
UNI GAME: TURNING A UNIFIED MULTIMODAL MODEL INTO ITS OWN ADVERSARY

Zhaolong Su¹, Wang Lu², Hao Chen³, Sharon Li⁴, Jindong Wang¹

¹William & Mary ²Independent Researcher ³Carnegie Mellon University ⁴University of Wisconsin–Madison

{zsu05, jdw}@wm.edu, newlw230630@gmail.com, haoc3@andrew.cmu.edu, sharonli@cs.wisc.edu

ABSTRACT

Unified Multimodal Models (UMMs) have shown impressive performance in both understanding and generation with a single architecture. However, UMMs still exhibit a fundamental inconsistency: understanding favors compact embeddings, whereas generation favors reconstruction-rich representations. This structural trade-off produces misaligned decision boundaries, degraded cross-modal coherence, and heightened vulnerability under distributional and adversarial shifts. In this paper, we present **UniGame**, a self-adversarial post-training framework that directly targets the inconsistencies. By applying a lightweight perturber at the shared token interface, UniGame enables the generation branch to actively *seek and challenge* fragile understanding, turning the model itself into its own adversary. Experiments demonstrate that UniGame significantly improves the consistency (+4.6%). Moreover, it also achieves substantial improvements in understanding (+3.6%), generation (+0.02), out-of-distribution and adversarial robustness (+4.8% and +6.2% on NaturalBench and AdvQA). The framework is architecture-agnostic, introduces < 1% additional parameters, and is complementary to existing post-training methods. These results position adversarial self-play as a general and effective principle for enhancing the coherence, stability, and unified competence of future multimodal foundation models. The official code is available at: <https://github.com/AIFrontierLab/UniGame>.

1 Introduction

Unified Multimodal Models (UMMs) have recently demonstrated impressive capability in both visual understanding and image generation with a unified architecture [4, 38, 28, 42, 41, 40]. By jointly leveraging a language model backbone and a visual tokenizer–decoder stack [18, 2], these models promise a unified interface for cross-modal reasoning, grounded perception, and controllable generation. Specifically, the large-scale pre-training establishes general multimodal capabilities, and the post-training stage (supervised fine-tuning, SFT, Figure 2a) can further improve their performance on downstream tasks with enhanced reliability.

Despite their great performance, UMMs exhibit structural *inconsistency* between their understanding and generation pathways [43, 37]. This inconsistency stems from the inherently conflicting nature of the two objectives, leading to the mismatch in various aspects such as *semantics* (i.e., the model can answer a question correctly yet fail to generate a corresponding image, or vice versa [6, 31]), *capability* (e.g., generation is harder to improve than understanding, or vice versa [30]), and *feature compactness* (e.g., understanding requires more compact feature space while generation prefers oppositely). Inconsistency widely exists in real-world applications, where models frequently encounter unexpected inputs far from the training manifold, compositional combinations unseen during training, counterfactual queries, or modality conflicts such as distribution shift [6, 14, 24, 25] and adversarial attack [16]. If not sufficiently studied, it would greatly undermine multimodal information fusion, model robustness, and further performance improvement.

It remains challenging and unexplored to improve the consistency of UMMs, primarily due to the unclear learning objectives: only the consequences, but not the causes, are known. Therefore, recent post-training approaches tend to fill this gap using surrogate objectives. Reconstruction-based approaches (Figure 2b) regenerate the original images through the semantic embedding space derived from visual perception [37, 43]. They optimize a unified reconstruction

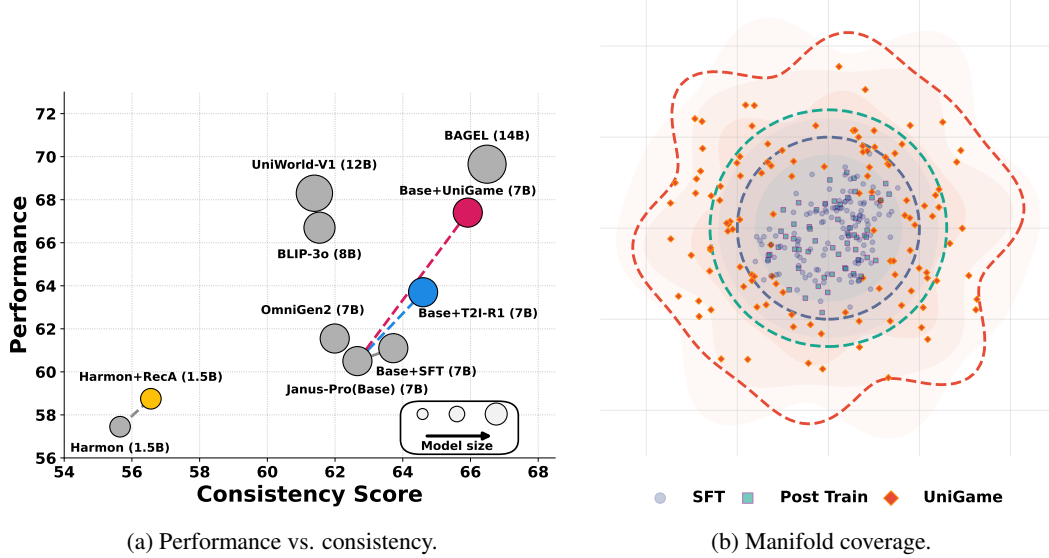


Figure 1: **Qualitative and quantitative analyses of UniGame.**¹ (a) The performance vs. consistency score of several models, indicating significant improvement of both metrics of our models. (b) The manifold produced by SFT, reconstruction-based Post Train, and UniGame. UniGame expands the training distribution toward hard yet realistic neighborhoods.

objective, which trains models within a closed auto-encoding loop. Reward-based methods (Figure 2c) typically optimize an ensembled reward function [13], combining task-specific or rule-based metrics to refine the output relying on external expert models. However, both are optimized using handcrafted objectives (reconstruction or reward), which only polish model behavior on a fixed training distribution and place no explicit constraints on the two coupling branches. As a result, they reproduce behaviors within a fixed manifold rather than expanding the shared generative space, leaving the inconsistency largely unresolved.

