HALLUMIRROR: DISCOVERING AND MITIGATING MULTIMODAL HALLUCINATIONS VIA CO-EVOLUTIONARY SELF-PLAY

Anonymous authors
Paper under double-blind review

ABSTRACT

Mitigating hallucinations in Multimodal Large Language Models (MLLMs) increasingly relies on constructed training supervision. However, existing methods typically optimize against predefined hallucination patterns, curated preferences, or model-generated samples whose challenges are not explicitly adapted to the model's evolving failure modes. We identify this mismatch as a key limitation: as the model improves, effective supervision should continually shift toward its current unresolved hallucination weaknesses. To this end, we propose **HalluMirror**, a co-evolutionary adversarial self-play framework that continually transforms model failures into new training challenges. HalluMirror couples an *Inducer*, optimized to generate fine-grained existence, attribute, and relation hallucinations that expose the current Challenger's weaknesses, with a *Challenger* that learns to detect and repair these evolving hard cases. Through alternating optimization, weakness discovery, adversarial exposure, and targeted repair form a closed self-improvement loop. To reliably scale this process, we further introduce a training-free grounding-based hallucination verifier that filters self-play-generated hallucinations without hallucination-specific human annotations. Extensive experiments across multiple MLLMs and hallucination benchmarks show that HalluMirror consistently improves robustness against diverse hallucination types. On GLM-4.1V-9B, HalluMirror improves performance across all eight hallucination benchmarks, with co-evolution identifying and addressing the model's vulnerability to relation hallucinations and yielding a 14.5% relative gain on FINER's multi-relation subset. All data and code will be made publicly available.

1 INTRODUCTION

In recent years, Multimodal Large Language Models (MLLMs) have achieved remarkable progress in visual understanding, visual question answering, and multimodal reasoning, demonstrating strong capabilities in cross-modal perception and generation. Despite this progress, MLLMs remain prone to multimodal hallucinations, where generated content is inconsistent with the visual evidence in the input image, including nonexistent objects, incorrect attributes, or invalid object relations (Bai et al., 2025b). Such errors substantially undermine the factual reliability of model outputs and hinder their deployment in high-stakes applications (Huang et al., 2024b). Therefore, mitigating multimodal hallucinations and strengthening the grounding of model generations in visual evidence are essential for building trustworthy multimodal AI systems (Zhao et al., 2024).

Existing studies have explored diverse paradigms for mitigating multimodal hallucinations, ranging from inference-time intervention to training-based alignment. Inference-time methods reduce hallucinations by recalibrating decoding distributions through contrastive branches (Leng et al., 2024; Huo et al., 2025; Im et al., 2026) or by post-hoc correcting generated responses (Zhou et al., 2024; Yin et al., 2024). While requiring no model retraining, these approaches often introduce additional inference overhead and provide limited persistent improvement in the model's intrinsic capabilities. Training-based alignment methods instead directly improve hallucination robustness through hallucination-aware data construction, human or AI feedback, and preference optimization (Yu et al., 2024b; Sarkar et al., 2025; Yang et al., 2025; Xiao et al., 2026). However, their supervision is largely

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.

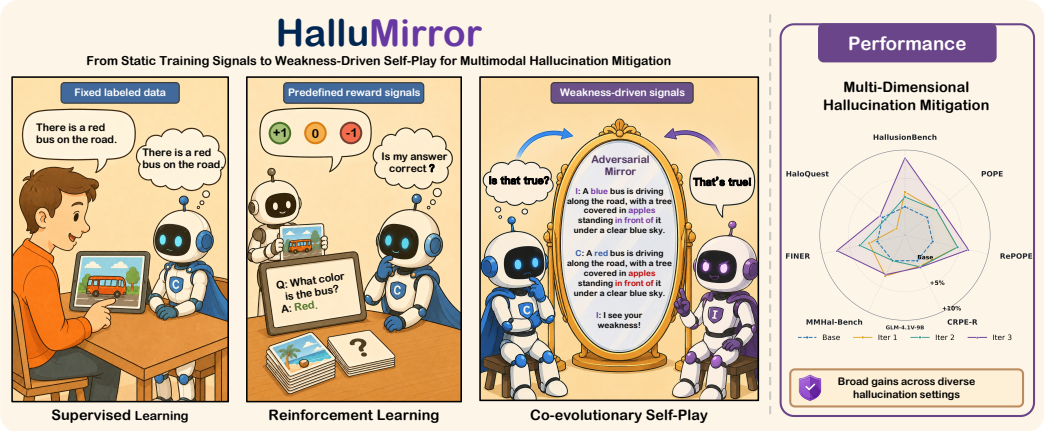

Figure 1: **HalluMirror** paradigm. (Left): **HalluMirror** replaces fixed supervision and predefined rewards with weakness-driven signals generated through co-evolutionary self-play. (Right): iterative self-play yields broad gains across diverse hallucination benchmarks.

shaped by predefined data-construction rules, external feedback, or curated preference signals, leaving the failure patterns emphasized during training largely determined by prescribed supervision.

To reduce reliance on prescribed supervision, recent automated and iterative alignment methods refresh supervision through model-generated feedback or self-generated training data (Yu et al., 2025; Lu et al., 2025). However, their training challenges remain primarily governed by model sampling or predefined strategies and curricula, rather than explicitly optimized to expose the model's evolving hallucination weaknesses. To address this gap, self-play has emerged as a promising paradigm for self-evolving model optimization. By coupling interacting or co-evolving roles, Self-Play can generate training challenges conditioned on the model's current capabilities, allowing supervision and task difficulty to evolve accordingly. This paradigm has progressed from language-model self-improvement (Zhao et al., 2025; Liu et al., 2026) to multimodal self-evolution (Wang et al., 2026a; He et al., 2025; Qiu et al., 2026), while Yang et al. (2026) further applies an evolving Generator–Detector game to textual faithfulness hallucination detection in RAG. These advances suggest a natural route to multimodal hallucination mitigation: continually turning the model's current failures into targeted training signals.

This leaves two key challenges: dynamically discovering and repairing model-specific hallucination weaknesses through self-play, and reliably scaling the resulting training signals without hallucination-specific human annotations. We therefore propose **HalluMirror**, a co-evolutionary framework for multimodal hallucination mitigation that integrates adversarial self-play with grounding-based hallucination verification. Within its self-play mechanism, HalluMirror couples two roles instantiated from the same pretrained MLLM, an *Inducer* and a *Challenger*, which adversarially "mirror" the model's own weaknesses: the Inducer generates fine-grained hallucinations that expose the Challenger's current weaknesses, while the Challenger learns to detect and repair these evolving hard cases. Through alternating optimization, exposed failures are continually converted into new training challenges, forming a closed loop of weakness discovery and hallucination mitigation. To support the scaling of this loop while maintaining the reliability of its training signals, HalluMirror incorporates a visual-grounding-based fine-grained hallucination verifier that automatically validates and filters existence, attribute, and relation hallucinations. Our main contributions are summarized as follows:

- We propose **HalluMirror**, a co-evolutionary adversarial self-play framework for multimodal hallucination mitigation. Through self-play, hallucination generation and correction co-evolve with model capabilities, continually exposing and repairing model-specific hallucination weaknesses.
- We introduce a **visual-grounding-based fine-grained hallucination verifier** within HalluMirror to support scalable self-play. The verifier validates and filters self-play-generated

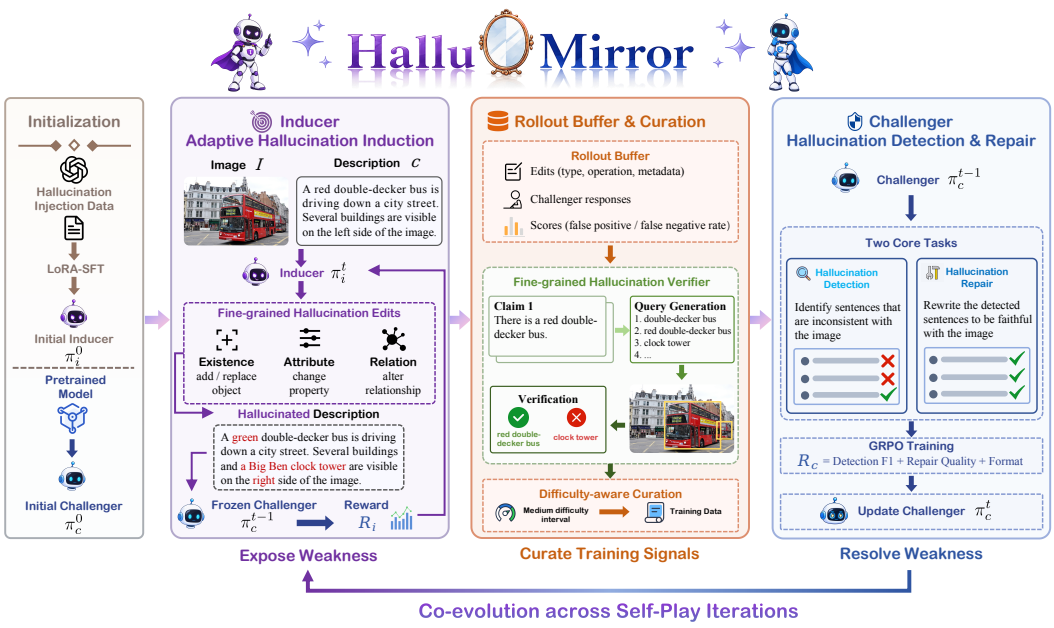

Figure 2: Overview of HalluMirror. The Inducer adaptively exposes the Challenger's hallucination weaknesses, while verified and curated interaction rollouts are used to train the Challenger for detection and repair. Alternating optimization enables both roles to co-evolve across self-play iterations.

existence, attribute, and relation hallucinations, providing reliable training signals without hallucination-specific human annotations.

- Extensive experiments across multiple MLLMs and hallucination benchmarks demonstrate the effectiveness and generalizability of HalluMirror. Evolution analysis further reveals that adversarial attacks adapt to model-specific vulnerabilities, accompanied by targeted improvements in the corresponding hallucination weaknesses.

2 METHOD

Problem Formulation. Let $\mathcal{D}$ denote a dataset of image–description pairs and let $(I, c) \in \mathcal{D}$ be a sampled instance, where I is an input image and $c = (s_1, \ldots, s_n)$ is a visually faithful description consisting of n sentences. HalluMirror instantiates two policies from the same pretrained MLLM: an *Inducer* π_i that intentionally introduces fine-grained hallucinations into c, and a *Challenger* π_c that detects and repairs the induced hallucinations. Given (I, c), the Inducer generates a sequence of K localized edits $e_{1:K} \sim \pi_i(\cdot \mid I, c)$, where each edit e_j is associated with one of three hallucination types $h_j \in \mathcal{H}$, with $\mathcal{H} = \{\text{existence}, \text{attribute}, \text{relation}\}$. Applying these edits yields a hallucinated description $\tilde{c}$. The Challenger then takes $(I, \tilde{c})$ as input and outputs sentence-level hallucination predictions $\hat{\mathbf{y}} \in \{0, 1\}^n$ together with a repaired description $\hat{c}$.

Building on this formulation, HalluMirror alternates adversarial optimization of the Inducer and Challenger, continually converting the Challenger's current failures into subsequent training challenges. To support reliable and scalable self-play, HalluMirror incorporates a visual-grounding-based fine-grained hallucination verifier that filters self-play-generated hallucinations. An overview of HalluMirror is shown in Fig. 2.

2.1 ADVERSARIAL HALLUCINATION SELF-PLAY

A fixed hallucination generator cannot adapt to the evolving weaknesses of the Challenger, potentially producing signals that no longer target its unresolved failure modes. HalluMirror addresses this limitation through alternating adversarial self-play. At iteration t, the current Challenger π_c^{t-1} is held fixed as the opponent while the Inducer is optimized from π_i^{t-1} to π_i^t to generate hallucinations

that expose its current weaknesses. This optimization produces a collection of adversarial rollouts $\mathcal{R}_t$, which are evaluated using the Challenger's responses and used to construct an iteration-specific training set $\mathcal{D}_t^{\text{adv}}$ through difficulty-aware selection. The resulting $\mathcal{D}_t^{\text{adv}}$ is then used to update the Challenger from π_c^{t-1} to π_c^t. One iteration of this alternating process is summarized as:

$$(\pi_i^t, \mathcal{R}_t) \leftarrow \text{Opt}_i\big(\pi_i^{t-1}, \pi_c^{t-1}\big), \qquad \mathcal{D}_t^{\text{adv}} \leftarrow \text{Gen}(\mathcal{R}_t), \qquad \pi_c^t \leftarrow \text{Opt}_c\big(\pi_c^{t-1}, \mathcal{D}_t^{\text{adv}}\big). \tag{1}$$

Here, Opt_i and Opt_c denote the policy optimization operators for the Inducer and Challenger, respectively, while Gen denotes difficulty-aware training data generation from the collected rollouts.

2.1.1 Adversarial Hallucination Induction

Supervised Initialization. A reliable adversarial interaction requires the Inducer to first acquire rule-consistent hallucination injection. However, we empirically observe that directly using the pretrained MLLM for this task often leads to invalid edits, excessive rewriting, or malformed structured outputs, providing an unstable starting point for self-play. We therefore synthesize 2K examples with GPT-5.4 using the same hallucination-injection instruction as in self-play and perform LoRA-based SFT to obtain the initial Inducer π_i^0. This stage only teaches the required editing behavior and output format. In contrast, the Challenger is initialized directly from the pretrained MLLM, since supervised warm-starting on constructed hallucinations could bias its initial failure profile toward predefined patterns rather than exposing its intrinsic weaknesses.

