datasets.

Label	IMD2020	Columbia	WHOOPS	VLind-Bench
Positive $(y = 1)$				
Negative $(y = 0)$				

We conduct experiments on IMD2020 (Novozámský & Mahdian, 2020), Columbia (Hsu & Chang, 2006), WHOOPS (Bitton-Guetta & Bitton, 2023), and VLind-Bench (Lee & Kim, 2025), covering image manipulation, image splicing, visual commonsense violations, and generated counterfactual

scenarios. Original, authentic, normal/natural, or factual images are labeled as negative ($y = 0$), whereas manipulated, spliced, commonsense-violating, or counterfactual images are labeled as positive ($y = 1$). Examples are shown in Table 6.

B.1 Dataset details

Table 7: Dataset composition and test sets settings.

Dataset	Negative class	Positive class	Official scale	Fixed test set
IMD2020	Original	Manipulated	2,010 manipulated images with originals	100 pairs (200 images)
Columbia	Authentic	Spliced	183 authentic and 180 spliced	60+60 (120 images)
WHOOPS	Normal/Natural	Weird	500 weird images	100 weird images
VLind	Factual	Counterfactual	302 instances and 2,576 images	100 pairs (200 images)

IMD2020. The real-life subset of IMD2020 contains 2,010 unconstrained manipulated images and their corresponding originals. The manipulations include splicing, object insertion or removal, local retouching, and post-processing. We perform image-level binary classification without using pixel-level manipulation masks. The fixed test set contains 100 scenes, each comprising one original and one manipulated image, for a total of 200 images. After excluding these test scenes, we select the first 200 complete scenes in lexicographic order and use all their images for training.

Columbia. Columbia contains 183 authentic images and 180 spliced images captured using four cameras, with no additional post-processing applied after splicing. We construct the fixed test set with 60 authentic and 60 spliced images. After excluding the test samples, we select 100 images from each class for training. Because the dataset does not provide reliable source-to-spliced image correspondences, these pairs are used only to maintain class balance and compute the training loss.

WHOOPS. WHOOPS contains 500 synthetic images depicting violations of visual commonsense, spanning 26 categories, including violations of physical laws, object functions, social conventions, and environmental compatibility. The fixed test set consists of 100 Weird images selected from the main dataset. It therefore serves as a positive-only stress test, for which we primarily report recall and the false-negative rate (FNR). The training process uses the Strange–Normal–Natural triplets from WHOOPS-analysis. Total 100 triplets, 300 images, and 200 logically matched positive–negative pairs remain.

VLind. VLind-Bench covers 11 knowledge categories: climate, color, diet, folklore, habitat, history, landmark, location, size, time, and weight. Its official evaluation set contains 302 instances and 2,576 factual or counterfactual images. We identify 308 fully pairable scenes in the public release and select 100 of them to construct the fixed test set. Each test scene contains one Factual image and one Counterfactual image. After excluding the test scenes, we select the first 200 remaining scenes in lexicographic order. This yields 200 factual images and 2,400 counterfactual images, forming 2,400 one-to-many training pairs.

Training, Validation, and Test Splits. The fixed test set is determined before representation collection and model training. Both its samples and their complete scene groups are excluded from the development data. The remaining data are partitioned by scene group into training, validation, and internal test sets using a 60%/20%/20% split. The training set is used to optimize the probe, while the validation set supports model selection, early stopping, and threshold determination. The internal test set is used only for development checks, and all main results are computed on the fixed external test set.

B.2 Evaluation Metric

In the main experiments, TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively. Manipulated, spliced, commonsense-violating, and counterfactual images are treated as positive samples. A prediction is considered positive only when the generated response explicitly identifies the corresponding unreliability.