Can a UMM expose and correct its own inconsistencies from within? In this paper, motivated by the observation that adversarial signals reliably surface brittle reasoning in vision–language models [6, 14, 24, 31], we propose **UniGame** (Figure 3), the first self-adversarial post-training framework for UMMs. UniGame treats the generation pathway as an active adversary that searches for visually plausible, decoder-constrained perturbations that maximally challenge the understanding branch. It pushes the model beyond fixed data manifolds and produces structured adversarial samples along the uncertainty regions (Figure 1a). Concretely, UniGame installs a lightweight perturber at the shared visual-token interface to create bounded, structured perturbations. These perturbations are decoded into realistic adversarial images, filtered through a semantic consistency check, and stored in a hard-example buffer. The understanding branch is then optimized to correctly reason over both clean inputs and these internally generated, semantically aligned counterexamples. This forms a minimax self-play process where generation seeks to expose weaknesses while understanding learns from them, effectively expanding the shared generative manifold toward fragile yet meaningful regions (Figure 1b; theoretical insights in Appendix G.3). Compared to existing efforts, UniGame explicitly converts representational weaknesses into decoded, semantically coherent counterexamples that efficiently harden understanding (Figure 2). Empirically, UniGame uncovers richer, actionable failure modes (e.g., counting, fine attributes, and occlusion in Figure 5a) and improves the consistency, performance, and robustness.

This paper makes the following contributions:

1. **Novel framework.** We propose UniGame as the first framework to formalize UMMs post-training as a self-play game to improve the consistency between the understanding and generative pathways.
2. **Self-play training algorithm.** We instantiate UniGame by devising a flexible co-training algorithm that combines the perturber, regularizer, and hardness-aware mining modules. The algorithm is agnostic to both UMM architectures and post-training approaches.

¹(a) Where consistency Score computed as the average of WISE and UnifiedBench, performance is averaged over understanding bench MMMU and generation bench like GenEval. (b) We randomly sample 100 images, extract their unified embeddings, project to 2D with UMAP [22]; colored regions visualize each method’s *on-manifold* coverage.

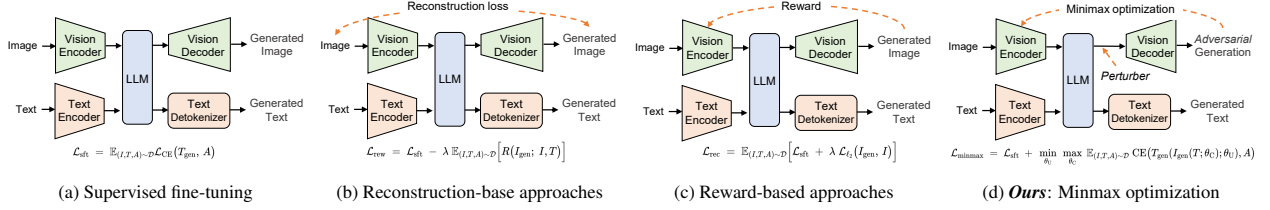


Figure 2: Illustration of four different post-training paradigms.

3. **Empirical improvement.** UniGame yields improvements in consistency (4.6%), understanding (+3.6%), generation (+0.02), OOD (+4.8%) and adversarial robustness (+6.2%).

2 Related Work

Unified Multimodal Models. UMMs aim to combine multimodal understanding and generation within a single backbone, enabling compact deployment and richer cross-modal reasoning [1, 38, 36, 3, 28, 4, 28]. Among these, BLIP3-o [1] explores hybrid autoregressive and diffusion training recipes to balance understanding and generation fidelity. Emu3 [38] treats images and text as an interleaved token stream and scales next-token prediction to unified multimodal outputs. TokenFlow [28] focuses on the tokenizer layer and introduces dual-granularity codebooks to reconcile the conflicting demands of discriminative understanding vs reconstructive generation. Despite architectural innovation, UMMs continue to face the inconsistency issue: the representation granularity and objective tension that underlie understanding and generation still cause ambiguous shared embeddings and latent failure modes that remain insufficiently addressed.

Post-training of UMMs. Aiming to resolve the inconsistency issue, existing post-training methods can be categorized into these types: (i) reconstruction and semantic-alignment losses to encourage fidelity [37, 43]; (ii) RL/reward-based optimization to directly optimize downstream metrics [32, 13]. Each family improves either fidelity or robustness [15, 2, 37, 6, 31, 32, 28]. For instance, RecA [37] leverages reconstruction alignment by conditioning generation on understanding embeddings and using reconstruction losses to bring representations closer, while T2I-R1 [13] enhances image generation through collaborative semantic- and token-level chain-of-thought combined with reinforcement learning. In the vision-language domain, AT has shown potential: VILLA introduces large-scale embedding-space perturbations across image and text modalities, improving robustness and generalization [6, 31]. Nevertheless, most methods either improve alignment or generation separately, and adversarial mechanisms are rarely incorporated into full UMMs. They commonly fail in one crucial respect: they do not exploit generation as an active adversarial process to strengthen the understanding branch. Specifically, adversarial or unguided embedding perturbations often produce off-manifold samples—unrealistic or semantically invalid artifacts, while reconstruction objectives do not intentionally surface decision-critical failure modes. Although RL-based schemes are effective, they are computationally costly and do not guarantee that these discovered examples remain decodable and semantically plausible [32, 13]. This leaves a key gap in truly improving the consistency of the understanding and generation pathways.

3 UniGame

3.1 Preliminary

Let \mathbf{x} denote an image, q a vision-grounded query, and a the ground-truth answer. After a frozen visual encoder and a projection layer, we write the unified visual tokenizer, which quantizes image representations into tokens aligned with the language model’s vocabulary embedding space as $\mathbf{z} = \text{Enc}(\mathbf{x}) \in \mathbb{R}^{N \times H}$, where N is the token length and H the hidden dimension.² Both visual and textual embeddings are then fed into a language model to learn high-level embeddings. The understanding branch U aims to minimize the discriminative loss of the textual output, and the generation branch G tends to reconstruct the input image:

$$\mathcal{L}_{\text{und}}(\theta_U) := -\mathbb{E}[\log p_U(a | \mathbf{z}, q)], \quad (1)$$

$$\mathcal{L}_{\text{gen}}(\theta_G) := \mathbb{E}[\ell_{\text{gen}}(G(\mathbf{z}), \mathbf{x})], \quad (2)$$

²In addition to encoder and projection layers, real encoder modules consists of other parts such as semantic encoder and tokenizer. We omit these details for simplicity.