Fine-grained Hallucination Injection. Fine-grained visual inconsistencies remain particularly challenging for current MLLMs: Xiao et al. (2026) shows that performance deteriorates as negative queries contain increasingly subtle mismatches among otherwise valid visual elements. Motivated by this observation, given a faithful description c, the Inducer performs localized edits $e_{1:K}$ over $\mathcal{H} = \{\text{existence}, \text{attribute}, \text{relation}\}$. Existence hallucinations add a nonexistent entity or replace an existing one, while attribute and relation hallucinations locally replace the corresponding visual property or relation. Each edit introduces a single inconsistency, modifies only the necessary textual span, and preserves non-targeted visual facts while remaining semantically plausible and linguistically natural. For existence hallucinations, we further favor plausible objects that commonly co-occur with the observed scene, following the object co-occurrence findings of Zhou et al. (2024). For attribute and relation hallucinations, we emphasize fine-grained visual mismatches by replacing grounded attributes or relations with plausible but incorrect alternatives, yielding descriptions that remain linguistically natural but contradict the image.

These localized edits also keep hallucinated descriptions close to their faithful counterparts, preventing trivial discrimination through large textual deviations. This design is consistent with Qiu et al. (2026), which shows that near-neighbor negative claims encourage stronger reliance on visual evidence as their semantic distance decreases. Inspired by Lu et al. (2025)'s curriculum of progressively reducing injected hallucinations, we define an iteration-dependent edit-count distribution p_t, where $p_t(K)$ denotes the proportion of samples assigned K edits in self-play iteration t. In later iterations, more probability mass is assigned to smaller K, yielding fewer injected hallucinations. This curriculum controls only the number of edits, while their types and specific content remain determined by the Inducer.

Rollout Buffer. During the adversarial interaction at iteration t, we maintain a rollout buffer $\mathcal{R}_t$ to retain the hallucination trajectories produced by the Inducer together with their corresponding feedback from the current Challenger π_c^{t-1}. These rollouts directly reflect the current Challenger's ability to handle the induced hallucinations and are subsequently reused to construct $\mathcal{D}_t^{\text{adv}}$. Reusing these interaction rollouts avoids an additional data-generation stage and reduces the cost of constructing Challenger training data.

2.1.2 Hallucination Detection and Repair

Difficulty-aware Data Curation. He et al. (2025) shows that filtering training samples by model-perceived difficulty can retain more informative supervision, excluding both trivial examples that the model already solves confidently and highly unstable or noisy ones. Following this principle, we construct Challenger training data according to the difficulty of the rollouts in $\mathcal{R}_t$. Specifically, we estimate the difficulty of each rollout $r \in \mathcal{R}_t$ using the current Challenger's detection errors:

$$d(r) = \alpha_{\text{fn}}\,\text{FNR}(r) + \alpha_{\text{fp}}\,\text{FPR}(r), \tag{2}$$

where $\mathrm{FNR}(r)$ measures the fraction of injected hallucinations missed by the Challenger, $\mathrm{FPR}(r)$ measures faithful content incorrectly identified as hallucinated, and α_{fn} and α_{fp} weight their contributions to the difficulty score. Among multiple rollouts from the same source instance (I, c), we retain the highest-difficulty rollout and rank the resulting candidates by $d(r)$. We then select the middle-difficulty portion to construct $\mathcal{D}_t^{\mathrm{adv}}$, focusing training on challenging yet informative cases.

Hallucination Detection and Repair. Given an adversarial example $(I, \tilde{c}) \in \mathcal{D}_t^{\mathrm{adv}}$, where $\tilde{c} = (\tilde{s}_1, \ldots, \tilde{s}_n)$, the Challenger π_c jointly detects and repairs the induced hallucinations. It predicts a sentence-level hallucination label $\hat{y}_j \in \{0, 1\}$ for each sentence $\tilde{s}_j$ and corrects detected hallucinations while leaving faithful sentences unchanged. During injection, we retain the faithful pre-injection description c together with the edit information. The edits provide sentence-level supervision for hallucination detection and localization, while c serves as the correction reference. An LLM judge further evaluates the semantic consistency between the repaired content and its faithful reference, encouraging valid corrections while avoiding unnecessary changes to faithful content.

2.1.3 Alternating Self-Play Optimization

HalluMirror alternately optimizes the Inducer and Challenger using GRPO (Shao et al., 2024) implemented with `veRL` (Sheng et al., 2025). At iteration t, we freeze the current Challenger π_c^{t-1} and optimize the Inducer to expose its unresolved hallucination weaknesses. The resulting rollouts are then curated into $\mathcal{D}_t^{\mathrm{adv}}$, which is used to optimize the Challenger and obtain π_c^t for the next iteration.

Role-specific Rewards. The two roles are optimized with complementary rewards. For the Inducer, we primarily reward hallucinations that escape detection by the current Challenger:

$$R_i = \lambda_{\mathrm{fn}} R_{\mathrm{fn}} + \lambda_{\mathrm{fp}} R_{\mathrm{fp}} + R_{\mathrm{aux}}, \quad \lambda_{\mathrm{fn}} > \lambda_{\mathrm{fp}}, \tag{3}$$

where R_{fn} rewards induced hallucinations missed by the Challenger, while R_{fp} reflects faithful content incorrectly identified as hallucinated. The larger weight on R_{fn} encourages the Inducer to preferentially expose hallucination patterns that remain unresolved by the current Challenger, while R_{aux} enforces structural validity and valid hallucination injection.

For the Challenger, the reward jointly evaluates hallucination detection and repair:

$$R_c = \lambda_{\mathrm{det}} R_{\mathrm{det}} + \lambda_{\mathrm{rep}} R_{\mathrm{rep}} + R_{\mathrm{fmt}}, \tag{4}$$

where R_{det} is the sentence-level hallucination classification F1 score between the predicted labels $\hat{\mathbf{y}}$ and injection-derived labels $\mathbf{y}$. R_{rep} evaluates each repaired hallucinated sentence against its faithful pre-injection reference using an LLM judge, which assigns a score of 1 to semantically consistent repairs. R_{fmt} enforces the required output structure.

Group-Relative Policy Optimization. Following standard GRPO, each input produces G sampled rollouts with role-specific rewards $\{R_k^{(g)}\}_{g=1}^{G}$. Their group-relative advantages are computed as:

$$\hat{A}_k^{(g)} = \frac{R_k^{(g)} - \mathrm{mean}\Big(\{R_k^{(j)}\}_{j=1}^{G}\Big)}{\mathrm{std}\Big(\{R_k^{(j)}\}_{j=1}^{G}\Big) + \epsilon}, \qquad k \in \{i, c\},\ g = 1, \ldots, G, \tag{5}$$

where $\epsilon > 0$ is a small constant for numerical stability. These relative advantages favor Inducer rollouts that better expose unresolved hallucination patterns and Challenger rollouts that better detect and repair the induced hallucinations.

2.2 Grounding-based Hallucination Verification

To ensure the reliability of large-scale self-play-generated hallucination data, we introduce a fine-grained verifier that determines whether injected hallucinations are inconsistent with visual evidence. Recent studies (Jin et al., 2026; Min et al., 2026) obtain fine-grained hallucination judgments by training specialized diagnosis or reward models with dedicated supervision, while Chen et al. (2024a) demonstrates that external visual evidence can support claim-level hallucination verification. Inspired by this, we develop a training-free grounding-based verifier using a unified visual grounding model to provide evidence for existence, attribute, and relation hallucinations.

HalluMirror must verify not only the existence of open-vocabulary entities, but also compositional visual semantics involving attributes and relations. We therefore employ Wang et al. (2026b) as

the visual evidence extractor, which supports open-vocabulary localization from natural-language queries and provides a unified interface for fine-grained visual grounding. Specifically, given a description x, we decompose it into visual claims $\mathcal{C}(x) = \{u_m\}_{m=1}^{M}$ and construct a grounding query q_m for each claim. The verification process is formulated as:

$$q_m = Q(u_m), \qquad E_m = \mathrm{Agg}(\mathrm{Locate}(I, q_m)), \qquad \hat{h}_m = V(u_m, E_m), \tag{6}$$

where $Q(\cdot)$ constructs the grounding query for visual claim u_m, $\mathrm{Locate}(\cdot)$ denotes the grounding operation performed by LocateAnything, $\mathrm{Agg}(\cdot)$ aggregates the grounded outputs into visual evidence E_m, and $V(\cdot)$ denotes the LLM verifier that predicts $\hat{h}_m \in \mathcal{H} \cup \{\mathrm{none}\}$. Detailed claim decomposition, query construction, and verification prompts are provided in Appendix B.3.

3 Experiments

3.1 Experimental Setup

Benchmarks. We evaluate HalluMirror on eight hallucination benchmarks covering different levels of visual granularity and response settings, with all evaluations following the official protocols:

- **Fundamental Visual Consistency.** Li et al. (2023) evaluates coarse-grained object hallucination by testing whether MLLMs can correctly distinguish existing from non-existing objects. Neuhaus & Hein (2025) provides a more reliable evaluation of object hallucination by correcting annotation errors inherited from MSCOCO. Wang et al. (2024a) evaluates object recognition and fine-grained visual relation comprehension, testing whether models can correctly identify entities and their relationships in an image.
- **Fine-grained Visual Hallucination.** Xiao et al. (2026) evaluates whether MLLMs can detect subtle visual inconsistencies embedded in fine-grained object, attribute, and relation descriptions. Yin et al. (2026) evaluates fine-grained visual perception under localized counter-commonsense content, testing whether models rely on actual visual evidence rather than commonsense priors.
- **General Hallucination Evaluation.** Guan et al. (2024) evaluates image-context reasoning and robustness against language hallucinations and visual illusions. Sun et al. (2024) evaluates hallucinations in open-ended responses across diverse visual perception and reasoning scenarios. Wang et al. (2024b) evaluates hallucination-prone visual question answering involving false premises, insufficient visual evidence, and challenging visual content.

Baselines. We compare HalluMirror with representative hallucination mitigation methods covering training-free decoding and training-based alignment. Among training-free methods, VCD (Leng et al., 2024) contrasts predictions conditioned on original and distorted visual inputs, SID (Huo et al., 2025) constructs a hallucination-amplified branch from less informative visual tokens for contrastive decoding, and Self-Aug (Im et al., 2026) uses query-conditioned visual augmentations with adaptive candidate truncation. Among training-based methods, HALVA (Sarkar et al., 2025) aligns correct and hallucinated phrases with KL regularization, HSA-DPO (Xiao et al., 2025) incorporates fine-grained hallucination severity into preference optimization, and OPA-DPO (Yang et al., 2025) performs preference optimization using expert-revised on-policy responses. We also include RLAIF-V (Yu et al., 2025), which iteratively learns from AI-generated preference feedback through deconfounded divide-and-conquer evaluation, and FINER-Tuning (Xiao et al., 2026), which applies DPO to preferences constructed from fine-grained positive and negative visual queries. Detailed baseline adaptation and implementation settings are provided in Appendix C.2.

Training Settings. Training images are randomly sampled from Visual Genome (Krishna et al., 2017), with descriptions generated by GPT-5.4 and retained only after passing our hallucination verification. We remove any images that overlap with the evaluation benchmarks, ensuring image-level disjointness between training and evaluation data. We consider three recent and competitive open-source MLLMs with diverse multimodal architectures: GLM-4.1V-9B (Team et al., 2025), Qwen3-VL-8B (Bai et al., 2025a), and MiniCPM-o-2.6 (Yao et al., 2025). GLM-4.1V-9B serves as our primary backbone for baseline reproduction, while the other two models are used to evaluate cross-model generalization. For supervised initialization, we train the Inducer on 2K synthesized hallucination-injection examples using LoRA ($r = 16$, $\alpha = 32$) with a learning rate of

Table 1: Results on hallucination benchmarks for GLM-4.1V-9B. Small numbers denote changes over the base model. HR: Hallucination Rate. Best/second-best results are shown in **bold**/underline.