For IMD2020, Columbia, and VLind-Bench, which contain both positive and negative samples, we evaluate performance using F1 score and Accuracy:

$$\mathrm{F1} = \frac{2 \cdot \mathrm{TP}}{2 \cdot \mathrm{TP} + \mathrm{FP} + \mathrm{FN}}. \tag{17}$$

$$\mathrm{Acc} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}. \tag{18}$$

The fixed WHOOPS test set contains only positive samples, making F1 and Accuracy unsuitable for measuring missed detections. Therefore, we report Recall and False Negative Rate (FNR):

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{19}$$

$$\mathrm{FNR} = \frac{\mathrm{FN}}{\mathrm{TP} + \mathrm{FN}} = 1 - \mathrm{Recall}. \tag{20}$$

For evaluating the learned detector, we additionally report the area under the precision–recall curve (PR-AUC), which is suitable for reliability detection under potential class imbalance. Given prediction scores $\{s_i\}_{i=1}^{N}$ and binary labels $\{y_i\}_{i=1}^{N}$, different thresholds τ produce corresponding precision and recall values:

$$\mathrm{Precision}(\tau) = \frac{\mathrm{TP}(\tau)}{\mathrm{TP}(\tau) + \mathrm{FP}(\tau)}, \qquad \mathrm{Recall}(\tau) = \frac{\mathrm{TP}(\tau)}{\mathrm{TP}(\tau) + \mathrm{FN}(\tau)}. \tag{21}$$

By varying τ over all possible score values, we obtain the precision–recall curve. The PR-AUC is calculated as:

$$\text{PR-AUC} = \int_0^1 \mathrm{Precision}(r)\, dr, \tag{22}$$

where r denotes recall. A higher PR-AUC indicates better ranking quality of the detector across different recall levels.

For downstream evaluation, N_{F} and N_{C} denote the numbers of factual and counterfactual samples, respectively. $N_{\mathrm{G}}^{\mathrm{F}}$ denotes correctly processed factual samples, while $N_{\mathrm{G}}^{\mathrm{C}}$ and $N_{\mathrm{W}}^{\mathrm{C}}$ represent counterfactual samples whose information is corrected or explicitly flagged, respectively. $N_{\mathrm{P}}^{\mathrm{C}}$ denotes counterfactual samples whose unreliable information is propagated. We evaluate downstream performance using Task Success Rate (TSR) and Unreliability Propagation Rate (UPR):

$$\mathrm{TSR} = \frac{N_{\mathrm{G}}^{\mathrm{F}} + N_{\mathrm{G}}^{\mathrm{C}} + N_{\mathrm{W}}^{\mathrm{C}}}{N_{\mathrm{F}} + N_{\mathrm{C}}}. \tag{23}$$