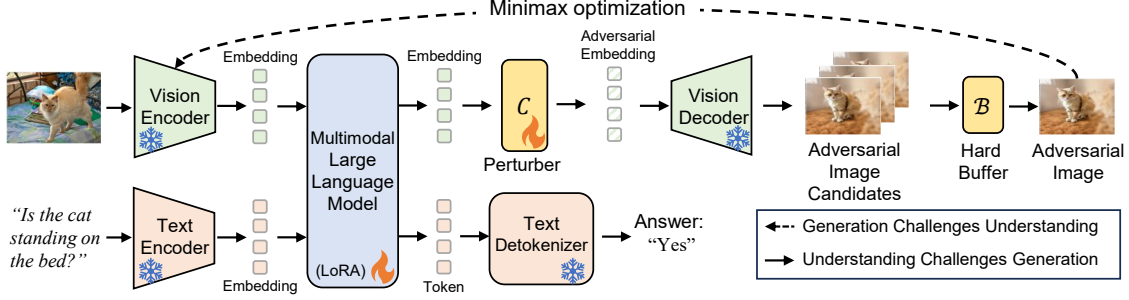


Figure 3: **Overview of UniGame.** This adversarial self-play improves understanding robustness and understanding-generation consistency. The perturber C is a lightweight (3-layer MLP) module and the hard buffer B is a filtering mechanism.

where $p_U(a \mid \mathbf{z}, q)$ is the predictive distribution of the understanding task and ℓ_{gen} is the reconstruction loss (e.g. MSE). These two objectives are commonly optimized jointly in a single multi-task objective:

$$\min_{\theta_U, \theta_G} \mathcal{L}_{\text{joint}} := \mathcal{L}_{\text{und}}(\theta_U) + \lambda \mathcal{L}_{\text{gen}}(\theta_G), \quad (3)$$

where λ is the trade-off hyperparameter.³

3.2 Motivation

UMMs inherently exhibit inconsistencies between the understanding and generation pathways due to their conflicting optimization requirements: the understanding branch favors task-oriented embeddings, but generation demands reconstruction-rich representations. Both branches operate on a shared generative manifold induced by encoding interface and decoder; any mismatch in how they carve up this manifold directly translates into structural inconsistencies. Improving consistency is challenging primarily owing to the lack of clear, direct learning objectives: We can only observe the consequences (e.g., semantic mismatches, capability gaps) but struggle to identify their underlying causes.

Existing efforts [37, 13] optimize for individual goals on a fixed set of data distributions, and place no explicit constraints on the two coupling training objectives. They encourage cooperative training, reproducing existing samples rather than expanding coverage of the shared manifold, where boundary behavior is most fragile, thus aggravating inconsistency. We argue that improving consistency requires *expanding* this shared manifold, especially around decision boundaries, instead of merely polishing the model within its comfort regions.

Considering the unified architecture of UMMs: can we improve the consistency *within* the model itself? We turn to adversarial training [21], which creates adversarial perturbations to explore understanding failures [6, 31]. This indicates that adversarial signals, if properly constrained, can serve as an effective mechanism for regularizing the decision boundaries of UMMs. Within the UMM architecture, we focus on converting generative priors into *decodable adversarial cases* that (i) remain semantically valid and (ii) reliably expose genuine reasoning failures in the understanding branch. This intuitively motivates the self-play training paradigm that makes best use of both of the understanding and generation branches as a minimax optimization framework. The generative pathway no longer passively follows alignment objectives: it is explicitly trained to produce realistic, on-manifold adversarial cases that challenge the understanding module, while the understanding branch is optimized to solve these internally generated challenges.

3.3 Overview of UniGame

As shown in Figure 3, the proposed **UniGame** introduces two lightweight, plug-in modules to a general UMM:

- **Perturber C :** A compact network with parameters θ_C ($|\theta_C| \ll \min(|\theta_U|, |\theta_G|)$) that maps the post-LM fused visual states $\hat{\mathbf{z}}$ to a perturbed token: $\tilde{\mathbf{z}} = C(\hat{\mathbf{z}}; \theta_C) = \hat{\mathbf{z}} + \boldsymbol{\delta}$, where $\|\boldsymbol{\delta}\| \leq \varepsilon_{\text{max}}$ is the budget to cap the perturbation magnitude for stabilization. We control the budget through an ablation study and use the best-performing setting in all main experiments. Details of the perturber architecture are in Appendix B.3.

³The term “jointly” here denotes simultaneous (multi-task) optimization of both branches—possibly sharing backbone parameters—rather than strictly sequential stage-wise training.

- **Hard-sample buffer \mathcal{B} :** A component that scores decoded candidates and stores hard, semantically plausible examples for replay via semantic-consistency check [30]:

$$\mathcal{B} = \{ G(\tilde{\mathbf{z}}) \mid H(\tilde{\mathbf{z}}) \geq \tau \}, \quad (4)$$

where $H = \text{CE}(p_U(\hat{a} \mid \text{Enc}(G(\tilde{\mathbf{z}})), q; \theta_U), a)$ is the cross-entropy loss and τ is the threshold.

We refer the original generation branch as the *clean* path, then the perturber introduces a *perturbed* path. In UniGame, a frozen visual encoder maps an input image to visual tokens at the unified interface. In the clean path, it is forwarded to the understanding head for supervised training with the query q ; In the perturbed path, the embeddings with perturbations generated by C are decoded by G into candidate images. The semantically consistent candidates are stored in the buffer \mathcal{B} . During training, the understanding module learns from hard examples from \mathcal{B} and clean samples, and the perturber C is optimized to generate challenging yet plausible cases. The vision encoder (SigLIP [46]) is frozen, and only LoRA [12] adapters on the LLM backbone and the Perturber C are trainable.

The overall training objective is a minimax game:

$$\min_{\theta_U} \max_{\theta_C} (\mathcal{L}_U(\theta_U) + \lambda \mathcal{L}_C(\theta_C; \theta_U)), \quad (5)$$

where $\lambda > 0$ controls the strength of the self-play signal between these two branches. We provide some **theoretical insights** to the convergence in Appendix G: Under bounded perturbation and decoder constraints, the perturber optimizes a lower bound of the worst-case understanding loss. This ensures the minimax dynamics remain stable and prevents off-manifold adversarial drift. We will elaborate on \mathcal{L}_U and \mathcal{L}_C in next section.