Method	Data	POPE		RePOPE		HallusionBench	CRPE	FINER	FREAK	MMHal-Bench		HaloQuest	Avg.↑
		Acc.↑	F1↑	Rec.↑	F1↑	Pair Acc.↑	Acc.↑	Pair Acc.↑	Acc.↑	Score↑	HR↓	Score↑	
GLM-4.1V-9B Base	–	86.91	85.33	87.74	91.46	25.49	88.15	51.40	32.40	3.33	33.33	58.61	62.44
Training-Free Decoding													
VCD *(CVPR'24)*	–	86.87	85.49	89.12	91.39	24.62	84.75	52.25	33.18	3.36	37.50	57.45	61.45
SID *(ICLR'25)*	–	87.22	85.79	88.27	91.69	26.15	87.83	51.16	**34.89**	3.31	36.46	58.94	62.50
Self-Aug *(ICLR'26)*	–	87.32	86.02	88.84	91.56	26.81	86.97	51.78	32.74	3.48	**30.21**	60.10	63.22
Training-Based Alignment													
FINER-Tuning *(CVPR'26)*	160K	86.73	85.08	86.89	91.40	27.91	87.73	**59.98**	34.11	3.47	**30.21**	61.05	**64.63**
HALVA *(ICLR'25)*	21.9K	87.02	85.48	87.76	91.49	25.05	88.10	51.35	31.37	3.33	35.42	58.44	61.98
HSA-DPO *(AAAI'25)*	8.4K	86.49	86.88	88.61	91.59	**28.07**	86.97	52.81	33.19	**3.64**	**30.21**	60.93	63.78
OPA-DPO *(CVPR'25)*	4.7K	87.31	85.91	89.18	92.21	13.85	85.58	46.98	22.54	2.92	44.79	54.80	57.14
RLAIF-V *(CVPR'25)*	83.1K	87.00	85.48	87.43	91.09	27.47	88.04	56.76	34.13	3.38	33.33	**61.40**	63.88
HalluMirror (Ours)													
Iter. 1	1.05K/it.	88.70 +1.79	87.84 +2.51	91.64 +3.90	**92.45** +0.99	26.15 +0.66	88.07 -0.08	52.15 +0.75	32.18 -0.22	3.43 +0.10	33.33 0.00	56.79 -1.82	62.79 +0.35
Iter. 2	1.05K/it.	88.73 +1.82	87.91 +2.58	91.69 +3.95	92.44 +0.98	25.93 +0.44	88.21 +0.06	53.02 +1.62	31.65 -0.75	3.33 0.00	33.33 0.00	58.28 -0.33	63.01 +0.57
Iter. 3	1.05K/it.	**89.38** +2.47	**88.84** +3.51	**93.42** +5.68	92.32 +0.86	27.69 +2.20	**88.26** +0.11	55.15 +3.75	32.93 +0.53	3.42 +0.09	32.29 +1.04	58.94 +0.33	63.98 +1.54

1×10^{-4}. For self-play optimization, we conduct three alternating iterations, each using 1.5K image–description pairs. Following the difficulty score $d(r)$ defined in Sec. 2.1.2, we retain rollouts within the difficulty-rank interval $[15\%, 85\%]$ to construct $\mathcal{D}_t^{\text{adv}}$. Both the Inducer and Challenger are optimized with GRPO using the `veRL` framework, with $G = 6$ rollouts per input and a learning rate of 3×10^{-6}. We apply LoRA ($r = 64$, $\alpha = 64$) to all non-visual linear layers during self-play. All training is conducted on two NVIDIA A800 80GB GPUs, with each self-play iteration taking approximately 24 hours for the 9B model.

3.2 Results on Hallucination Benchmarks

As shown in Table 1, HalluMirror yields broad improvements over the GLM-4.1V-9B base model across diverse hallucination settings. At Iter. 3, all reported metrics improve in their preferred directions, with notable gains of $+2.47$/$+3.51$ points on POPE Acc./F1, $+5.68$ on RePOPE Recall, $+2.20$ on HallusionBench, and $+3.75$ on FINER. HalluMirror achieves the best results on POPE Acc./F1, RePOPE Recall, and CRPE, while Iter. 1 obtains the best RePOPE F1. Its average score increases from 62.44 to 63.98, outperforming all training-free decoding baselines and remaining competitive with training-based alignment methods using 1.05K curated self-play examples at each iteration. On the open-ended MMHal-Bench and HaloQuest, HalluMirror also improves over the base model, with relatively modest gains compared with those on discriminative benchmarks. More importantly, by Iter. 3, HalluMirror achieves improvements over the base model across all reported metrics, with each self-play iteration using only 1.05K curated examples. It ranks second overall behind FINER-Tuning (64.63), which exhibits particularly strong performance on FINER. These results highlight the value of dynamically adapting training signals to the model's current weaknesses, enabling broad hallucination mitigation with a compact set of targeted training examples.

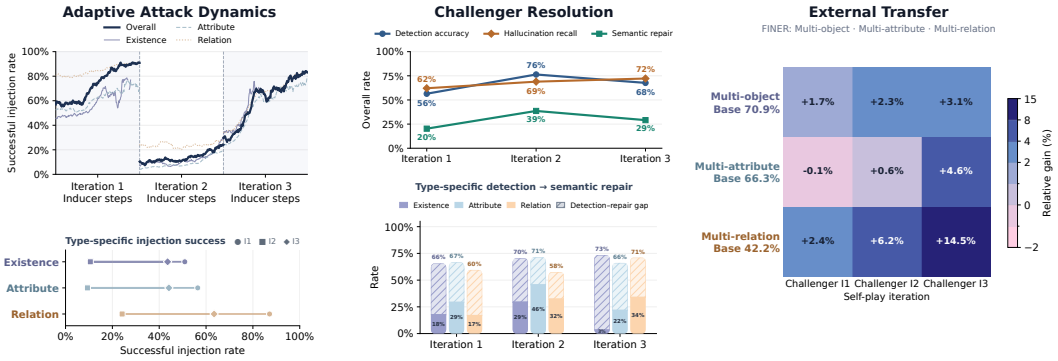

Figure 3: Co-evolution of the Inducer and Challenger across self-play iterations on GLM-4.1V-9B, together with external transfer to hallucination benchmark.

3.3 Self-Play Evolution Analysis

Self-play adaptively exposes and repairs model-specific hallucination weaknesses. Figure 3 reveals a clear alternating co-evolution pattern in HalluMirror. Within each Inducer optimization stage, the successful injection rate rises as the Inducer adapts to the current Challenger. After the Challenger is updated, the injection rate drops markedly at Iter. 2, indicating that previously effective attacks have become easier to resolve. Subsequent Inducer optimization then recovers the attack success, with a sharp increase at Iter. 3 as new unresolved weaknesses are discovered. Type-wise results further identify relation hallucinations as the dominant weakness of GLM-4.1V-9B, consistently yielding the highest injection success. Across iterations, the Challenger's hallucination recall increases from 62% to 72%, while detection accuracy and semantic repair peak at Iter. 2 before declining as the Inducer exposes harder weaknesses at Iter. 3. Relation repair improves steadily, whereas existence and attribute repair follow the overall trend, with existence repair dropping to only 3% at Iter. 3, suggesting that the Inducer generates increasingly subtle hallucinations that are difficult to correct. Detection consistently exceeds exact repair, potentially due to the larger solution space of fine-grained hallucination repair. Importantly, these dynamics transfer to the independent FINER benchmark, where relative improvements increase across all three settings and are largest for multi-relation, rising from 2.4% at Iter. 1 to 14.5% at Iter. 3. Together, these results demonstrate the intended co-evolutionary loop, where each Challenger update suppresses existing attacks, while the Inducer continually discovers new vulnerabilities that drive subsequent targeted improvement.

3.4 Cross-Model Generalization

Table 2: Cross-model generalization results on MiniCPM-o-2.6 and Qwen3-VL-8B. Small numbers denote changes over the corresponding base model. HR denotes Hallucination Rate. Best/second-best results within each backbone are shown in **bold**/underline.

Method	POPE		RePOPE		HallusionBench	CRPE	FINER	FREAK	MMHal-Bench		HaloQuest
	Acc.↑	F1↑	Rec.↑	F1↑	Pair Acc.↑	Acc.↑	Pair Acc.↑	Acc.↑	Score↑	HR↓	Score↑
MiniCPM-o-2.6											
Base	87.91	87.50	93.39	91.70	23.74	85.95	49.33	32.28	3.15	44.79	59.93
Iter. 1	87.92 +0.01	87.52 +0.02	**93.90** +0.51	91.22 -0.48	25.49 +1.75	85.96 +0.01	49.51 +0.18	32.70 +0.42	3.18 +0.03	44.79 0.00	60.93 +1.00
Iter. 2	87.88 -0.03	87.44 -0.06	93.27 -0.12	91.75 +0.05	25.05 +1.31	85.95 0.00	49.82 +0.49	**34.05** +1.77	3.43 +0.28	38.54 +6.25	**61.26** +1.33
Iter. 3	**88.18** +0.27	**87.92** +0.42	93.36 -0.03	**91.95** +0.25	**26.59** +2.85	**85.98** +0.03	**50.06** +0.73	**34.05** +1.77	**3.55** +0.40	**34.38** +10.41	60.43 +0.50
Qwen3-VL-8B											
Base	88.78	88.33	94.24	93.29	32.31	88.32	64.73	39.88	3.34	**44.79**	82.95
Iter. 1	89.36 +0.58	**89.26** +0.93	**95.90** +1.66	92.64 -0.65	34.29 +1.98	88.54 +0.22	64.12 -0.61	39.99 +0.11	3.36 +0.02	45.83 -1.04	83.28 +0.33
Iter. 2	89.71 +0.93	89.12 +0.79	93.08 -1.16	**93.41** +0.12	33.85 +1.54	88.42 +0.10	64.21 -0.52	40.12 +0.24	3.36 +0.02	46.88 -2.09	83.28 +0.33
Iter. 3	**89.73** +0.95	89.15 +0.82	93.16 -1.08	**93.41** +0.12	**34.95** +2.64	**88.74** +0.42	**65.20** +0.47	**41.42** +1.54	**3.43** +0.09	45.83 -1.04	**84.27** +1.32

As shown in Table 2, HalluMirror exhibits distinct improvement profiles across MLLM backbones. For MiniCPM-o-2.6, the largest gains appear on HallusionBench and FREAK, together with a substantial 10.41-point reduction in MMHal-Bench hallucination rate. For Qwen3-VL-8B, pronounced improvements are observed on HallusionBench, FREAK, and HaloQuest. By Iter. 3, MiniCPM and Qwen outperform their respective base models on 10 and 9 of the 11 metrics, respectively, while the remaining metrics show limited degradation. These model-dependent improvement patterns are consistent with HalluMirror's adaptive self-play mechanism, where training challenges evolve with the current Challenger rather than following a fixed optimization target. Broad improvements across two additional MLLM families further support the cross-model generalizability of HalluMirror.

3.5 General Capability Evaluation

Table 3: General capability results across different MLLM backbones. Small numbers denote changes over the corresponding base model. Best/second-best results within each backbone are shown in **bold**/underline.

Benchmark	GLM-4.1V-9B				MiniCPM-o-2.6				Qwen3-VL-8B			
	Base	Iter. 1	Iter. 2	Iter. 3	Base	Iter. 1	Iter. 2	Iter. 3	Base	Iter. 1	Iter. 2	Iter. 3
MME↑	85.56	85.13 -0.43	**85.64** +0.08	85.20 -0.36	86.14	86.02 -0.12	**86.18** +0.04	86.14 0.00	89.55	**90.19** +0.64	89.34 -0.21	89.39 -0.16
MMBench↑	**84.04**	83.99 -0.05	83.90 -0.14	83.88 -0.16	82.17	82.24 +0.07	82.17 0.00	**82.33** +0.16	84.87	85.35 +0.48	**85.42** +0.55	85.40 +0.53
MathVista↑	68.37	**69.07** +0.70	67.47 -0.90	68.87 +0.50	67.37	67.77 +0.40	67.27 -0.10	**68.37** +1.00	70.07	**72.27** +2.20	71.37 +1.30	70.97 +0.90

We evaluate MME for general perception and cognition, MMBench for visual understanding and reasoning, and MathVista for visual mathematical reasoning. As shown in Table 3, GLM-4.1V-9B shows slight decreases on MME and MMBench, whereas MiniCPM-o-2.6 and Qwen3-VL-8B exhibit modest overall gains. All three models improve on MathVista at Iter. 3. Considering the substantial faithfulness improvements observed above, these limited changes in general capability suggest a favorable trade-off between hallucination mitigation and capability preservation.

3.6 Ablation on Self-Play Training

Adaptive self-play is crucial for effective hallucination mitigation. As shown in Table 4, SFT trains the Challenger on a fixed hallucination set generated by the initial Inducer π_i^0, whose injection strategy remains unchanged regardless of the Challenger's evolving performance. Fixed-Data GRPO uses the same static data but replaces SFT with GRPO, enabling a controlled comparison of the optimization paradigm. Static SFT yields mixed improvements, whereas Fixed-Data GRPO performs better on most metrics. More importantly, HalluMirror consistently outperforms Fixed-Data GRPO across all reported metrics, showing that dynamically adapting hallucination generation to the Challenger's unresolved weaknesses provides substantial additional gains. These results directly validate the effectiveness of HalluMirror's adaptive self-play mechanism.

Table 4: Ablation results of the self-play training strategy on GLM-4.1V-9B. Fixed/Adaptive denote training with fixed data and dynamically evolved self-play data, respectively. HR denotes Hallucination Rate. Best/second-best results are shown in **bold**/underline.

Method	Training Data		POPE		RePOPE		HallusionBench	CRPE	FINER	FREAK	MMHal-Bench		HaloQuest
	Fixed	Adaptive	Acc.↑	F1↑	Rec.↑	F1↑	Pair Acc.↑	Acc.↑	Pair Acc.↑	Acc.↑	Score↑	HR↓	Score↑
Base	–	–	86.91	85.33	87.74	91.46	25.49	88.15	51.40	32.40	3.33	33.33	58.61
SFT	✓	–	86.26	84.43	85.96	91.01	27.25	87.97	52.81	30.67	3.33	33.33	56.29
Fixed-Data GRPO	✓	–	87.71	86.49	89.32	91.90	26.37	87.26	54.73	32.45	3.36	33.33	57.45
HalluMirror (Iter. 3)	–	✓	**89.38**	**88.84**	**93.42**	**92.32**	**27.69**	**88.26**	**55.15**	**32.93**	**3.42**	**32.29**	**58.94**

3.7 Hallucination Verifier Evaluation

To validate the effectiveness of the HalluMirror Verifier, we compare it with existing hallucination detectors on POPE for object-existence hallucinations and FINER for more challenging fine-grained hallucinations. We include Self-Check using our LLM judge model, Qwen3.8-27B, to assess the added value of our visual-evidence-based verification pipeline over direct model self-checking. As shown in Table 5, HalluMirror achieves the best overall performance, reaching 90.00% on POPE and 82.59% average accuracy on FINER. Compared with the strongest baseline, Self-Check, HalluMirror performs comparably on POPE while achieving higher FINER accuracy, with a particularly large gain on relation hallucinations (81.60% vs. 76.00%) despite being slightly lower on attributes (79.50% vs. 81.28%). This demonstrates stronger overall capability for fine-grained hallucination verification, supporting reliable validation of scalable self-play-generated data.