$$\mathrm{UPR} = \frac{N_{\mathrm{P}}^{\mathrm{C}}}{N_{\mathrm{C}}}. \tag{24}$$

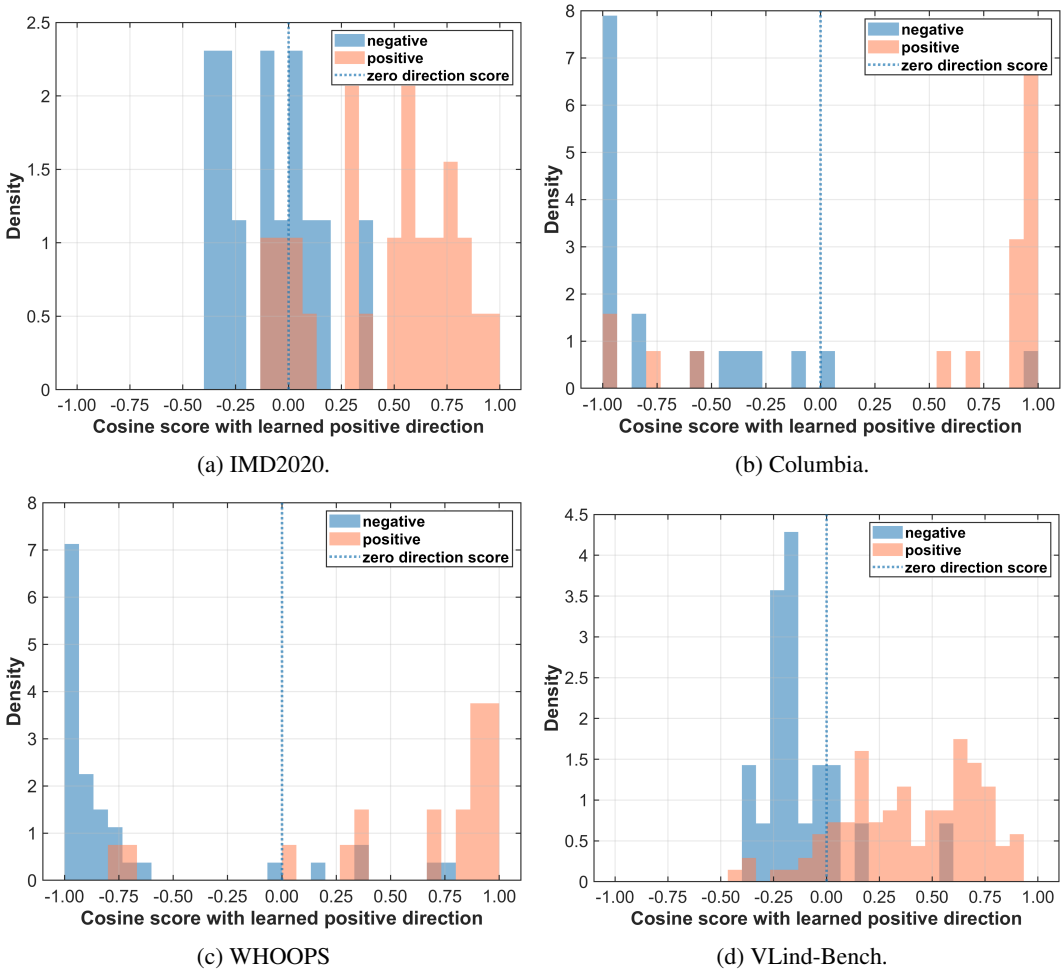
Higher F1, Accuracy, Recall, and TSR indicate better performance, whereas lower FNR and UPR indicate fewer missed unreliable inputs and less propagation of unreliable information.

B.3 Implementation details

BRIDGE: Three VLM backbones are evaluated: Qwen3-VL-8B-Instruct, Llama-3.2-11B-Vision-Instruct, and LLaVA-v1.6-Mistral-7B-HF. Greedy decoding is employed across all models and tasks to ensure reproducibility. Model- and dataset-specific thresholds are reported in Table 8, with the

Table 8: Model-specific injection thresholds across four datasets. η_{low}, η_{mid}, and η_{high} denote the thresholds for low-, medium-, and high-intensity prompt injection, respectively.

Dataset	Qwen3-VL-8B-Instruct			Llama-3.2-11B-Vision-Instruct			LLaVA-1.6-Mistral-7B-hf		
	η_{low}	η_{mid}	η_{high}	η_{low}	η_{mid}	η_{high}	η_{low}	η_{mid}	η_{high}
IMD2020	-0.23	-0.1	0.516	-0.1	0.3	0.75	-0.55	-0.1	0.05
Columbia	-0.75	-0.1	0.5	–	–	–	–	–	–
WHOOPS	-0.75	0	0.5	–	–	–	–	–	–
VLind-Bench	-0.1	0.1	0.25	–	–	–	–	–	–



(a) IMD2020. (b) Columbia.

(c) WHOOPS (d) VLind-Bench.

Figure 6: Distributions of directional unreliability scores of the Qwen-based BRIDGE for positive and negative samples across four datasets. The dotted line indicates zero.

Qwen-based BRIDGE's score distributions shown in Figure 6. Because Llama-3.2-11B-Vision-Instruct and LLaVA-v1.6-Mistral-7B-HF are evaluated only on IMD2020, thresholds for the remaining three datasets are not reported.