3.4 The Self-Play Training Process

The training objective in Eq. (5) consists primarily of two adversarial and iterative steps: to enable the understanding and generation branches to challenge each other. The complete training procedure is presented in Appendix A and we introduce only these two challenging steps.

Understanding Challenges Generation (the solid arrows in Figure 3). This is the naive feedforward path that optimizes the understanding module to prevent the generation branch from confusing it, i.e., to “challenge” the generation branch. The clean path forwards the original visual tokens \mathbf{z} directly to the understanding head U , producing the model’s nominal predictions and the standard supervised loss.

Here, the clean path preserves the original semantic information and simply computes the supervised loss; semantic plausibility and regularization are enforced on the generation side via the perturber’s norm and CLIP-based filtering from the hard-example mining buffer. Formally:

$$\begin{aligned} \mathcal{L}_U(\theta_U) = & \mathbb{E}_{\text{clean}} [\text{CE}(p_U(\hat{a} \mid \mathbf{z}, q; \theta_U), a)] \\ & + \beta \mathbb{E}_{\mathcal{B}} [\text{CE}(p_U(\hat{a} \mid \mathbf{z}, q; \theta_U), a)], \end{aligned} \quad (6)$$

where the first term keeps the model accurate on clean data, the second term forces U to correctly answer on current adversarial examples and mined hard cases, and $\beta > 0$ is a trade-off hyperparameter.

Generation Challenges Understanding (the dashed arrow in Figure 3). In this process, the perturbed embedding $\tilde{\mathbf{z}}$ will be rendered by the decoder G into image candidates: $\tilde{\mathbf{x}} = G(\tilde{\mathbf{z}})$, which are then subject to semantic-consistency checks (e.g., CLIP [30] similarity) and re-encoding/scoring by the understanding module. This path intentionally produces *on-manifold* adversarial examples to challenge the understanding branch, and hard candidates are stored in the buffer \mathcal{B} for replay. Formally, the perturber is updated to maximize the understanding loss:

$$\begin{aligned} \mathcal{L}_C(\theta_C; \theta_U) = & \mathbb{E}_{\text{clean}} \text{CE}(p_U(\hat{a} \mid \text{Enc}(G(C(\hat{\mathbf{z}}; \theta_C))), q; \theta_U), a) - \lambda \|\delta\|^2. \end{aligned} \quad (7)$$

3.5 Discussion

UniGame vs. GANs. UniGame is different from conventional GANs [9] that train a generator to fool a discriminator [9, 29]. First, GAN needs an extra discriminator for the adversarial game, while ours can operate *within* a UMM to leverage its own understanding and generation branches. Second, GANs primarily focus on generation tasks while ours targets both generation and understanding, involving more complex training and optimization process.

UniGame vs. Adversarial Training (AT). UniGame is the first attempt of applying AT [21, 10] to UMMs, but has the following differences. First, AT is mainly used for enhancing adversarial robustness, while ours explores AT for

Table 1: Consistency⁴ evaluation on UnifiedBench and WISE (\uparrow). Models marked with[†] denote our base model.

| Model | Params | UnifiedBench | WISE | Avg | Consistency Score |
|----------------------------|--------|--------------|-------------|--------------|----------------------------------|
| BAGEL [4] | 14B | 83.48 | 0.41 | 41.95 | 66.49 |
| UniWorld-V1 [19] | 12B | 78.99 | 0.35 | 39.67 | 61.39 |
| BLIP-3o [1] | 8B | 76.56 | 0.39 | 38.48 | 61.54 |
| OmniGen2 [41] | 7B | 83.31 | 0.30 | 41.81 | 61.99 |
| Janus-Pro [†] [3] | 7B | 82.77 | 0.35 | 41.54 | 63.66 |
| Harmon [40] | 1.5B | 65.41 | 0.41 | 32.90 | 55.65 |
| Show-o [42] | 1.3B | 69.16 | 0.30 | 34.73 | 53.50 |
| Janus-Pro+SFT [3] | 7B | 83.20 | 0.37 | 41.79 | 64.72 (+1.06) [‡] |
| Harmon+RecA [37] | 1.5B | 66.94 | 0.40 | 33.67 | 56.16 (+0.51) [‡] |
| Janus-Pro+UniGame | 7B | 85.20 | 0.43 | 42.82 | 68.32 (+4.66)[‡] |

consistency improvement. Second, UniGame differs fundamentally from prior AT by enforcing decoder-constrained, on-manifold image perturbations, enabling self-generated adversarial cases that remain semantically meaningful.

Extensibility. UniGame is agnostic to most UMM architectures and post-training approaches. First, since it is a general training framework that only introduces a lightweight trainable perturber, it is flexible to be integrated into most UMMs. Second, it does not conflict with existing methods, but can serve as their complement for further improvement on consistency and performance. We further explore the intergration of UniGame with emerging post-training method e.g., [13, 37], we train their post-trained model and demonstrate further improvements (see §4.6).

4 Experiments

In this section, we conducted extensive experiments to evaluate the proposed UniGame.

4.1 Experimental Setup

Tasks and datasets. We evaluated UniGame on popular benchmarks. Specifically, VQAv2 [11], MMMU [45], POPE [17], and MMBench [20] are adopted for understanding evaluation and GenEval [8] is employed for generation evaluation. For the evaluation of consistency, we report WISE score [23] and UnifiedBench [43]. For OOD robustness, we adopt NaturalBench [14], a challenging benchmark comprising real-world images captured in natural, uncontrolled environments (e.g., low lighting, occlusion, unusual viewpoints) that test robust visual reasoning. For adversarial robustness, we adopt AdVQA [16], an adversarially constructed VQA dataset where questions are intentionally designed to mislead models through linguistic ambiguity and visual distractors.

Implementation details. We implemented UniGame on a popular Janus-Pro-7B [3] UMM for main experiments, and further validated with two toy models that simulate distinct UMMs designs, similar to UAE [43]. The perturber is implemented as a 3-layer MLP operating on the shared visual-token space $\mathbb{R}^{N \times H}$, adding only 2.1M parameters and outputting token-wise perturbations followed by normalization and clipping. We adopt the open-source VQAv2 training set and CC3M [35]. More training details are in Appendix B.