Table 5: Hallucination detection performance on POPE and FINER. We report prediction accuracy (%), measured as the proportion of detector outputs matching the benchmark annotations. Best/second-best results are shown in **bold**/underline.

Detector	POPE	FINER			
	Random	Exist.↑	Attr.↑	Rel.↑	Avg.↑
Self-Check (Qwen3.8-27B)	89.90	85.40	**81.28**	76.00	80.89
FaithScore-OFA (Jing et al., 2024)	55.90	71.97	68.80	65.37	68.71
FaithScore-LLaVA (Jing et al., 2024)	61.90	48.05	47.50	61.20	52.25
UniHD (Chen et al., 2024a)	82.34	64.60	56.48	–	60.56
Perceval (Min et al., 2026)	87.60	62.60	48.55	50.80	53.98
HalluMirror Verifier (Ours)	**90.00**	**86.69**	79.50	**81.60**	**82.59**

4 Conclusion & Limitations

HalluMirror explores adversarial self-play as a mechanism for dynamic self-evolution in multimodal hallucination mitigation, enabling training signals to adapt to evolving model capabilities. This offers a path toward continual discovery and mitigation of model-specific hallucination weaknesses. Nevertheless, HalluMirror currently focuses on perceptual hallucinations within a fixed taxonomy, leaving reasoning-intensive hallucinations unexplored. Applying self-play to a broader range of hallucinations is a promising direction for future research.

AI USE STATEMENT

In this work, we used generative AI tools for limited language editing and polishing, assistance in identifying relevant literature for related work, and code debugging. We also used generative AI tools for training data synthesis and LLM-based evaluation. All AI-assisted content was reviewed by the authors: relevant literature and citations were verified against the original sources, code suggestions were tested for correctness, and all AI-assisted writing was manually reviewed and revised. We did not use generative AI to generate experimental results or draw final scientific conclusions. We take responsibility for the final content of this work, including text, claims, code, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide the methodological and experimental details required to reproduce HalluMirror in the main paper and Appendix. Specifically, the Appendix includes the complete self-play algorithm, training hyperparameters, prompt templates, and benchmark evaluation protocols. The supplementary material contains the corresponding implementation code for reproducing the experimental results. The complete codebase and datasets will be publicly released upon acceptance.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-VL Technical Report. *arXiv preprint arXiv:2511.21631*, 2025a.

Zechen Bai, Pichao Wang, Tianjun Xiao, Tong He, Zongbo Han, Zheng Zhang, and Mike Zheng Shou. Hallucination of Multimodal Large Language Models: A Survey. *arXiv preprint arXiv:2404.18930*, 2025b.

Yishuo Cai, Renjie Gu, Jiaxu Li, Xuancheng Huang, Junzhe Chen, Xiaotao Gu, and Minlie Huang. MHALO: Evaluating MLLMs as Fine-grained Hallucination Detectors. In *Findings of ACL*, pp. 9197–9222, 2025.

Xiang Chen, Chenxi Wang, Yida Xue, Ningyu Zhang, Xiaoyan Yang, Qiang Li, Yue Shen, Lei Liang, Jinjie Gu, and Huajun Chen. Unified hallucination detection for multimodal large language models. In *Annu. Meeting Assoc. Comput. Linguistics*, pp. 3235–3252, 2024a.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-Play Fine-Tuning Converts Weak Language Models to Strong Language Models. *arXiv preprint arXiv:2401.01335*, 2024b.

Pengyu Cheng, Yong Dai, Tianhao Hu, Han Xu, Zhisong Zhang, Lei Han, Nan Du, and Xiaolong Li. Self-playing Adversarial Language Game Enhances LLM Reasoning. In *Adv. Neural Inform. Process. Syst.*, pp. 126515–126543, 2024.

Yuhan Fu, Ruobing Xie, Xingwu Sun, Zhanhui Kang, and Xirong Li. Mitigating Hallucination in Multimodal Large Language Model via Hallucination-targeted Direct Preference Optimization. In *Findings of ACL*, pp. 16563–16577, 2025.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, et al. Hallusionbench: An Advanced Diagnostic Suite for Entangled Language Hallucination and Visual Illusion in Large Vision-Language Models. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 14375–14385, 2024.

Yicheng He, Chengsong Huang, Zongxia Li, Jiaxin Huang, and Yonghui Yang. VisPlay: Self-Evolving Vision-Language Models from Images. *arXiv preprint arXiv:2511.15661*, 2025.

Qidong Huang, Xiaoyi Dong, Pan Zhang, Bin Wang, Conghui He, Jiaqi Wang, Dahua Lin, Weiming Zhang, and Nenghai Yu. OPERA: Alleviating Hallucination in Multi-Modal Large Language Models via Over-Trust Penalty and Retrospection-Allocation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 13418–13427, 2024a.

Xiaowei Huang, Wenjie Ruan, Wei Huang, Gaojie Jin, Yi Dong, Changshun Wu, Saddek Bensalem, Ronghui Mu, Yi Qi, Xingyu Zhao, Kaiwen Cai, Yanghao Zhang, Sihao Wu, Peipei Xu, Dengyu Wu, Andre Freitas, and Mustafa A. Mustafa. A Survey of Safety and Trustworthiness of Large Language Models through the Lens of Verification and Validation. *Artif. Intell. Rev.*, 57(7):175, 2024b.

Fushuo Huo, Wenchao Xu, Zhong Zhang, Haozhao Wang, Zhicheng Chen, and Peilin Zhao. Self-Introspective Decoding: Alleviating Hallucinations for Large Vision-Language Models. In *Int. Conf. Learn. Represent.*, pp. 24272–24295, 2025.

Eun Woo Im, Muhammad Kashif Ali, and Vivek Gupta. Self-Aug: Query and Entropy Adaptive Decoding for Large Vision-Language Models. In *Int. Conf. Learn. Represent.*, pp. 136345–136368, 2026.

Weilin Jin, Mingyu Wang, Wenbo Li, Haoyang Huang, Yifan Wu, Ying Li, Gang Huang, and Zhonghai Wu. HalluScope: Fine-grained Hallucination Diagnosis for Multimodal Large Language Models. *arXiv preprint arXiv:2607.21105*, 2026.

Liqiang Jing, Ruosen Li, Yunmo Chen, and Xinya Du. FaithScore: Fine-grained Evaluations of Hallucinations in Large Vision-Language Models. In *Findings of EMNLP*, pp. 5042–5063, 2024.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *Int. J. Comput. Vis.*, 123(1):32–73, 2017.

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 *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 13872–13882, 2024.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji rong Wen. Evaluating Object Hallucination in Large Vision-Language Models. In *Conf. Empirical Methods Natural Lang. Process.*, pp. 292–305, 2023.

Bo Liu, Leon Guertler, Simon Yu, Zichen Liu, Penghui Qi, Daniel Balcells, Mickel Liu, Cheston Tan, Weiyan Shi, Min Lin, Wee Sun Lee, and Natasha Jaques. SPIRAL: Self-Play on Zero-Sum Games Incentivizes Reasoning via Multi-Agent Multi-Turn Reinforcement Learning. In *Int. Conf. Learn. Represent.*, pp. 35407–35434, 2026.

Fuxiao Liu, Kevin Lin, Linjie Li, Jianfeng Wang, Yaser Yacoob, and Lijuan Wang. Mitigating Hallucination in Large Multi-Modal Models via Robust Instruction Tuning. In *Int. Conf. Learn. Represent.*, pp. 57689–57733, 2024.

Yifan Lu, Ziqi Zhang, Chunfeng Yuan, Jun Gao, Congxuan Zhang, Xiaojuan Qi, Bing Li, and Weiming Hu. Mitigating Hallucinations in Large Vision-Language Models by Self-Injecting Hallucinations. *arXiv preprint arXiv:2509.11287*, 2025.

Yingqian Min, Kun Zhou, Yifan Li, Yuhuan Wu, Han Peng, Yifan Du, Wayne Xin Zhao, Min Yang, and Ji-Rong Wen. Improving Vision-language Models with Perception-centric Process Reward Models. *arXiv preprint arXiv:2604.24583*, 2026.

Yannic Neuhaus and Matthias Hein. RePOPE: Impact of Annotation Errors on the POPE Benchmark. *arXiv preprint arXiv:2504.15707*, 2025.

Cong-Duy Nguyen, Xiaobao Wu, Duc Anh Vu, Shuai Zhao, Thong Nguyen, and Anh Tuan Luu. CutPasteFind: Efficient Multimodal Hallucination Detector with Visual-aid Knowledge Base. *arXiv preprint arXiv:2502.12591*, 2025.

Lin Qiu, Hanqing Zeng, Yao Liu, Bingjun Sun, Guangdeng Liao, and Ji Liu. DUEL: Adversarial Self-Play for Multimodal Reasoning. *arXiv preprint arXiv:2605.24794*, 2026.

Pritish Sahu, Karan Sikka, and Ajay Divakaran. Pelican: Correcting Hallucination in Vision-LLMs via Claim Decomposition and Program of Thought Verification. In *Conf. Empirical Methods Natural Lang. Process.*, pp. 8228–8248, 2024.

Pritam Sarkar, Sayna Ebrahimi, Ali Etemad, Ahmad Beirami, Sercan Arik, and Tomas Pfister. Mitigating Object Hallucination in MLLMs via Data-augmented Phrase-level Alignment. In *Int. Conf. Learn. Represent.*, pp. 73796–73846, 2025.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv preprint arXiv:2402.03300*, 2024.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. HybridFlow: A Flexible and Efficient RLHF Framework. In *EuroSys*, pp. 1279–1297, 2025.

Zhiqing Sun, Sheng Shen, Shengcao Cao, Haotian Liu, Chunyuan Li, Yikang Shen, Chuang Gan, Liang-Yan Gui, Yu-Xiong Wang, Yiming Yang, Kurt Keutzer, and Trevor Darrell. Aligning Large Multimodal Models with Factually Augmented RLHF. In *Findings of ACL*, pp. 13088–13110, 2024.

V Team, Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, Shuaiqi Duan, Weihan Wang, Yan Wang, Yean Cheng, Zehai He, Zhe Su, Zhen Yang, Ziyang Pan, Aohan Zeng, Baoxu Wang, Bin Chen, Boyan Shi, Changyu Pang, Chenhui Zhang, Da Yin, Fan Yang, Guoqing Chen, Haochen Li, Jiale Zhu, Jiali Chen, Jiaxing Xu, Jiazheng Xu, Jing Chen, Jinghao Lin, Jinhao Chen, Jinjiang Wang, Junjie Chen, Leqi Lei, Letian Gong, Leyi Pan, Mingdao Liu, Mingde Xu, Mingzhi Zhang, Qinkai Zheng, Ruiliang Lyu, Shangqin Tu, Sheng Yang, Shengbiao Meng, Shi Zhong, Shiyu Huang, Shuyuan Zhao, Siyan Xue, Tianshu Zhang, Tianwei Luo, Tianxiang Hao, Tianyu Tong, Wei Jia, Wenkai Li, Xiao Liu, Xiaohan Zhang, Xin Lyu, Xinyu Zhang, Xinyue Fan, Xuancheng Huang, Yadong Xue, Yanfeng Wang, Yanling Wang, Yanzi Wang, Yifan An, Yifan Du, Yiheng Huang, Yilin Niu, Yiming Shi, Yu Wang, Yuan Wang, Yuanchang Yue, Yuchen Li, Yusen Liu, Yutao Zhang, Yuting Wang, Yuxuan Zhang, Zhao Xue, Zhengxiao Du, Zhenyu Hou, Zihan Wang, Peng Zhang, Debing Liu, Bin Xu, Juanzi Li, Minlie Huang, Yuxiao Dong, and Jie Tang. GLM-4.5V and GLM-4.1V-Thinking: Towards Versatile Multimodal Reasoning with Scalable Reinforcement Learning. *arXiv preprint arXiv:2507.01006*, 2025.

Omkar Thawakar, Shravan Venkatraman, Ritesh Thawkar, Abdelrahman Shaker, Hisham Cholakkal, Rao Muhammad Anwer, Salman Khan, and Fahad Khan. EvoLMM: Self-Evolving Large Multimodal Models with Continuous Rewards. *arXiv preprint arXiv:2511.16672*, 2025.

Qinsi Wang, Bo Liu, Tianyi Zhou, Jing Shi, Yueqian Lin, Yiran Chen, Hai Li, Kun Wan, and Wentian Zhao. Vision-Zero: Scalable VLM Self-Evolution via Multi-Agent Self-Play. In *Int. Conf. Learn. Represent.*, pp. 106716–106744, 2026a.

Shihao Wang, Shilong Liu, Yuanguo Kuang, Xinyu Wei, Yangzhou Liu, Zhiqi Li, Yunze Man, Guo Chen, Andrew Tao, Guilin Liu, Jan Kautz, Lei Zhang, and Zhiding Yu. LocateAnything: Fast and High-Quality Vision-Language Grounding with Parallel Box Decoding. *arXiv preprint arXiv:2605.27365*, 2026b.

Weiyun Wang, Yiming Ren, Haowen Luo, Tiantong Li, Chenxiang Yan, Zhe Chen, Wenhai Wang, Qingyun Li, Lewei Lu, Xizhou Zhu, Yu Qiao, and Jifeng Dai. The All-Seeing Project V2: Towards General Relation Comprehension of the Open World. In *Eur. Conf. Comput. Vis.*, pp. 471–490, 2024a.