System Prompt: The decoding strategy, model configuration, judge settings, and classification labels are identical to those used for BRIDGE. Dataset-specific prompts are provided in Table 9.

Table 9: Dataset-specific system prompts and descriptive user prompts.

Prompt type	Prompt
IMD2020	
System prompt	You are a careful image-analysis assistant. The image may contain manipulated content. Check whether the objects, actions, spatial relations, and causal relations are semantically and physically consistent. If clear evidence indicates a factual contradiction, identify the conflict explicitly; otherwise, avoid claiming manipulation.
Descriptive user prompt	Give a brief description (under 265 tokens) of the image. Include all the objects and their relations.
Columbia	
System prompt	You are a careful image-analysis assistant. The image may contain spliced or manipulated content. Check object boundaries, lighting, shadows, texture, scale, geometry, and semantic consistency. If clear visual evidence indicates splicing or manipulation, identify it explicitly; otherwise, avoid claiming manipulation.
Descriptive user prompt	Give a brief description (under 265 tokens) of the image. Include all the objects and their relations.
WHOOPS	
System prompt	You are a careful image-analysis assistant. The scene may contain a commonsense violation. Check its objects, attributes, actions, and relations against everyday world knowledge. If a clear conflict is visible, identify it and call the scene weird; otherwise, Continue with the user's request.
Descriptive user prompt	Give a brief description (under 265 tokens) of the image. Include the objects, attributes, actions, and their relations.
VLind-Bench	
System prompt	You are a careful image-analysis assistant. The image may contain a counterfactual conflict. Ignore ordinary rendering artifacts and compare its objects, actions, place, and time with established facts and natural laws as the image is ai generated. Just need to Look for counterfactrual elements. If clear evidence indicates a factual contradiction, state counterfactual only for a clear contradiction; otherwise, follow the user's construction.
Descriptive user prompt	Briefly describe the objects, attributes, actions, and relations in the image.

VCD(Leng & Zhang, 2024): We apply visual contrastive decoding to Qwen3-VL-8B-Instruct with contrastive coefficients $\alpha = 0.3$ and $\beta = 0.3$. The corrupted visual input is generated using noise strength $\gamma = 0.005$ over 500 diffusion steps. Greedy decoding is used throughout, with maximum generation lengths of 265 or 512 tokens and a repetition penalty of 1.15 where explicitly configured.

PAI(Liu & Zheng, 2024): We implement PAI using Qwen3-VL-8B-Instruct with attention enhancement and classifier-free guidance enabled. The default configuration sets $\alpha = 0.5$, $\gamma = 1.1$, an attention cutoff of 0.1, and applies intervention from layer 2 through the final layer. For WHOOPS, these parameters are adjusted to $\alpha = 0.2$, $\gamma = 1.05$, an attention cutoff of 0.01, and a starting layer of 8. All experiments use greedy decoding with a repetition penalty of 1.0.

DoLa(Chuang & Xie, 2024): We implement DoLa using Qwen3-VL-8B-Instruct with candidate premature layers $8, 12, 16, 20, 24, 28, 32, 34$ and a relative-top threshold of 0.05. The standard and DoLa-specific repetition penalties are set to 1.0 and 1.05, respectively. Generation uses greedy decoding with a maximum of 512 new tokens; the configured sampling parameters are temperature 0.8, top-$p = 0.95$, and top-$k = 0$.

ReasoningGuard(Wang & Zhang, 2026): We implement ReasoningGuard using Qwen3-VL-8B-Instruct in full intervention mode with eager attention. Attention sinks are identified from the final

layer using $\lambda = 1.0$, a maximum window size of 25, and a minimum secondary-sink length of three tokens. We generate 10 reasoning paths with a batch size of 10, zero branching budget, temperature 0.7, top-$p = 1.0$, and top-$k = 20$. Task-adapted prompts are injected without punctuation preference or visible output. Final generation uses greedy decoding, a maximum of 512 new tokens, a repetition penalty of 1.2, and a three-token no-repeat constraint.