Baselines. We compare UniGame to two categories of baselines: (1) **Different UMMs:** (i) *Auto-regressive models* [3, 36, 28, 7, 42], which unify understanding and generation through next-token prediction in a shared token space, with improvements in vision tokenizers; (ii) *Diffusion-based models* [1, 41], which leverage latent diffusion for generation while maintaining autoregressive understanding (e.g., BLIP-3o [1], OmniGen2 [41]); (iii) *Hybrid architectures* [19, 4], which employ specialized modules for different modalities (e.g., UniWorld-V1 [19], BAGEL [4]). (2) **Post-training methods:** (i) *Reconstruction-based alignment*, which uses caption-then-reconstruct cycles to enhance understanding-generation consistency (e.g., RecA [37]); (ii) *Reward-based approaches*, which use reward function to refine their outputs (e.g., [13]) Unlike these methods, UniGame introduces decoder-constrained self-adversarial training, converting latent inconsistencies into visually coherent counterexamples to improve reasoning robustness while preserving generation fidelity.

⁴Consistency Score = $0.6 \times \text{UnifiedBench} + 0.4 \times (\text{WISE} \times 100)$, jointly assessing self-consistency in understanding generated content (UnifiedBench) and prompt-image alignment (WISE), results marked with [‡] show improvement over base model.

Table 2: Results for understanding benchmarks (\uparrow).

| Model | Params | VQAv2 _{test} | MMMU | MMBench | POPE | Overall |
|----------------------------|--------|-----------------------|------|---------|-------------|-------------|
| TokenFlow-XL [28] | 14B | 77.6 | 43.2 | 76.8 | 87.8 | 71.3 |
| BAGEL [4] | 14B | — | 55.3 | 85.0 | — | — |
| UniWorld-V1 [19] | 12B | — | 58.6 | 83.5 | — | — |
| BLIP3-o [1] | 8B | 83.1 | 50.6 | 83.5 | — | — |
| Emu3 [38] | 8B | 75.1 | 31.6 | 58.5 | 85.2 | 62.6 |
| SEED-X [7] | 7B | 71.2 | 35.6 | 70.1 | 84.1 | 65.2 |
| Chameleon [36] | 7B | 66.0 | 22.4 | — | — | — |
| OminiGen2 [41] | 7B | — | 53.1 | 79.1 | — | — |
| Liquid [39] | 7B | 63.5 | — | — | 76.8 | — |
| Janus-Pro [†] [3] | 7B | 78.2 | 41.0 | 79.2 | 87.4 | 71.4 |
| Show-o [42] | 1.3B | 69.4 | 26.7 | — | 80.0 | — |
| SFT | 7B | 79.5 | 41.2 | 79.5 | 87.6 | 71.9 |
| RecA [37] | 1.5B | — | 35.7 | — | 83.9 | — |
| UAE [43] | — | — | — | — | — | — |
| Ours | 7B | 83.4 | 43.8 | 83.2 | 89.6 | 75.0 |

Table 3: Results of text-to-image generation on GenEval [8] (\uparrow).

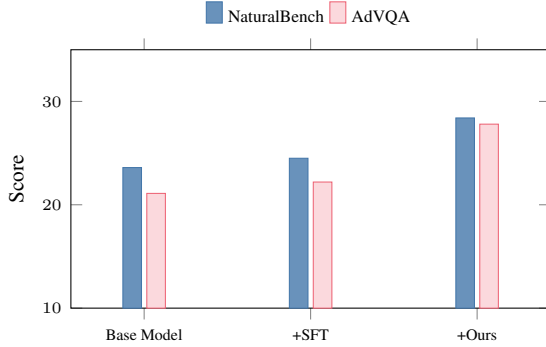
| Model | S. Obj. | Two Obj. | Counting | Colors | Position | Color Attri. | Overall |
|----------------------------|---------|----------|----------|--------|----------|--------------|---------|
| SEED-X [7] | 0.97 | 0.58 | 0.26 | 0.80 | 0.19 | 0.14 | 0.49 |
| Show-o [42] | 0.95 | 0.52 | 0.49 | 0.82 | 0.11 | 0.28 | 0.53 |
| D-DiT [18] | 0.97 | 0.80 | 0.54 | 0.76 | 0.32 | 0.50 | 0.65 |
| TokenFlow-XL [28] | 0.95 | 0.60 | 0.41 | 0.81 | 0.16 | 0.24 | 0.55 |
| Chameleon [36] | — | — | — | — | — | — | 0.39 |
| OminiGen2 [41] | 0.99 | 0.86 | 0.64 | 0.85 | 0.31 | 0.55 | 0.70 |
| Janus-Pro [†] [3] | 0.99 | 0.89 | 0.59 | 0.90 | 0.79 | 0.66 | 0.80 |
| SFT | 0.99 | 0.90 | 0.60 | 0.91 | 0.80 | 0.65 | 0.81 |
| UAE [43] | — | — | — | — | — | — | 0.86 |
| RecA [37] | 1.00 | 0.98 | 0.71 | 0.93 | 0.76 | 0.77 | 0.86 |
| Ours | 0.99 | 0.91 | 0.62 | 0.93 | 0.80 | 0.68 | 0.82 |

4.2 Consistency Evaluation

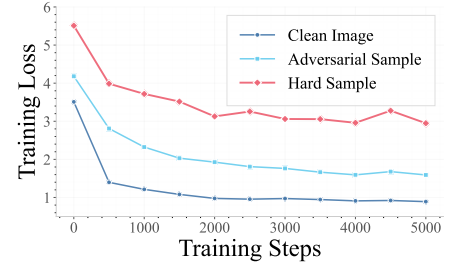
We evaluated consistency on two benchmarks: UnifiedBench [43] and WISE score [23]. UnifiedBench is a reconstruction-based benchmark tailored for UMMS, where the “caption–generate–compare” protocol measures how consistently information is preserved when images is converted into text and decoded back into images; the unified score is defined as the similarity between the ground truth and reconstructed images. WISE is a world-knowledge-informed text-to-image benchmark with 1000 knowledge-intensive prompts that are scored by a multimodal judge along consistency with the prompt, realism, and aesthetic quality; its overall Score emphasizes how faithfully the generated image reflects the textual description, making it a natural testbed for text-to-image consistency.