Zhecan Wang, Garrett Bingham, Adams Wei Yu, Quoc V Le, Thang Luong, and Golnaz Ghiasi. HaloQuest: A Visual Hallucination Dataset for Advancing Multimodal Reasoning. In *Eur. Conf. Comput. Vis.*, pp. 288–304, 2024b.

Junfei Wu, Qiang Liu, Ding Wang, Jinghao Zhang, Shu Wu, Liang Wang, and Tieniu Tan. Logical Closed Loop: Uncovering Object Hallucinations in Large Vision-Language Models. In *Findings of ACL*, pp. 6944–6962, 2024.

Rui Xiao, Sanghwan Kim, Yongqin Xian, Zeynep Akata, and Stephan Alaniz. FINER: MLLMs Hallucinate under Fine-grained Negative Queries. *arXiv preprint arXiv:2603.17662*, 2026.

Wenyi Xiao, Ziwei Huang, Leilei Gan, Wanggui He, Haoyuan Li, Zhelun Yu, Fangxun Shu, Hao Jiang, and Linchao Zhu. Detecting and Mitigating Hallucination in Large Vision Language Models via Fine-Grained AI Feedback. In *AAAI*, pp. 25543–25551, 2025.

Shiping Yang, Shining Liang, Weihao Liu, Wenbiao Ding, Linjun Shou, Lu Cheng, and Angel X. Chang. Hallucination Self-Play: Bootstrapping Reinforced Detector via Evolved Generator. *arXiv preprint arXiv:2607.07993*, 2026.

Zhihe Yang, Xufang Luo, Dongqi Han, Yunjian Xu, and Dongsheng Li. Mitigating Hallucinations in Large Vision-Language Models via DPO: On-Policy Data Hold the Key. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 10610–10620, 2025.

Yuan Yao, Tianyu Yu, Chongyi Wang, Junbo Cui, Bokai Xu, Hongji Zhu, Tianchi Cai, Fuwei Huang, Tianran Wang, Wenshuo Ma, Yixuan Zhou, Haoye Zhang, Zonghao Guo, Chi Chen, Haoyu Wang, Zhihui He, Haoyu Li, Hanyu Liu, Luoyuan Zhang, Ge Zhou, Siyuan Li, Zhi Zheng, Jie Zhou, Yuxuan Li, Kaihuo Zhang, Yudong Mei, Hanqing Zhao, Yueying Chen, Zhongwu Zhai, Hanbin Wang, Ganqu Cui, Ning Ding, Xu Han, Zhiyong Wu, Zhiyuan Liu, and Maosong Sun. MiniCPM-o 2.6: A GPT-4o Level MLLM for Vision, Speech, and Multimodal Live Streaming on Your Phone, 2025. URL `https://github.com/OpenBMB/MiniCPM-o`.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Tong Xu, Hao Wang, Dianbo Sui, Yunhang Shen, Ke Li, Xing Sun, and Enhong Chen. Woodpecker: Hallucination correction for multimodal large language models. *Sci. China Inf. Sci.*, 67(12):220105, 2024.

Zhihan Yin, Jianxin Liang, Yueqian Wang, Yao Yifeng, Huishuai Zhang, and Dongyan Zhao. FREAK: A Fine-grained Hallucination Evaluation Benchmark for Advanced MLLMs. In *Int. Conf. Learn. Represent.*, pp. 49651–49684, 2026.

Qifan Yu, Juncheng Li, Longhui Wei, Liang Pang, Wentao Ye, Bosheng Qin, Siliang Tang, Qi Tian, and Yueting Zhuang. HalluciDoctor: Mitigating Hallucinatory Toxicity in Visual Instruction Data. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 12944–12953, 2024a.

Tianyu Yu, Yuan Yao, Haoye Zhang, Taiwen He, Yifeng Han, Ganqu Cui, Jinyi Hu, Zhiyuan Liu, Hai-Tao Zheng, Maosong Sun, and Tat-Seng Chua. RLHF-V: Towards Trustworthy MLLMs via Behavior Alignment from Fine-Grained Correctional Human Feedback. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 13807–13816, 2024b.

Tianyu Yu, Haoye Zhang, Qiming Li, Qixin Xu, Yuan Yao, Da Chen, Xiaoman Lu, Ganqu Cui, Yunkai Dang, Taiwen He, Xiaocheng Feng, Jun Song, Bo Zheng, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. RLAIF-V: Open-Source AI Feedback Leads to Super GPT-4V Trustworthiness. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 19985–19995, 2025.

Huining Yuan, Zelai Xu, Zheyue Tan, Xiangmin Yi, Mo Guang, Kaiwen Long, Haojia Hui, Boxun Li, Xinlei Chen, Bo Zhao, Xiao-Ping Zhang, Chao Yu, and Yu Wang. MARSHAL: Incentivizing Multi-Agent Reasoning via Self-Play with Strategic LLMs. In *Int. Conf. Learn. Represent.*, pp. 6209–6236, 2026.

Xiaofeng Zhang, Yuanchao Zhu, Chaochen Gu, Xiaosong Yuan, Qiyan Zhao, Jiawei Cao, Barrett Tang, Sinan Fan, Yaomin Shen, Chen Shen, and Hao Tang. Hallucination Begins Where Saliency Drops. In *Int. Conf. Learn. Represent.*, pp. 15062–15083, 2026.

Andrew Zhao, Yiran Wu, Yang Yue, Tong Wu, Quentin Xu, Yang Yue, Matthieu Lin, Shenzhi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. Absolute Zero: Reinforced Self-play Reasoning with Zero Data. *arXiv preprint arXiv:2505.03335*, 2025.

Tianyi Zhao, Liangliang Zhang, Yao Ma, and Lu Cheng. A Survey on Safe Multi-Modal Learning Systems. In *ACM SIGKDD Conf. Knowl. Discov. Data Min.*, pp. 6655–6665, 2024.

Zhiyuan Zhao, Bin Wang, Linke Ouyang, Xiaoyi Dong, Jiaqi Wang, and Conghui He. Beyond Hallucinations: Enhancing LVLMs through Hallucination-Aware Direct Preference Optimization. *arXiv preprint arXiv:2311.16839*, 2023.

Yiyang Zhou, Chenhang Cui, Jaehong Yoon, Linjun Zhang, Zhun Deng, Chelsea Finn, Mohit Bansal, and Huaxiu Yao. Analyzing and Mitigating Object Hallucination in Large Vision-Language Models. In *Int. Conf. Learn. Represent.*, pp. 56969–56998, 2024.

APPENDIX

In this appendix, we provide additional related work, implementation details, experimental setup, more results and analyses and discussions of limitations and potential impacts. The appendix is organized as follows.

A. Related Work

1. Multimodal Hallucination Mitigation
2. Self-Play and Self-Evolving Learning
3. Multimodal Hallucination Detection

B. Implementation Details

1. Self-Play Algorithm
2. Training Hyperparameters
3. Prompt Templates

C. Experimental Setup

1. Benchmarks and Evaluation Metrics
2. Baseline Reproduction Details
3. Ablation Settings

D. More Results and Analyses

1. Fine-Grained Robustness across Hallucination Dimensions
2. Self-Play Dynamics
3. Qualitative Examples and Failure Analysis

E. Limitations and Future Work

A RELATED WORK

A.1 MULTIMODAL HALLUCINATION MITIGATION

Multimodal hallucination refers to inconsistencies between model-generated content and visual evidence, typically involving object existence, attributes, and relations (Bai et al., 2025b). Training-free methods mitigate hallucinations at inference time through contrastive decoding (Leng et al., 2024; Im et al., 2026), internal-signal-guided generation (Huang et al., 2024a; Zhang et al., 2026), or post-hoc revision (Zhou et al., 2024; Yin et al., 2024), but often incur additional inference cost without improving the model itself. Training-based methods instead enhance hallucination robustness using constructed negative supervision (Liu et al., 2024; Yu et al., 2024a), human or AI feedback (Yu et al., 2024b; Zhao et al., 2023), or targeted preference data (Fu et al., 2025; Xiao et al., 2026), yet their effectiveness remains constrained by the quality and coverage of the constructed supervision. Recent iterative approaches (Yu et al., 2025; Lu et al., 2025) further refresh feedback or self-generated hallucinations during training, but their supervision is not explicitly coupled to the model's evolving fine-grained failure profile, making it difficult to continuously concentrate learning on its most vulnerable hallucination modes. HalluMirror addresses this gap by explicitly optimizing hallucination generation against the current Challenger, allowing training signals to adapt to its evolving weaknesses.

A.2 SELF-PLAY AND SELF-EVOLVING LEARNING

Self-play and self-evolving learning have recently emerged as important paradigms for reducing dependence on external supervision and improving models' autonomous learning capabilities. In

language models, existing approaches promote continual improvement through self-generated preferences and training signals (Chen et al., 2024b), game-driven capability evolution (Cheng et al., 2024; Liu et al., 2026; Yuan et al., 2026), and task generation and solving (Zhao et al., 2025). In multimodal models, related studies further construct self-generated or consistency-based learning signals from unlabeled images (He et al., 2025; Wang et al., 2026a; Thawakar et al., 2025), while adversarial generation and visual verification have been explored to enhance visual reasoning (Qiu et al., 2026). More closely related to hallucination research, Lu et al. (2025) continually updates multimodal preference supervision through self-injected hallucinations and iterative curricula, but its training difficulty is primarily governed by predefined injection strategies and curriculum schedules rather than the model's current fine-grained failures. Yang et al. (2026) constructs a dynamic hallucination curriculum through Generator–Detector self-play, but focuses on textual hallucination detection in RAG settings. These studies highlight self-play as a scalable mechanism for capability-adaptive optimization. HalluMirror extends this paradigm to fine-grained multimodal hallucination mitigation by jointly evolving hallucination generation, detection, and repair under visual-grounding-based verification.

A.3 Multimodal Hallucination Detection

Post-hoc multimodal hallucination detection examines whether generated responses are supported by visual evidence. Existing approaches can be broadly organized into training-free detection and learned hallucination detection. Within training-free detection, methods based on visual entailment, question answering, or consistency checking assess generated content using existing models: Jing et al. (2024) verifies extracted atomic facts against the image and aggregates the results into faithfulness scores, while Wu et al. (2024) identifies potential object hallucinations through logical consistency in object–attribute question-answering loops. These approaches depend on the verifying model's visual reliability, and consistent answers alone do not guarantee visual correctness. Tool-assisted methods (Chen et al., 2024a; Yin et al., 2024; Sahu et al., 2024) gather evidence through visual perception tools or executable verification programs to assess claims and, in some cases, guide subsequent correction. Nguyen et al. (2025) further combines object detection, visual question answering, and similarity checks against a visual knowledge base. Such pipelines provide explicit intermediate evidence, but depend on the capabilities of their constituent modules and can propagate errors across decomposition, perception, and evidence interpretation. Learned detectors acquire verification capabilities through dedicated training with fine-grained supervision: Cai et al. (2025) localizes hallucinated spans, Min et al. (2026) identifies perceptually erroneous claims for token-level feedback, and Jin et al. (2026) combines localization, type classification, and diagnostic explanations. Despite providing detailed feedback, these approaches require additional detector training and remain dependent on the quality and coverage of their supervision. Building on evidence-based verification, our approach uses a unified visual grounding model to validate existence, attribute, and relation hallucinations without task-specific detector training.

B Implementation Details

B.1 Self-Play Algorithm

Algorithm 1 summarizes HalluMirror's self-play procedure, including policy initialization, alternating Inducer–Challenger optimization, and difficulty-aware data curation.

B.2 Training Hyperparameters

Table 6 summarizes the core optimization and self-play hyperparameters used in our experiments. Unless otherwise specified, we use the same self-play schedule across all three backbone models. Please refer to our open-source codebase for additional hyperparameter details and complete reproducible experiment examples. Each self-play iteration of GLM-4.1V-9B takes approximately 24 hours on two NVIDIA A800 80GB GPUs. For Qwen3-VL-8B and MiniCPM-o-2.6, we use a single NVIDIA A800 80GB GPU, with each iteration taking approximately 20 hours.

Algorithm 1: HalluMirror: Co-Evolutionary Hallucination Self-Play

Input: Faithful image–description pairs $\mathcal{D}$; pretrained MLLM; edit-count distributions $\{p_t\}_{t=1}^{N}$; group size G; number of iterations N.
Output: Evolved Inducer π_i^N and Challenger π_c^N.