Deep-learning baselines: All models use 224×224 inputs, a batch size of 16, a maximum of 50 epochs, a learning rate and weight decay of 10^{-4}, early stopping with a patience of five epochs, automatic mixed precision, and a random seed of 42. Pretrained initialization is applied where supported. **1) CNN:** The CNN comprises four 3×3 convolutional stages with 32, 64, 128, and 256 channels, each incorporating batch normalization, followed by a binary classification head. **2) ResNet:** ResNet-18 uses an initial 7×7 convolution and four residual stages with 64, 128, 256, and 512 channels, each containing two residual blocks, followed by a binary classification head. **3) ViT:** ViT-B/16 uses 16×16 patches, 12 Transformer encoder layers, a hidden dimension of 768, and an MLP dimension of 3,072. Each image produces 196 patch tokens and one classification token.

Fine-tuning: We fine-tune Qwen3-VL-8B-Instruct using LoRA with rank $r = 16$, $\alpha = 32$, and a dropout rate of 0.05. LoRA adapters are applied to the `q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, and `down_proj` modules, with no bias parameters trained. The model is optimized for 3 epochs using AdamW with a learning rate of 1×10^{-4} and no weight decay. We use a per-device batch size of 1 with 8 gradient accumulation steps, resulting in an effective batch size of 8 on a single GPU. The learning rate follows a cosine schedule with a warmup ratio of 0.03, and gradients are clipped to a maximum norm of 1.0. Training is performed with BF16 precision and TF32 enabled, while FP16 and gradient checkpointing are disabled. We evaluate and save checkpoints after each epoch, retain at most two checkpoints, and select the final checkpoint according to the lowest validation loss. All experiments use a random seed of 42.

PrivGate(Hu et al., 2026): We adapt PrivGate to Qwen3-VL-8B-Instruct using the raw hidden state of the final multimodal prompt token at layer 18. We train a standardized linear logistic-regression probe with ℓ_2 regularization, balanced class weights, and $C \in \{0.01, 0.1, 1, 10\}$ on an 80/20 group-wise train/validation split. The routing threshold is the highest validation threshold achieving at least 95% recall. At inference time, samples below the threshold use the ordinary descriptive prompt, while high-risk samples are routed to a fixed dataset-specific reasoning prompt. We retain PrivGate's binary pre-generation routing and do not use activation editing, attention sinks, or multi-level intervention. All experiments use greedy decoding with at most 265 new tokens and a repetition penalty of 1.0. The exact System-2 prompts are provided in Table 10.

B.4 Implementation Details of External Routers

To investigate whether an additional visual model can serve as the routing module, we construct three external-router baselines based on CNN, ResNet-18, and ViT-B/16. All routers are trained as image-only binary classifiers and use the same training pool and fixed test split as BRIDGE. The fixed evaluation samples are strictly excluded from router training, and the remaining samples are divided group-wise into 80% training and 20% validation sets to avoid scene-level leakage.

Router architectures. The CNN router consists of four 3×3 convolutional stages with 32, 64, 128, and 256 channels, respectively, each followed by batch normalization and ReLU activation, with spatial downsampling between stages and a final binary classification head. The ResNet router adopts ResNet-18, while the Transformer-based router uses ViT-B/16. ResNet-18 and ViT-B/16 are initialized with ImageNet-pretrained weights, whereas the CNN is trained from scratch.

Training. All routers take 224×224 images as input and are optimized using AdamW with balanced binary cross-entropy. We use a batch size of 16, automatic mixed precision, a maximum of 15 epochs, early stopping with a patience of four epochs, and a random seed of 42. During training, images are resized to 224×224, randomly horizontally flipped with probability 0.5, and normalized using the ImageNet mean and standard deviation. Validation and test images use the same resizing and normalization without random augmentation.