We followed Protocol 1 from [43] to evaluate the consistency of UniGame. Specifically, we randomly sampled 100 images from LAION-5B [34] as a test set. For each image, the model first generated a caption (understanding), based on which the model then reconstructed the image (generation). Finally, we computed the similarity scores between the original and reconstructed images using four vision-language backbones (CLIP [30], SigLIP [46], DINO-v2 [27], and DreamSim [5]).

As shown in Table 1, UniGame substantially improves the unification score across all metrics compared to both the Janus-Pro baseline and SFT-only training. This improvement demonstrates that self-play training not only enhances understanding, but also strengthens the bidirectional consistency between the understanding and generation branches, forcing the model to maintain semantic coherence across modalities and reducing the understanding-generation gap inherent in dual unified architectures.



(a) OOD and adv. robustness.



(b) Training Loss Comparison

Figure 4: (a) Robustness evaluation. (b) We observe that over 5K of training steps, the hard-sample loss persistently dominates that of Clean/Adversarial, suggesting UniGame continuously generates samples that are most challenging for the current model state.









Table 4: **Ablation: UniGame vs embedding perturbation.** All methods use the same perturbation budget ($\varepsilon_{\max}=0.02$), update schedule, and 16k training steps. We report VQAv2 accuracy (%).

| Method | Performance |
|--|-------------|
| Baseline (SFT) | 79.5 |
| <i>Embedding-only perturbation</i> | |
| Random noise in token space | 78.5 |
| Adversarial emb. | 78.9 |
| Adv. emb. + Cosine Similarity | 79.6 |
| Adv. emb. + Cosine + Buffer | 80.2 |
| <i>Decoder-constrained perturbation (Ours)</i> | |
| Decoding only | 81.5 |
| Decoding + Cosine Similarity | 82.2 |
| Decoding + CLIP | 82.7 |
| Full (+ CLIP + Buffer) | 83.4 |

4.3 Benchmark Results

UniGame improves both understanding and generation. Table 2 and Table 3 show the results on understanding and generation benchmarks, respectively. The results demonstrate significant improvements of UniGame over most competitors in both tasks. Specifically for understanding tasks, UniGame shows an average improvement of 3.1% over SFT, and 3.6% over the baseline model. UniGame also outperforms other larger models such as TokenFlow-XL, Emu3, and BLIP-3o, demonstrating that creating adversarial examples can serve as an effective augmentation approach. As for generation, UniGame outperforms most competitors such as TokenFlow-XL and OminiGen, achieving stronger performance than the base model and SFT. Note that UniGame performs slightly worse than UAE and RecA (0.82 vs. 0.86), which is mainly due to UAE and RecA’s explicit post-training on generation tasks, while ours was primarily trained on understanding tasks.

UniGame improves model robustness. We evaluated the OOD and adversarial robustness on NaturalBench [14] and AdvQA [16], respectively. Following Li et al. [14], we report *Group Accuracy (G-Acc)*, which awards one point only when a model correctly answers all four (image, question) pairs in a test sample. For AdvQA, we report standard accuracy as used in prior work. As shown in Figure 4a, UniGame exhibits significant improvement in OOD and adversarial benchmarks (4.8% and 6.2% gain, respectively), suggesting its strong performance in robustness, confirming UniGame effectively expand the decision boundaries, as in Figure 5a we explicitly probe four fine-grind visual reasoning cases, where base models fail but UniGame reasoned correctly. For space reasons, we present full experiments, additional ablations, and detailed OOD/adversarial robustness breakdowns in Appendix C and D.

| | | | |
|--|---|---|---|
| (C1) Are two dogs present in the image? Baseline: No (X) No (✓) Ours: Yes (✓) No (✓) | (C2) Is the monster truck jumping over vehicles? Baseline: Yes (X) Yes (✓) Ours: No (✓) Yes (✓) | (C3) What is the women doing in the image? Baseline: Standing (X) Standing (✓) Ours: Sitting (✓) Sitting (✓) | (C4) Is anyone holding an instrument? Baseline: No (X) No (✓) Ours: Yes (✓) No (✓) |
|  there is a pizza on a plate with a slice missing |  there are many vegetables on display at a farmers market |  there is a woman standing in a kitchen preparing food |  there is a cat sitting on a window sill looking out the window |
|  skateboarder in mid air doing a trick at a skate park |  there is a street sign that is on the corner of guadala jara |  there are two sheep wearing coats standing in a field |  a dog looking out of a car window in a rear view mirror |

(a) Case study for close-ended and open-ended understanding tasks.

| Baseline | UniGame | Baseline | UniGame | Baseline | UniGame | Baseline | UniGame |
|---|---|--|---|---|--|--|---|
|  |  |  |  | <p><i>Four red cubes stacked in a 2x2 square on the left, and four blue spheres on the right. One small green cube is placed between the cubes and spheres.</i></p> | <p><i>Three broccoli in a glass bowl on the left side of a table, two carrots lying on the side. A red sticker with the number "5" attached to the bowl.</i></p> | <p><i>Grand Canyon at sunrise, layered red rock formations, dramatic lighting, epic landscape.</i></p> | <p><i>A blue-eyed Siamese cat sitting on a green velvet armchair.</i></p> |

(b) Case study for generation tasks.

Figure 5: Qualitative case studies of UniGame understanding and generation.

4.4 Ablation Study

Table 4 shows the comparison between the embedding-only and decoder-constrained adversarial perturbations under matched settings. For embedding-only baselines, we apply perturbations directly in the visual token space without decoding, using cosine similarity constraints to prevent excessive token drift. The strongest embedding baseline incorporating adversarial perturbations, token-space cosine constraints, and buffer replay achieves 80.2% accuracy on VQAv2, representing a modest +0.7% improvement over the SFT baseline. In contrast, our decoder-constrained approach forces perturbations to pass through the model’s native decoder, rendering adversarial tokens into realistic images before evaluation. Notably, even *without* CLIP filtering, decoding alone improves accuracy to 81.5% (+2.0% over SFT and +1.3% over embedding perturbation), demonstrating that on-manifold constraints are inherently superior to token-space constraints. When we apply cosine similarity constraints in the decoded image feature space, performance further increases to 82.2%. Replacing feature-level cosine with CLIP’s text-image semantic matching yields 82.7%, validating that semantic constraints outperform purely geometric ones.