Initialize π_i^0 through hallucination-injection SFT and π_c^0 directly from the same pretrained MLLM
for $t \leftarrow 1$ **to** N **do**
  // Adversarial hallucination induction
  Set $\pi_i \leftarrow \pi_i^{t-1}$, freeze π_c^{t-1}, and initialize $\mathcal{R}_t \leftarrow \varnothing$
  for *each Inducer training step* **do**
    Sample a minibatch from $\mathcal{D}$ and assign edit counts $K \sim p_t$
    For each (I, c), sample G edit sequences $e_{1:K}^{(g)} \sim \pi_i(\cdot \mid I, c)$ and construct $\tilde{c}^{(g)} = T(c, e_{1:K}^{(g)})$
    Obtain sentence-level hallucination predictions $\hat{\mathbf{y}}^{(g)}$ from π_c^{t-1} for each $(I, \tilde{c}^{(g)})$
    Compute $R_i^{(g)}$ using Eq. (3) and $\hat{A}_i^{(g)}$ using Eq. (5); update π_i with GRPO
    Store rollouts and detection feedback in $\mathcal{R}_t$
  Set π_i^t to the selected Inducer checkpoint
  // Difficulty-aware data curation
  Validate rollouts through format, editing-rule checks and hallucination verification
  Rank the remaining rollouts by $d(r)$ in Eq. (2), retaining the hardest one per source pair (I, c)
  Construct $\mathcal{D}_t^{\mathrm{adv}}$ from the middle 70%, retaining injection-derived labels $\mathbf{y}$ and faithful references c
  // Hallucination detection and repair
  Set $\pi_c \leftarrow \pi_c^{t-1}$
  for *each Challenger training step* **do**
    Sample a minibatch from $\mathcal{D}_t^{\mathrm{adv}}$
    For each $(I, \tilde{c})$, sample G responses $(\hat{\mathbf{y}}^{(g)}, \hat{c}^{(g)}) \sim \pi_c(\cdot \mid I, \tilde{c})$
    Compute $R_c^{(g)}$ using Eq. (4) and $\hat{A}_c^{(g)}$ using Eq. (5); update π_c with GRPO
  Set π_c^t to the selected Challenger checkpoint
return π_i^N, π_c^N

B.3 Prompt Templates

This section provides the core prompts used in HalluMirror. The hallucination verifier consists of two prompting stages: joint claim decomposition and grounding-query planning, followed by grounding-evidence interpretation, as detailed in Hallucination Verifier Prompts. The self-play process uses role-specific prompts for the Inducer, which generates fine-grained hallucinations, and the Challenger, which detects and repairs them, as detailed in Self-Play Prompts.

C Experimental Setup

C.1 Benchmarks and Evaluation Metrics

We evaluate all models on eight complementary hallucination benchmarks, whose coverage is summarized in Table 7. POPE (Li et al., 2023) and RePOPE (Neuhaus & Hein, 2025) provide controlled binary probes of object-existence hallucination under random, popular, and adversarial negative sampling. RePOPE further reduces evaluation noise through human-reannotated labels and the removal of ambiguous cases. HallusionBench (Guan et al., 2024) evaluates entangled visual illusions and language-induced hallucinations using paired questions and images. CRPE (Wang et al., 2024a) examines compositional relation understanding by separately probing relation existence, subjects, predicates, and objects, whereas FINER-CompreCap (Xiao et al., 2026) targets fine-grained negative queries involving multiple objects, attributes, and relations. FREAK (Yin et al., 2026) further broadens the evaluation to both multiple-choice and free-form tasks spanning object recognition, attributes and counting, spatial relations, visual reasoning, and OCR. Finally, MMHal-Bench (Sun et al., 2024) and HaloQuest (Wang et al., 2024b) assess grounded open-ended generation under diverse real-world failure cases, including false-premise, visually challenging, and insufficient-context questions. Collectively, these benchmarks span binary, multiple-choice, and open-ended response formats while covering both coarse-grained object existence and fine-grained visual semantics. They

Table 6: Core hyperparameters for self-play training.

Parameter	Value
Model and Sequence Settings	
Maximum prompt length	4096
Maximum response length	1024
Maximum rollout model length	6144
LoRA rank / scaling factor	$64/64$
Optimization Settings	
Optimizer	AdamW
Learning rate	3×10^{-6}
Warm-up ratio	0.03
Weight decay	0.01
Training batch size	8
Policy mini-batch size	8
Gradient norm clipping	1.0
RL Settings	
Policy clipping ratio	0.2
Policy-update epochs	1
KL loss coefficient	0.01
KL reward penalty coefficient	0.0
Rollouts per prompt	6
Rollout decoding	Maximum new tokens $= 1024$, temperature $= 1.0$, top-$p = 1.0$
Maximum tokens per GPU	8192
Self-Play Settings	
Number of self-play rounds	3
Inducer samples per round	1500
Hallucination-injection distribution	Round 1: $[0.15, 0.35, 0.35, 0.15]$ for 1–4 edits Round 2: $[0.30, 0.40, 0.20, 0.10]$ for 1–4 edits Round 3: $[0.55, 0.30, 0.15]$ for 1–3 edits
Challenger-data retention ratio	Middle 70% according to Challenger difficulty
Minimum transform consistency	1.0
Opponent-model decoding	Maximum new tokens $= 1024$, temperature $= 0$, top-$p = 1.0$
Inducer reward coefficients	$(\lambda_{\text{fn}}, \lambda_{\text{fp}}) = (1.6, 0.4)$
Challenger reward coefficients	$(\lambda_{\text{det}}, \lambda_{\text{rep}}) = (1.0, 1.0)$, with $R_{\text{rep}} = 0.8R_{\text{sem}} + 0.2R_{\text{edit}}$

Table 7: Coverage of the evaluation benchmarks across answer formats, multimodal capabilities, and benchmark designs.

Dataset	Scale	Answer Format			Evaluated Capability					Benchmark Design			
	Img./Q.	Y/N	MCQ	Open	Obj.	Attr. /Cnt.	Rel. /Spa.	Reas.	OCR	Unans.	Illus.	Prior	Reann.
POPE (Li et al., 2023)	500 / 9,000	✓	–	–	✓	–	–	–	–	–	–	✓	–
RePOPE (Neuhaus & Hein, 2025)	500 / 8,185	✓	–	–	✓	–	–	–	–	–	–	✓	✓
HallusionBench (Guan et al., 2024)	346 / 1,129	✓	–	–	–	–	–	✓	✓	–	✓	✓	–
CRPE (Wang et al., 2024a)	1,536 / 6,448	–	✓	–	✓	–	✓	–	–	–	–	✓	–
FINER-CompreCap (Xiao et al., 2026)	560 / 17,084	–	✓	–	✓	✓	✓	–	–	✓	–	–	–
FREAK (Yin et al., 2026)	1,786 / 1,799	–	✓	✓	✓	✓	✓	✓	✓	–	–	✓	–
MMHal-Bench (Sun et al., 2024)	96 / 96	–	–	✓	✓	✓	✓	✓	✓	✓	–	–	–
HaloQuest (Wang et al., 2024b)	375 / 608	–	–	✓	–	✓	✓	✓	–	✓	–	✓	–

Abbreviations. ✓ indicates that the corresponding dimension is explicitly included as a task, category, or benchmark construction; – indicates that it is not explicitly targeted. **Img./Q.**: number of unique images / original questions; **Y/N**: binary yes/no questions; **MCQ**: multiple-choice questions; **Open**: open-ended response generation; **Obj.**: object existence or detection; **Attr./Cnt.**: attribute recognition or object counting; **Rel./Spa.**: relational or spatial understanding; **Reas.**: image-grounded visual reasoning; **OCR**: recognition or understanding of text and symbols in images; **Unans.**: false-premise, ill-posed, or insufficient-context questions; **Illus.**: visual illusions or deliberately misleading visual patterns; **Prior**: benchmark constructions targeting frequency, co-occurrence, linguistic, commonsense, or parametric priors; **Reann.**: human-reannotated labels with ambiguous cases removed.

capture hallucinations caused by visual ambiguity, misleading priors, unanswerable questions, and annotation noise. For every benchmark, we use its released evaluation set and follow the official task formulation, answer interpretation or judging procedure, and sample-level scoring protocol. When a benchmark reports multiple subsets, we additionally aggregate the corresponding official subset scores to obtain a compact benchmark-level result for the main table. In particular, Pair Acc. is a strict consistency metric: HallusionBench credits a matched question pair only when all paired instances are answered correctly, whereas FINER credits a positive–negative pair only when both questions are correct. CRPE Acc. similarly counts a circular sample as correct only when all four answer-option rotations are answered correctly. The reported FINER result is the macro-average of Pair Acc. over its four subsets, while FREAK first averages the six option permutations of each multiple-choice item before averaging over the original samples. The final Avg. is the unweighted mean of the primary metric for each benchmark, using F1 for POPE and RePOPE and $100 - \mathrm{HR}$ for MMHal-Bench.

To evaluate the fine-grained hallucination detection capability of the HalluMirror Verifier, we use FINER (Xiao et al., 2026), which covers existence, attribute, and relation hallucinations. For each evaluation instance, the verifier takes the image, question, and model response as input and predicts both whether the response contains a hallucination and its hallucination type. We compare the predicted hallucination status and type with the corresponding ground-truth annotations and measure the proportion of predictions that match the ground truth. This evaluation assesses the verifier's ability to both identify hallucinated responses and distinguish among fine-grained hallucination categories.

C.2 Baseline Reproduction Details

For controlled comparison, we adapted all baselines to the same GLM-4.1V-9B backbone and evaluated them using identical benchmark splits and scoring protocols. For trainable baselines, we preserved their method-defining objectives and architecture-independent hyperparameters whenever applicable, while replacing model-specific components, such as tokenization, conversation templates, visual preprocessing, and module mappings, with their native GLM counterparts. Specifically, FINER (Xiao et al., 2026) was reproduced using its officially released 160K preference dataset, with 159,200 samples used for DPO training and 800 reserved for validation. HSA-DPO (Xiao et al., 2025) was trained on the officially released 8,386 preference pairs while retaining its hallucination-severity-aware preference objective. HALVA (Sarkar et al., 2025) used the released 21,874 contrastive samples and an equally sized reference subset, preserving both its phrase-level contrastive objective and forward-KL regularization. For OPA-DPO (Yang et al., 2025), we generated on-policy responses using GLM, revised them through a fixed multimodal LLM judge, and followed the original two-stage OPA-SFT and multi-pair OPA-DPO optimization procedure. For RLAIF-V (Yu et al., 2025), we trained GLM for two epochs on all 83,132 officially released preference pairs rather than repeating the full iterative data-generation process. This released-data transfer setting is supported by the original study's findings that RLAIF-V feedback data generalize across different MLLMs and consistently reduce hallucination. Finally, VCD (Leng et al., 2024), SID (Huo et al., 2025), and Self-Aug (Im et al., 2026) were implemented as training-free decoding baselines by transferring their respective contrastive decoding mechanisms to GLM without additional training data or parameter updates. Overall, these implementations preserve the essential learning or decoding mechanism of each baseline while controlling for backbone and evaluation differences, and should therefore be interpreted as method-level reproductions on GLM rather than numerical reproductions of the original model-specific checkpoints.

C.3 Ablation Settings

We design two ablations to disentangle the effects of the optimization objective and adaptive self-play. In both settings, the initial Inducer is frozen to construct a static hallucination dataset that matches the cumulative training-data size and hallucination-count distributions of the three-round main experiment. Static SFT directly trains the Challenger to recover the original faithful description from its hallucinated counterpart, thereby removing both GRPO optimization and adaptive data generation. Fixed-Data GRPO trains the Challenger on the same static dataset while retaining the original prompt, reward functions, and GRPO configuration, thus removing only adaptive data generation. Therefore, comparing Static SFT with Fixed-Data GRPO isolates the effect of the optimiza-

tion paradigm, whereas comparing Fixed-Data GRPO with HalluMirror reveals the contribution of adapting the Inducer to the evolving Challenger.

D More Results and Analyses

D.1 Fine-Grained Robustness across Hallucination Dimensions

Table 8: Fine-grained results on the three POPE and RePOPE subsets for GLM-4.1V-9B. Small numbers denote performance improvements or degradations relative to the corresponding base model. Best/second-best results within each benchmark subset are shown in **bold**/underline.

Method	POPE			RePOPE		
	Acc.↑	Rec.↑	F1↑	Acc.↑	Rec.↑	F1↑
Random						
Base	87.87	76.20	86.26	94.09	88.87	92.63
Iter. 1	90.37 +2.50	81.60 +5.40	89.44 +3.18	95.03 +0.94	92.49 +3.62	93.95 +1.32
Iter. 2	90.50 +2.63	81.87 +5.67	89.60 +3.34	94.95 +0.86	92.49 +3.62	93.87 +1.24
Iter. 3	**91.70** +3.83	**84.60** +8.40	**91.07** +4.81	**95.10** +1.01	**94.05** +5.18	**94.13** +1.50
Popular						
Base	86.83	76.07	85.24	92.56	86.84	91.08
Iter. 1	88.60 +1.77	81.73 +5.66	87.76 +2.52	**93.29** +0.73	91.03 +4.19	92.23 +1.15
Iter. 2	88.57 +1.74	81.93 +5.86	87.75 +2.51	**93.29** +0.73	91.11 +4.27	92.24 +1.16
Iter. 3	**89.20** +2.37	**84.53** +8.46	**88.67** +3.43	93.18 +0.62	**93.04** +6.20	**92.27** +1.19
Adversarial						
Base	86.03	76.07	84.49	92.06	87.53	90.70
Iter. 1	87.13 +1.10	81.60 +5.53	86.38 +1.89	92.21 +0.15	91.41 +3.88	91.21 +0.51
Iter. 2	87.13 +1.10	81.93 +5.86	86.43 +1.94	**92.25** +0.19	91.49 +3.96	**91.26** +0.56
Iter. 3	**87.23** +1.20	**84.53** +8.46	**86.88** +2.39	91.51 -0.55	**93.18** +5.65	90.66 -0.04

To examine whether the aggregate improvements of HalluMirror persist across different hallucination types and granularities, we conduct a focused subset-level analysis on POPE, RePOPE, FINER, and CRPE. As summarized in Table 7, these benchmarks provide explicit and complementary subdivisions that closely align with the hallucination dimensions targeted by HalluMirror. POPE and RePOPE isolate object-existence hallucination under random, popular, and adversarial negative sampling, thereby covering different object-frequency and co-occurrence priors. RePOPE further employs reannotated examples to reduce label ambiguity. FINER extends the evaluation to multi-object, multi-attribute, multi-relation, and Wh-question cases, while CRPE decomposes relational understanding into existence, subject, predicate, and object components. Their structured answer formats and semantically interpretable subsets enable direct comparison across controlled hallucination factors.