For hyperparameter selection, the CNN searches learning rates $\{10^{-3}, 3 \times 10^{-4}\}$ and weight decay $\{0, 10^{-4}\}$ with dropout 0.3. ResNet-18 searches learning rates $\{10^{-4}, 3 \times 10^{-5}\}$ and weight decay

Table 10: Dataset-specific prompts used for the PrivGate baseline.

Dataset	Prompt
IMD2020	*System-2:* You are the reasoning-enhanced visual reliability branch. Before answering, carefully check whether the depicted objects, actions, spatial relations, causal relations, and scene semantics are mutually consistent. If clear visual evidence supports manipulation, explicitly state that the image appears manipulated and briefly identify the contradiction. Otherwise, complete the description normally and avoid unsupported claims.
Columbia	*System-2:* You are the reasoning-enhanced image-authenticity branch. Before answering, carefully inspect object boundaries, lighting, shadows, texture, scale, geometry, and semantic consistency. If clear visual evidence supports splicing or manipulation, explicitly state that the image appears manipulated and briefly identify the evidence. Otherwise, complete the description normally and avoid unsupported claims.
WHOOPS	*System-2:* You are the reasoning-enhanced commonsense branch. Before answering, compare the visible objects, attributes, actions, and relations with ordinary real-world knowledge. If a clear commonsense violation is visible, explicitly call the scene weird and briefly explain the conflict. Otherwise, describe it normally without treating a merely unusual but possible event as a violation.
VLind	*System-2:* You are the reasoning-enhanced factual-consistency branch. Before answering, compare the depicted objects, attributes, actions, location, and time with established facts and natural laws. Ignore ordinary artifacts of AI-generated imagery. If a clearly supported contradiction exists, explicitly state that the scene is counterfactual and briefly explain why. Otherwise, describe it normally and avoid overclaiming.

$\{10^{-5}, 10^{-4}\}$ with dropout 0.2. ViT-B/16 searches learning rates $\{3 \times 10^{-5}, 10^{-5}\}$ and weight decay $\{10^{-5}, 10^{-4}\}$ with dropout 0.1. The best configuration is selected according to validation F1.

For each trained router, we additionally calibrate a classification threshold τ on the validation set. Specifically, we search $\tau \in [0.05, 0.95]$ with a step size of 0.01 and select the threshold that maximizes validation F1, using Recall and proximity to 0.5 as tie breakers. The external test set is never used for either model or threshold selection.

Routing and generation. Given an image x, the external classifier predicts an unreliability probability

$$p_{\text{ext}}(x) = \sigma\left(f_{\text{ext}}(x)\right), \tag{25}$$

where f_{ext} denotes the output logit of the corresponding visual router. The routing decision is defined as

$$\mathcal{R}_{\text{ext}}(x) = \begin{cases} P_G, & p_{\text{ext}}(x) < \tau, \\ P_V, & p_{\text{ext}}(x) \geq \tau, \end{cases} \tag{26}$$

where P_G is the descriptive prompt and P_V is a dataset-specific explicit verification prompt. Thus, the visual classifier is used only as a router and does not directly determine the final prediction.

All three router variants use the same downstream Qwen3-VL-8B-Instruct model, such that the comparison isolates the effect of the routing mechanism. A negative router prediction retains the descriptive generation path, whereas a positive prediction switches the VLM to the corresponding explicit verification prompt. Generation uses greedy decoding with a maximum of 265 new tokens, a repetition penalty of 1.2, and a three-token no-repeat constraint.

The additional overhead shown in Figure 5 denotes the *stored checkpoint size* of each routing module rather than its parameter count, reflecting the additional storage footprint required for deployment.

C Locating the G–D Gap

We further detail how the three complementary signals used in Section 3.2 are computed, following the order of *activation*, *attention*, and *output probability*.

Table 11: Prompt templates used for the external-router baselines.