This ablation reveals three key insights: (i) the embedding-level perturbations can only leverage weak adversarial signals (+0.7%) because they operate in an abstract space disconnected from visual semantics; (ii) decoder constraints enforce on-manifold perturbations, yielding stronger adversarial training (+2.0%); (iii) using CLIP to maintain semantics further amplifies gains by ensuring adversarial samples remain semantically consistent with the query text. Together, these components establish a principled framework for self-adversarial training in UMMs.

4.5 Case Study

We provide case studies on both understanding and generation tasks for qualitative analysis.

Understanding tasks. Figure 5a illustrates four challenging categories of visual reasoning tasks: object counting, object interaction, spatial relation and location, and crowd object detection. UniGame outperformed the baseline models in all scenarios. For instance, in C4 (crowd object detection), dense and overlapping objects in crowded scenes challenge both localization and recognition. The baseline produces vague or incorrect answers, whereas UniGame maintains accuracy by learning from decoded adversarial samples that emphasize occlusion and clutter. These improvements align with our quantitative gains, confirming that UniGame systematically addresses decision-critical reasoning failures rather than merely fitting to benchmark statistics. And we evaluate the open-ended captioning task in Figure 5a, the qualitative examples align with our quantitative gains on benchmarks, and suggest that UniGame helps the model move toward semantically richer and accurate descriptions. More analysis is in Appendix E.1.

Table 5: UniGame can be plugged into an existing post-training pipeline with modest extra training to jointly improve understanding, generation, and unification. Starting from a RecA-trained model Harmon 1.5B, we further train with 5K UniGame steps (~ 10 GPU-h), yielding consistent improvements.

| Method | MMMU <i>understanding</i> | GenEval <i>generation</i> | UnifiedBench <i>consistency</i> |
|-------------|------------------------------|------------------------------|------------------------------------|
| RecA | 35.7 | 0.86 | 66.94 |
| RecA + Ours | 36.2 (+0.5) | 0.86 (−) | 68.21 (+1.27) |

Table 6: Extensibility analysis using two toy backbones.

| Model | Baseline | +UniGame | Trainable |
|--------------------|----------|---------------|------------------------------|
| UMM-1 (Qwen2.5-VL) | 60.4 | 66.4 (+6.0%) | $\sim 1.43\%$ (100.3M / 7B) |
| UMM-2 (GPT-OSS) | 28.9 | 53.2 (+24.3%) | $\sim 0.45\%$ (133.9M / 30B) |

Generation tasks. Figure 5b compares generations from the same prompts before and after post-training with UniGame. Overall speaking, UniGame helps UMMs to generate more faithful, accurate, and stylistic images. For instance, on the synthetic shapes example, the baseline model already produces plausible objects but often violates fine-grained layout constraints (e.g., incorrect left/right ordering or cube–sphere counts), whereas UniGame yields images that respect the specified 2×2 red cube stack, the correct number of blue spheres, and the spatial relations such as “on the left / on the right” and “between”. More explanations of other cases are in Appendix E.2. Together with the understanding cases, it suggests that UniGame enhances cross-modal consistency without sacrificing, and in some cases even improving the generation quality.

4.6 Extensibility and Efficiency

UniGame remains agnostic to UMM architectures and is computationally efficient compared to other post-training methods. In this section, we evaluate its generality and efficiency using the full set of VQAv2 on $2 \times H100$ (80 GB) with mixed precision. Unless noted otherwise, we use image generation size=384 and a global batch size of 8.

Extensibility. We first implement UniGame as a complement to RecA [37]. The results in Table 5 show consistent performance gains. Specifically, UniGame outperforms the original RecA by 0.5 on MMMU for understanding and 1.27 on UnifiedBench for consistency, while remaining the same on GenEval. These results indicate that UniGame can serve as a lightweight, plug-and-play post-training module that can be integrated into existing pipelines, requiring only minimal additional computation.

In addition to RecA, we further constructed two architectures: (1) *UMM-1* uses a Qwen2.5-VL [44] backbone with a SigLIP2 [46] understanding encoder and a Stable Diffusion-1.5 [33] image branch; (2) *UMM-2* keeps the vision/generation stack unchanged and replaces the backbone with GPT-OSS [26]. Since GPT-OSS is designed for texts, we inserted a trainable 2-layer MLP that projects vision embeddings into the language space. Table 6 demonstrates that UniGame is agnostic to model architectures and can improve the performance of all three different backbones. Moreover, the gains are achieved with fewer trainable parameters (e.g., $\sim 0.45\%$ for UMM-2 and $\sim 1.43\%$ for UMM-1), indicating its parameter-efficient generalization.

Efficiency. We further evaluate the efficiency of UniGame in comparison with RecA [37] and UAE [43] on MMMU. Table 7 shows that while achieving stronger performance, UniGame uses fewer trainable parameters, indicating its efficiency over existing post-training approaches.

4.7 Convergence and Hyperparameter Analysis

Finally, we present a large-scale analysis on the convergence and hyperparameters (e.g., the hard buffer threshold τ , trade-off β , and perturbation budget δ ; see Figure 11). To study the training dynamic, we also conduct extensive ablation study on the learning rates of *two major minmax opponents* in Figure 6b. Further, we systematically study the minimax dynamics by visualizing the optimization trajectory of each run. The best configuration yields a well-behaved minimax trajectory, where the two players alternate smoothly without divergence (see Appendix 9). We also probe the self-play dynamics between the two opponents and clearly observe the interaction: the two branches alternately dominate the training objective, exhibiting a stable tug-of-war behavior, and change of dominance, see Figure 10 and

Table 7: Efficiency study. Trainable parameter ratios are estimated from the official repository.

| Method | Baseline | +UniGame | Trainable |
|---------|----------|--------------|--------------------|
| ReCA | 34.7 | 35.7 (+1.0%) | ~91% (1.4B / 1.5B) |
| UAE | — | — | ~1% (0.1B / 11B) |
| UniGame | 41.0 | 43.8 (+2.8%) | ~1% (100.3M / 7B) |

12. More details in Appendix F demonstrate that UniGame offers a steady training process and stays relatively robust to different hyperparameter choices. Appendix G further presents some theoretical insights.