Table 9: Fine-grained results on the FINER and CRPE subsets for GLM-4.1V-9B. Small numbers denote performance improvements or degradations relative to the corresponding base model. Best/second-best results within each benchmark subset are shown in **bold**/underline.

Method	FINER				CRPE			
	Multi-Obj.↑	Multi-Attr.↑	Multi-Rel.↑	Wh.↑	Exist.↑	Subj.↑	Pred.↑	Obj.↑
Base	70.92	66.27	42.24	26.15	95.13	78.72	67.80	72.39
Iter. 1	72.10 +1.18	66.21 -0.06	43.27 +1.03	27.04 +0.89	95.08 -0.05	78.93 +0.21	67.44 -0.36	72.39 0.00
Iter. 2	72.54 +1.62	66.69 +0.42	44.86 +2.62	27.98 +1.83	95.17 +0.04	78.72 0.00	**67.90** +0.10	**72.70** +0.31
Iter. 3	**73.14** +2.22	**69.32** +3.05	**48.36** +6.12	**29.75** +3.60	**95.21** +0.08	**79.34** +0.62	67.71 -0.09	**72.70** +0.31

As shown in Tables 8 and 9, Iter. 3 improves 23 of the 26 reported subset-level metrics over the base model. On POPE, it achieves consistent gains in accuracy, recall, and F1 across the random, popular, and adversarial subsets. On RePOPE, it improves all metrics on the random and popular subsets and substantially increases adversarial recall by 5.65 points, although adversarial accuracy and F1 decrease slightly by 0.55 and 0.04 points, respectively. Iter. 3 also improves all four FINER dimensions, with gains of 2.22, 3.05, 6.12, and 3.60 points on Multi-Obj., Multi-Attr., Multi-Rel., and Wh., respectively. On CRPE, it improves existence, subject, and object performance, while predicate accuracy decreases by 0.09 points. Although the optimal iteration varies across several

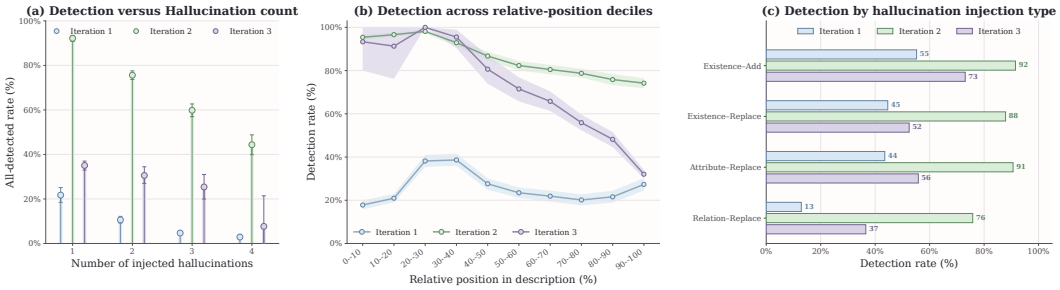

Figure 4: Fine-grained Challenger weaknesses exposed during Inducer optimization on GLM-4.1V-9B. (a) Complete-detection rate as a function of the number of hallucinations injected into each description. (b) Hallucination detection rate across relative-position deciles of the generated description. (c) Detection rate for different hallucination injection types. Colors denote different self-play iterations. Error bars in (a) and shaded regions in (b) represent 95% cluster-bootstrap confidence intervals over input examples.

dimensions and a few minor regressions remain, the predominantly positive results demonstrate that HalluMirror yields broad improvements across different negative-sampling strategies, hallucination types, and evaluation granularities.

D.2 Self-Play Dynamics

We further analyze the hallucination patterns revealed during self-play from three complementary perspectives, as shown in Fig. 4. **(i) Multiple hallucinations amplify detection difficulty.** As shown in Fig. 4(a), the probability of detecting all injected hallucinations consistently decreases as the number of hallucinations increases, indicating that multiple simultaneous errors substantially increase the difficulty of comprehensive detection. Across iterations, the Iteration-2 Challenger achieves the highest complete-detection rate after being optimized against the preceding attacks, while the subsequent decline in Iteration 3 suggests that the adapted Inducer successfully discovers new weaknesses in the updated Challenger. **(ii) Hallucinations near the end of a response are more difficult to detect.** Figure 4(b) reveals a clear positional effect: detection performance generally decreases as hallucinations occur later in the generated description, with the degradation becoming particularly pronounced toward the final portion of the output. **(iii) Semantically subtle replacements expose type-specific weaknesses.** As illustrated in Fig. 4(c), detection difficulty also depends strongly on the hallucination injection type. Replacing an existing object with a plausible co-occurring but visually absent object is consistently harder to detect than directly adding a nonexistent object. Attribute and especially relation replacements further expose substantial weaknesses, with relation hallucinations exhibiting the lowest detection rates across all iterations. Together, these results demonstrate that self-play progressively uncovers structured Challenger vulnerabilities associated with hallucination density, relative position, and injection type, rather than merely generating uniformly difficult examples.

We conducted a cross-play experiment to examine whether the Inducer progressively generates harder-to-detect hallucinations and whether the Challenger acquires increasingly robust detection capabilities. Specifically, we selected 100 held-out images that were disjoint from the self-play training data, used four Inducer checkpoints (I_0–I_3) to construct separate attack sets, and evaluated every attack set against all four Challenger checkpoints (C_0–C_3). As shown in Fig. 5, the detection rate generally decreases for later Inducers when the Challenger is fixed, indicating that self-play provides increasingly challenging training signals. Conversely, the final Challenger C_3 substantially outperforms the initial C_0 across all attack sets, demonstrating an overall improvement in hallucination detection. A particularly clear co-evolutionary pattern appears on the I_3 attacks: the detection rate drops to only 11% for C_2, suggesting that I_3 successfully exposes a previously unresolved weakness, whereas the subsequently optimized C_3 raises the rate to 93% on the identical attack set. Together, these results provide direct evidence that self-play alternately discovers new Challenger vulnerabilities and improves its ability to detect the resulting hallucinations.

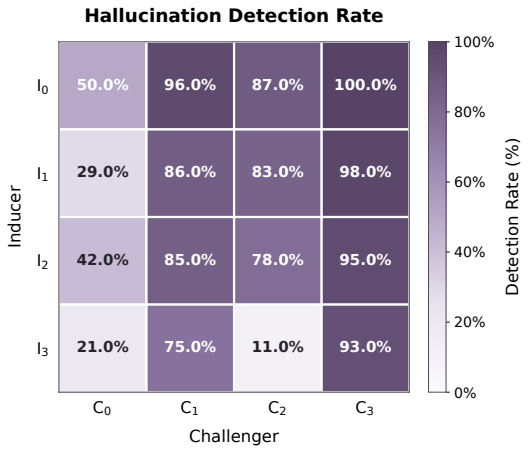

Figure 5: **Cross-play evaluation across self-play iterations.** Rows denote attacks generated by Inducers I_0–I_3, while columns denote Challenger checkpoints C_0–C_3.

D.3 Qualitative Examples and Failure Analysis

Self-play progressively strengthens both hallucination induction and detection. As illustrated in Fig. 7, the Inducer evolves from relatively conspicuous perturbations, such as simple color substitutions, toward more subtle and challenging hallucinations involving object existence and fine-grained relations. For example, later iterations alter structural relations between train carriages or the spatial configuration of a fire hydrant's shadow, while largely preserving the surrounding description. Meanwhile, Fig. 8 shows a corresponding improvement in the Challenger: hallucinations involving spatial relations and object counting that are missed by early checkpoints are successfully detected and corrected in later iterations, while more subtle attribute inconsistencies are eventually identified by the final Challenger. These qualitative examples reveal a clear co-evolutionary pattern in which an increasingly deceptive Inducer continuously exposes new weaknesses, and the Challenger gradually acquires stronger fine-grained hallucination detection and correction capabilities.

HalluMirror exhibits stronger fine-grained visual grounding than competing baselines. The benchmark examples in Figs. 9 and 10 further demonstrate that the capabilities acquired through self-play generalize beyond the constructed adversarial examples. Across diverse settings, including incorrect premises in FINER, object existence in POPE and RePOPE, object and relation understanding in CRPE, and visually challenging questions in HallusionBench, HalluMirror consistently produces answers that are better aligned with the actual visual evidence. In several cases, the base model and multiple training- or decoding-based baselines are distracted by plausible textual cues or coarse semantic patterns, whereas HalluMirror correctly distinguishes subtle mismatches in object existence, attribute, and relation. These observations suggest that HalluMirror improves not only hallucination resistance, but also the model's ability to produce semantically coherent responses that remain faithfully grounded in fine-grained visual content.

HalluMirror Verifier enables fine-grained hallucination verification for open-ended responses. As illustrated in Fig. 6, HalluMirror Verifier jointly considers the input image, question, and model-generated description to determine whether individual claims conflict with visual facts. Through semantic visual localization, it identifies subtle inconsistencies in open-ended responses and assigns them to the corresponding hallucination categories, including existence, attribute, and relation hallucinations. These examples demonstrate the verifier's ability to provide fine-grained and interpretable judgments for visually grounded response verification.

Remaining failures mainly arise when visual evidence is weak, deceptive, or conflicts with strong semantic priors. As shown in Fig. 11, HalluMirror still exhibits several characteristic failure modes. For instance, it may infer an unseen attribute when the relevant visual evidence is unavailable, misjudge subtle spatial relations, fall prey to visual illusions, or favor a commonsense configuration over what is actually observable in the image. The FREAK example is particularly indicative of prior-driven hallucination, where the model supplies plausible object configurations despite insufficient visual support, while the HallusionBench example highlights vulnerability to

misleading perceptual cues. These cases suggest that most hallucination samples used in the current training process do not explicitly cover scenarios where visual evidence strongly conflicts with language priors, limiting the model's ability to resolve such conflicts. A promising direction is therefore to incorporate learning from carefully controlled image perturbations or counterfactual visual edits, encouraging the model to explicitly contrast changed visual evidence and thereby strengthen the alignment between fine-grained visual facts and language priors.

E LIMITATIONS AND FUTURE WORK

HalluMirror currently focuses on three common types of multimodal hallucination, namely existence, attribute, and relation hallucinations. While these categories cover a broad range of fine-grained visual inconsistencies, they do not fully capture more complex hallucination phenomena, such as reasoning errors arising in long-context or multi-step multimodal tasks. Nevertheless, our results demonstrate that HalluMirror can effectively identify model-specific weaknesses through adversarial self-play and continuously transform these weaknesses into new training signals. Future work may therefore extend the current framework to broader hallucination settings, including more complex reasoning-oriented and long-horizon multimodal tasks.

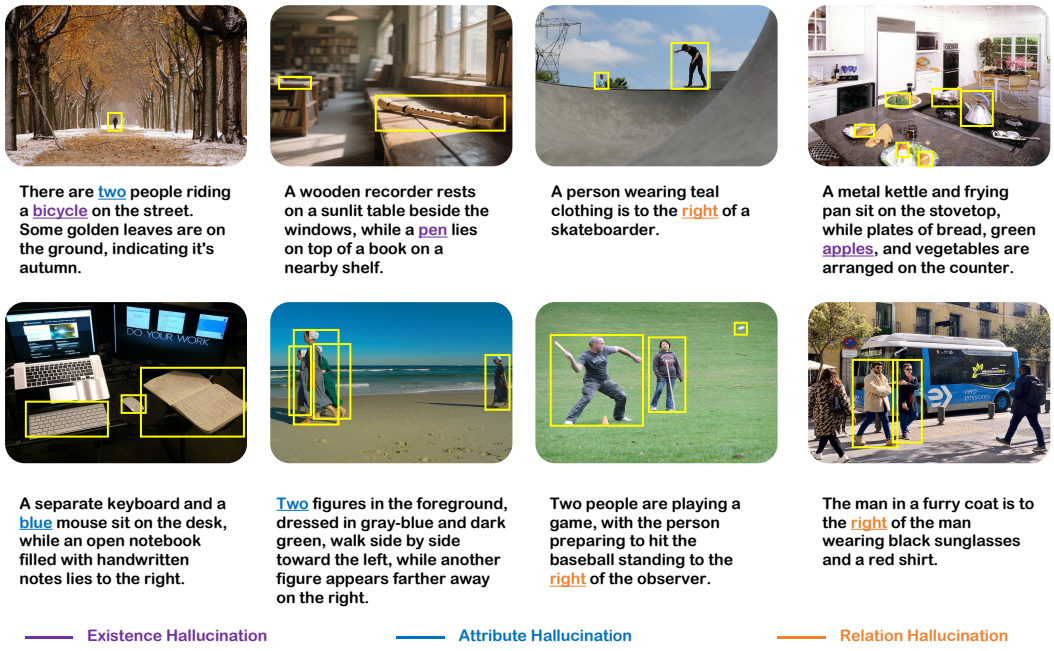

Figure 6: **Qualitative results of the HalluMirror Verifier on image captioning tasks.** The verifier decomposes each model-generated description into minimal visual claims and constructs positive grounding queries for semantic visual localization. The localized visual evidence is then used to identify hallucinations and determine their categories.