Dataset	Descriptive Prompt	Verification Prompt
IMD2020	Give a brief description (under 265 tokens) of the image. Include all visible objects and their relations.	Carefully inspect the image for signs of digital manipulation. Check object boundaries, texture continuity, lighting, shadows, reflections, geometry, perspective, scale, and semantic consistency. Briefly describe the image, then explicitly conclude whether the image appears manipulated or not manipulated, and give the strongest visual evidence.
Columbia	Give a brief description (under 265 tokens) of the image. Include all visible objects and their relations.	Carefully inspect the image for image splicing or compositing. Focus on pasted-object boundaries, halos, local color or illumination mismatch, inconsistent sharpness, texture, scale, perspective, shadows, and semantic discontinuities. Briefly describe the image, then explicitly conclude whether it appears manipulated/spliced or not manipulated, and state the strongest evidence.
WHOOPS	Give a brief description (under 265 tokens) of the image. Include the objects, attributes, actions, and their relations.	Check the depicted scene against ordinary real-world commonsense expectations. Consider physical, biological, functional, social, spatial, temporal, object-use, identity, and semantic consistency. A scene can be weird even if it is technically possible; do not require strict physical impossibility and do not dismiss a clear commonsense conflict merely because the scene could be staged, artistic, symbolic, or unusual. Briefly describe the scene, then explicitly conclude whether it is weird or normal and explain the strongest conflict.
VLind	Briefly describe the objects, attributes, actions, and relations in the image.	Check the depicted content against established real-world facts and natural laws. Consider identity, object properties, biology, physics, geography, time, location, causality, and semantic relations. Ignore ordinary rendering artifacts that do not change the depicted fact. Briefly describe the image, then explicitly conclude whether the scene is counterfactual or factual and state the strongest factual contradiction if one exists.

Activation. To quantify the linear separability of reliable and manipulated inputs, we train an independent linear SVM for each layer ℓ and generation position t. For sample i, let $\mathbf{c}_{i,\ell,t}$ denote the cosine-center representation defined in Section 3.1. Only samples with valid representations at position t are retained.

For each (ℓ, t), the representations are standardized and reduced to at most 50 principal components. A class-balanced linear SVM is then evaluated using three-fold stratified cross-validation, with standardization and PCA fitted exclusively on each training fold to prevent information leakage. We report the mean positive-class F1 score across folds, treating manipulated images as the positive class. Higher F1 indicates stronger linear separability of manipulation-related information. The resulting layer–position heatmap therefore localizes where such evidence emerges during generation and does not measure the performance of the final BRIDGE detector.

Attention. We measure how much attention each generated token allocates to the visual input. Let $\mathcal{A}_{i,\ell,k,t,j}$ denote the post-softmax attention weight for sample i, from the query corresponding to generation position t to key position j, at layer ℓ and attention head k. We define the set of

visual-token positions $\mathcal{V}_i$ as all <|image_pad|> tokens in the original multimodal input. The visual-attention ratio is computed as

$$\rho^{\text{vis}}_{i,\ell,k,t} = \frac{\sum_{j\in\mathcal{V}_i} \mathcal{A}_{i,\ell,k,t,j}}{\sum_j \mathcal{A}_{i,\ell,k,t,j} + \epsilon}, \tag{27}$$

where $\epsilon = 10^{-12}$ is used for numerical stability. All non-image positions, including textual prompt tokens, special tokens, and previously generated tokens, are included in the denominator. For each layer and generation position, we average over all attention heads and all samples with valid attention tensors:

$$\bar{\rho}^{\text{vis}}_{\ell,t} = \frac{1}{N_t H} \sum_{i=1}^{N_t} \sum_{k=1}^{H} \rho^{\text{vis}}_{i,\ell,k,t}, \tag{28}$$

where H is the number of attention heads and N_t is the number of valid samples at position t. We compute the same quantity separately under the descriptive and explicit verification prompts. The resulting curves therefore measure aggregate visual attention during generation, rather than attention to a manually selected image region.