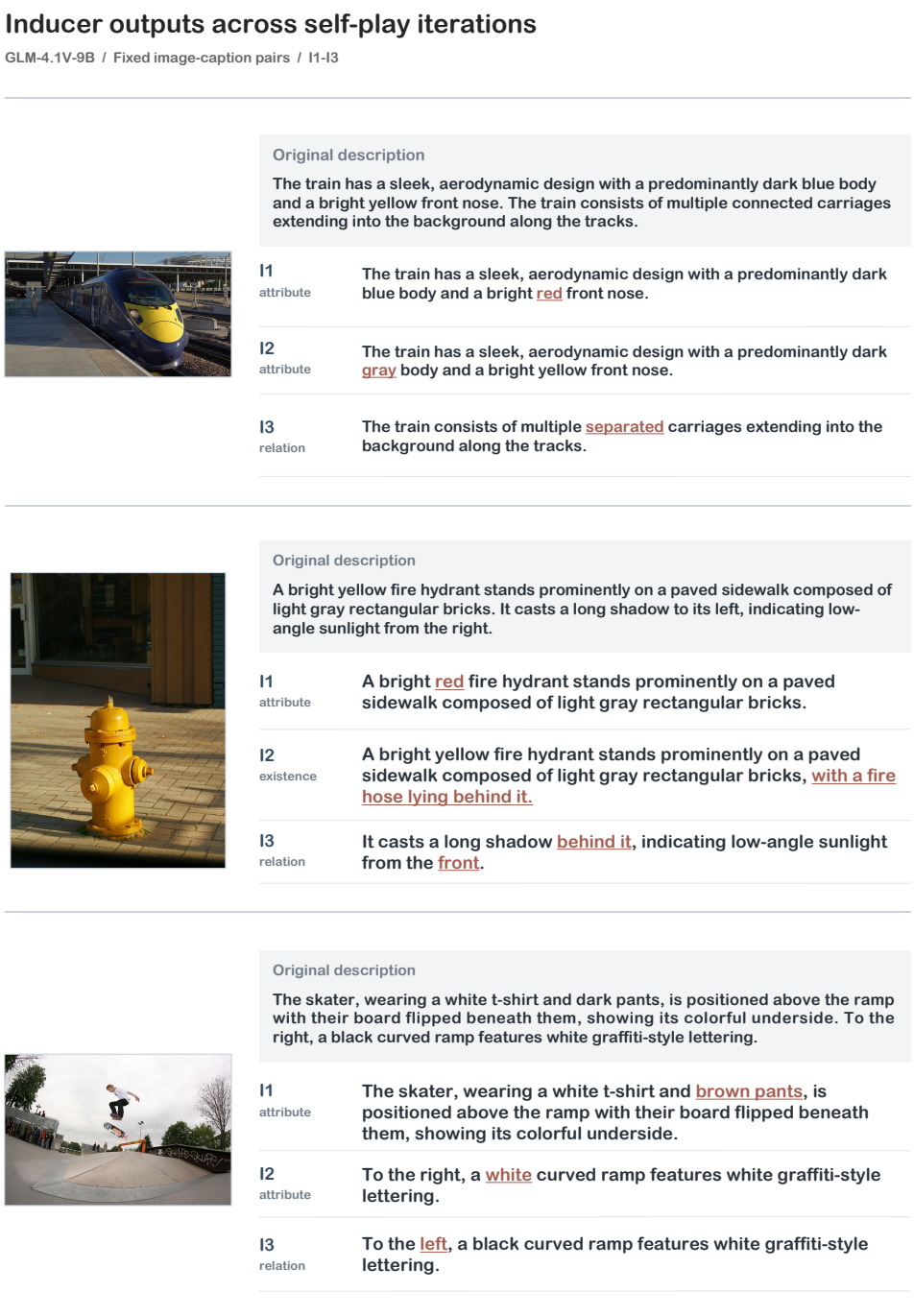

Figure 7: **Qualitative examples of Inducer evolution across self-play iterations.** Given fixed image-caption pairs, the Inducer progressively generates more subtle hallucinations involving attributes, object existence, and relations.

Challenger responses across self-play iterations

GLM-4.1V-9B / Detection and target-sentence repair / C0-C3

(a) Spatial relation / bus position



Injected sentence

This black-and-white photograph captures a street scene with a large bus parked on the right side of the frame, facing forward.

Model	Detected?	Output
C0	No	... a large bus parked on the right side of the frame, facing forward.
C1	No	... a large bus parked on the right side of the frame, facing forward.
C2	Yes	... a large bus parked on the left side of the frame, facing forward.
C3	Yes	... a large bus parked on the left side of the frame, facing forward.

(b) Counting / pieces of luggage



Injected sentence

The image shows four pieces of luggage placed on a concrete sidewalk.

Model	Detected?	Output
C0	No	The image shows four pieces of luggage placed on a concrete sidewalk.
C1	No	The image shows four pieces of luggage placed on a concrete sidewalk.
C2	Yes	The image shows two pieces of luggage placed on a concrete sidewalk.
C3	Yes	The image shows three pieces of luggage placed on a concrete sidewalk.

(c) Color / color of the clothes



Injected sentence

In the background, a grassy field stretches beyond a wire fence, where four blurred figures stand observing, with the nearer figure dressed in blue.

Model	Detected?	Output
C0	No	..., with the nearer figure dressed in blue.
C1	No	..., with the nearer figure dressed in blue.
C2	No	..., with the nearer figure dressed in blue.
C3	Yes	..., with the nearer figure dressed in pink.

Figure 8: **Qualitative examples of Challenger evolution across self-play iterations.** Later Challengers more reliably detect and correct fine-grained hallucinations.

Benchmark comparisons with multiple baselines

GLM-4.1V-9B / Complete choices and model answers

(a) FINER / Incorrect question premise



Question: What is parked on the road with pink and green dirt?

Answer choices

A. The road is not with pink and green dirt, but is with a paved surface.
B. The car with a destination sign indicating "CROSSTOWN", with graphic designs, and with number 2103.
C. The truck with a destination sign indicating "CROSSTOWN", with graphic designs, and with number 2103.
D. The boat with a destination sign indicating "CROSSTOWN", with graphic designs, and with number 2103.
E. The bus with a destination sign indicating "CROSSTOWN", with graphic designs, and with number 2103.

Correct answer A

Methods	Answer
Base, HALVA, HSA-DPO, OPA-DPO, RLAIF-V, VCD, SID, Self-Aug	E ✗
FINER	A ✓
HalluMirror (C3)	A ✓

(b) CRPE / Object existence



Question: Which of the following objects exists in the image?

Answer choices

A. Laptop.
B. Sky.
C. Light.
D. Tent.

Correct answer B

Methods	Answer
Base, FINER, HALVA, HSA-DPO, OPA-DPO, RLAIF-V, VCD, SID	C ✗
Self-Aug	B ✓
HalluMirror (C3)	B ✓

(c) HallusionBench / Character identification



Question: Is Jerry the Mouse in this figure?

Correct answer Yes

Methods	Answer
Base, FINER, HSA-DPO, OPA-DPO, RLAIF-V, VCD, SID	No ✗
HALVA, Self-Aug	Yes ✓
HalluMirror (C3)	Yes ✓

Figure 9: **Qualitative comparison with representative hallucination mitigation baselines.** HalluMirror produces responses that are more faithfully grounded in fine-grained visual evidence across diverse hallucination benchmarks.

Benchmark comparisons with multiple baselines

GLM-4.1V-9B / Complete choices and model answers

(d) RePOPE / Object existence



Question: Is there a person in the image?

Correct answer Yes

Methods	Answer
Base, FINER, HALVA, HSA-DPO, OPA-DPO, RLAIF-V, VCD, SID, Self-Aug	No ✗
HalluMirror (C3)	Yes ✓

(e) RePOPE / Object existence



Question: Is there a chair in the image?

Correct answer Yes

Methods	Answer
Base, FINER, HALVA, HSA-DPO, OPA-DPO, RLAIF-V, VCD, SID, Self-Aug	No ✗
HalluMirror (C3)	Yes ✓

(f) POPE / Object existence



Question: Is there a clock in the image?

Correct answer Yes

Methods	Answer
Base, FINER, HALVA, HSA-DPO, OPA-DPO, RLAIF-V, VCD, SID, Self-Aug	No ✗
HalluMirror (C3)	Yes ✓

(g) CRPE / Fine-grained relation



Question: What is on the gravel?

Answer choices

A. The railroad is on the gravel.
B. The tree is on the gravel.
C. The gravel is on the gravel.
D. The train is on the gravel.

Correct answer A

Methods	Answer
Base, HALVA, HSA-DPO, OPA-DPO, RLAIF-V, VCD, Self-Aug	D ✗
FINER, SID	A ✓
HalluMirror (C3)	A ✓

Figure 10: **Qualitative comparison with representative hallucination mitigation baselines.** HalluMirror produces responses that are more faithfully grounded in fine-grained visual evidence across diverse hallucination benchmarks.



Figure 11: **Representative failure cases of HalluMirror.** Remaining errors involve unseen attributes, subtle spatial relations, visual illusions, and conflicts between visual evidence and language priors.

Prompts for Hallucination Verification

Claim Decomposition and Grounding Query Planning:

Given a question and model response, decompose the response into minimal, self-contained visual claims and generate the positive LocateAnything queries required to verify them. Use the question to resolve short answers, negation, and coreference.

Hallucination categories: Existence concerns an incorrect claim about whether an object is present. Attribute concerns an incorrect property or quantity of an existing object. Relationship concerns an incorrect spatial relation, action, or interaction between objects.

Claim rules: Preserve the meaning and polarity of the response, and keep one verifiable predicate in each claim. Retain the subject identity and all participants required to interpret attributes and relationships. Split independent visual facts into separate claims, while excluding subjective, speculative, and visually unverifiable content.

Query rules: Generate only positive queries, including for facts denied by the response. Use a base-object query to verify the existence of each required subject. For quantity claims, query the relevant object or object subset without including number words. For attribute claims, query both the subject and the complete object–attribute phrase. For relationship claims, query the required participants and the complete relation phrase. Preserve necessary references so that evidence from different object instances is not mixed.

– –

Grounding-Evidence Interpretation:

Interpret the grounding results together with the original question and response. First determine the meaning, polarity, and hallucination category of each claim. Then evaluate only the queries belonging to that claim and respect their subject dependencies.

Base-object evidence determines whether the required subjects exist. Quantity evidence is obtained from the number of detected instances. Attribute evidence must support the stated property on the corresponding subject. Relationship evidence must support the complete relation involving the specified participants.

Decision rules: Apply existence verification before evaluating dependent attributes and relationships. If a required subject is absent, report only Existence Hallucination for that claim and skip its dependent checks. If the subject exists, report Attribute Hallucination when its stated quantity or property conflicts with the grounding evidence. Report Relationship Hallucination when all required participants exist but the stated relation is unsupported.

For negative claims, visual support for the denied fact indicates a hallucination, while absence of that fact supports the response. Attribute and Relationship Hallucination may both be reported through separate claims. Return one structured result per claim with a concise explanation based only on the decisive visual evidence.

Prompts for Hallucination Induction and Correction

Inducer: Hallucination Injection:

You inject natural visual hallucinations into clean image captions for training a vision-language hallucination challenger. The accompanying image is authoritative. The clean caption is only a reference and may omit true visual details. Do not assume that something is absent simply because it is not mentioned in the caption.

Core rules: Each edit must modify exactly one numbered source sentence, introduce exactly one atomic false fact, preserve all unrelated clean content, and remain fluent, plausible, visually checkable, and clearly false when checked against the image. Never reuse a sentence index or repeat the same `injected_text`. When several valid choices exist, prefer diverse edit types, entities, and mechanisms. Generate exactly {`injection_budget`} valid edits, but never invent an invalid edit merely to satisfy the budget.

Source-slot integrity: The `modified_sentence` must be a local revision of the exact text in its source slot. Preserve every clause, entity, predicate, and fact except for one changed factual span. Never copy, move, merge, summarize, or borrow content from another source slot. Minor grammar changes are allowed only when necessary for fluency.

Hallucination types: For *existence*, replace one visible object with one absent concrete object, or add exactly one absent concrete object that plausibly co-occurs with a visible object. For *attribute*, replace exactly one explicit attribute while keeping the object identity unchanged; valid targets include color, material, state, condition, orientation, pattern, shape, transparency, and exact quantity. For *relation*, replace a clear and objectively checkable spatial, containment, contact, action, role-binding, partner-binding, instance-binding, or orientation-target relation. Do not add objects, attributes, clauses, or sentences in attribute and relation edits.

Never target subjective or ambiguous details, scene-level appearance, OCR, logos, identities, world knowledge, intentions, emotions, causality, time order, object parts, vague counts, or uncertain image details. Return JSON edits only; do not return a final merged paragraph or explain your choices.

Challenger: Hallucination Detection and Repair:

You are playing the role of a hallucination challenger. The caption below may contain multiple subtle visual hallucinations. Hallucinations are visual claims that are not supported by the image. The main hallucination types are *existence*, *attribute*, and *relation*.

A hallucination may appear in two repair-relevant forms. For a replacement-style hallucination, repair the unsupported word or short phrase with the smallest possible edit. For an addition-style hallucination, remove only the unsupported word, phrase, clause, or extra sentence while preserving all clean information.

Task: For each numbered sentence, decide whether it contains any visual hallucination using exactly "yes" or "no", and provide a `corrected_sentence`. Copy clean sentences unchanged; minimally repair replacement-style hallucinations and remove only unsupported additions. Then rewrite the full caption as `corrected_description` so that it is faithful to the image while preserving all clean information.

Include exactly one judgement for every numbered sentence and use the original sentence indexes. The `corrected_description` must be complete and consistent with the corrected sentences in order. Do not delete an entire numbered sentence, and do not add explanations, markdown, or extra keys.