Probability probe. To examine whether the internally separable distinction is transferred to the final decoding distribution, we analyze the next-token probabilities produced by the final language-model head. At generation position t, the probability distribution for sample i is

$$\mathbf{p}_{i,t} = \text{softmax}\left(W_{\text{LM}}\mathbf{h}_{i,L,t} + b_{\text{LM}}\right). \tag{29}$$

For each class $g \in \{0, 1\}$, we first compute the class-mean distribution

$$\bar{\mathbf{p}}_{g,t} = \frac{1}{N_g} \sum_{i:y_i=g} \mathbf{p}_{i,t}. \tag{30}$$

We then select the top-K tokens from each of the two class-mean distributions and form their union $\mathcal{U}_t$. In our experiments, $K = 20$. The two distributions are restricted to $\mathcal{U}_t$ and renormalized:

$$\tilde{p}_{g,t}(j) = \frac{\bar{p}_{g,t}(j)}{\sum_{u\in\mathcal{U}_t} \bar{p}_{g,t}(u)}, \qquad j \in \mathcal{U}_t. \tag{31}$$

Their difference is quantified using Jensen–Shannon divergence,

$$D_{\text{JS}}\left(\tilde{\mathbf{p}}_{0,t} \,\|\, \tilde{\mathbf{p}}_{1,t}\right) = \frac{1}{2} D_{\text{KL}}\left(\tilde{\mathbf{p}}_{0,t} \,\|\, \mathbf{m}_t\right) + \frac{1}{2} D_{\text{KL}}\left(\tilde{\mathbf{p}}_{1,t} \,\|\, \mathbf{m}_t\right), \tag{32}$$

where $\mathbf{m}_t = (\tilde{\mathbf{p}}_{0,t} + \tilde{\mathbf{p}}_{1,t})/2$. Natural logarithms are used.

To distinguish class-specific separation from ordinary variation within a class, we additionally construct within-class baselines. For each class, its samples are randomly divided into two disjoint halves, and the same top-K union and Jensen–Shannon computation is applied between the two subsets. This procedure is repeated 50 times, and we report the mean and standard deviation across repetitions at each generation position. Consequently, the between-class divergence is interpreted relative to the intrinsic within-class variability: a between-class divergence that is not larger than the within-class baseline indicates that the manipulation distinction is only weakly expressed in the final next-token distribution.

D SPECIFIC INFERENCE EFFICIENCY AND GENERATION FLUENCY RESULTS

For N generated tokens, let p_t denote the probability assigned to the t-th token and T_{dec} the total decoding time in seconds. We compute:

$$\text{PPL} = \exp\left(-\frac{1}{N}\sum_{t=1}^{N} \log p_t\right), \qquad \text{Lat.} = \frac{1000 T_{\text{dec}}}{N}, \qquad \text{TPS} = \frac{N}{T_{\text{dec}}}.$$

Table 12: Per-dataset comparison of generation fluency and inference efficiency. Lat. denotes latency in ms/token, TPS denotes throughput in tokens/s, and PPL is dimensionless.

	Manipulation Detection						Semantic Unreliability Detection					
	IMD2020			Columbia			WHOOPS			VLind-Bench		
Method	PPL ↓	Lat. ↓	TPS ↑	PPL ↓	Lat. ↓	TPS ↑	PPL ↓	Lat. ↓	TPS ↑	PPL ↓	Lat. ↓	TPS ↑
Descriptive Prompt	2.76	139.57	7.16	2.66	158.62	6.30	2.78	152.36	6.56	3.21	98.96	10.11
BRIDGE	4.26	166.82	5.99	4.18	184.08	5.43	4.29	203.15	4.92	4.15	211.38	4.73

Lower PPL and latency indicate better fluency and efficiency, whereas higher TPS indicates faster generation. Averaged across four datasets, BRIDGE increases PPL from 2.85 to 4.22 and latency from 137.38 to 191.36 ms/token, while reducing TPS from 7.53 to 5.27. This expected degradation arises because injection interrupts the original decoding trajectory and introduces an additional reconsideration stage, trading some fluency and efficiency for more reliability-aware generation.