5 Conclusion and Limitation

UniGame is the first self-adversarial post-training framework to improve the consistency of UMMs. It formulated a minimax optimization game of the understanding and generation branches, thus enabling the model to autonomously discover its own failures. UniGame consistently showed increased consistency, performance, and robustness, highlighting the great potential of optimizing UMMs within the models for further improvement.

Limitations. This work has following limitations. First, we primarily evaluate Janus-Pro-7B; broader model coverage may reveal additional insights. Second, we use a limited set of datasets, and future work should test UniGame on more diverse and challenging benchmarks.

References

- [1] Jiu hai Chen, Zhiyang Xu, Xichen Pan, Yushi Hu, Can Qin, Tom Goldstein, Lifu Huang, Tianyi Zhou, Saining Xie, Silvio Savarese, et al. Blip3-o: A family of fully open unified multimodal models-architecture, training and dataset. *arXiv preprint arXiv:2505.09568*, 2025. 3, 6, 7
- [2] Xiaokang Chen. Janus: Decoupling visual encoding for unified multimodal understanding and generation, 2024. 1, 3
- [3] Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*, 2025. 3, 6, 7
- [4] Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, et al. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025. 1, 3, 6, 7
- [5] Stephanie Fu, Netanel Tamir, Shobhita Sundaram, Lucy Chai, Richard Zhang, Tali Dekel, and Phillip Isola. Dreamsim: Learning new dimensions of human visual similarity using synthetic data. *arXiv preprint arXiv:2306.09344*, 2023. 7
- [6] Zhe Gan. Large-scale adversarial training for vision-and-language representation learning. *arXiv preprint arXiv:2006.06195*, 2020. 1, 2, 3, 4
- [7] Yuying Ge, Sijie Zhao, Jinguo Zhu, Yixiao Ge, Kun Yi, Lin Song, Chen Li, Xiaohan Ding, and Ying Shan. Seed-x: Multimodal models with unified multi-granularity comprehension and generation. *arXiv preprint arXiv:2404.14396*, 2024. 6, 7
- [8] Dhruva Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36:52132–52152, 2023. 6, 7, 14
- [9] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014. 5
- [10] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 5
- [11] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 6, 14
- [12] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022. 5
- [13] Dongzhi Jiang, Ziyu Guo, Renrui Zhang, Zhuofan Zong, Hao Li, Le Zhuo, Shilin Yan, Pheng-Ann Heng, and Hongsheng Li. T2i-r1: Reinforcing image generation with collaborative semantic-level and token-level cot. *arXiv preprint arXiv:2505.00703*, 2025. 2, 3, 4, 6
- [14] Baiqi Li, Zhiqiu Lin, Wenxuan Peng, Jean de Dieu Nyandwi, Daniel Jiang, Zixian Ma, Simran Khanuja, Ranjay Krishna, Graham Neubig, and Deva Ramanan. Naturalbench: Evaluating vision-language models on natural adversarial samples. *Advances in Neural Information Processing Systems*, 37:17044–17068, 2024. 1, 2, 6, 8, 15

- [15] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 3
- [16] Linjie Li, Jie Lei, Zhe Gan, and Jingjing Liu. Adversarial vqa: A new benchmark for evaluating the robustness of vqa models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2042–2051, 2021. 1, 6, 8, 15
- [17] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*, 2023. 6, 14
- [18] Zijie Li, Henry Li, Yichun Shi, Amir Barati Farimani, Yuval Kluger, Linjie Yang, and Peng Wang. Dual diffusion for unified image generation and understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2779–2790, 2025. 1, 7
- [19] Bin Lin, Zongjian Li, Xinhua Cheng, Yuwei Niu, Yang Ye, Xianyi He, Shenghai Yuan, Wangbo Yu, Shaocong Wang, Yunyang Ge, et al. Uniworld: High-resolution semantic encoders for unified visual understanding and generation. *arXiv preprint arXiv:2506.03147*, 2025. 6, 7
- [20] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pages 216–233. Springer, 2024. 6, 14
- [21] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 4, 5
- [22] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018. 2
- [23] Yuwei Niu, Munan Ning, Mengren Zheng, Weiyang Jin, Bin Lin, Peng Jin, Jiaqi Liao, Chaoran Feng, Kunpeng Ning, Bin Zhu, et al. Wise: A world knowledge-informed semantic evaluation for text-to-image generation. *arXiv preprint arXiv:2503.07265*, 2025. 6, 7, 15
- [24] Changdae Oh, Zhen Fang, Shawn Im, Xuefeng Du, and Yixuan Li. Understanding multimodal llms under distribution shifts: An information-theoretic approach. In *International Conference on Machine Learning*, 2025. 1, 2
- [25] Changdae Oh, Jiatong Li, Shawn Im, and Sharon Li. Visual instruction bottleneck tuning. In *Advances in Neural Information Processing Systems*, 2025. 1
- [26] OpenAI. gpt-oss-120b & gpt-oss-20b model card, 2025. 10
- [27] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 7
- [28] Liao Qu, Huichao Zhang, Yiheng Liu, Xu Wang, Yi Jiang, Yiming Gao, Hu Ye, Daniel K Du, Zehuan Yuan, and Xinglong Wu. Tokenflow: Unified image tokenizer for multimodal understanding and generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2545–2555, 2025. 1, 3, 6, 7
- [29] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 5
- [30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. Pmlr, 2021. 1, 5, 7
- [31] Javad Rajabi. Token perturbation guidance for diffusion models, 2025. 1, 2, 3, 4
- [32] Shyam Sundhar Ramesh. Group robust preference optimization in reward-free rlhf. *arXiv preprint arXiv:2405.20304*, 2024. 3
- [33] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2022. 10
- [34] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35:25278–25294, 2022. 7
- [35] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of ACL*, 2018. 6, 14
- [36] Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024. 3, 6, 7
- [37] XuDong Wang. Reconstruction alignment improves unified multimodal models, 2024. 1, 3, 4, 6, 7, 10
- [38] Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yuezhe Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024. 1, 3, 7
- [39] Junfeng Wu, Yi Jiang, Chuofan Ma, Yuliang Liu, Hengshuang Zhao, Zehuan Yuan, Song Bai, and Xiang Bai. Liquid: Language models are scalable and unified multi-modal generators. *arXiv preprint arXiv:2412.04332*, 2024. 7

- [40] Sizhe Wu, Wenwei Zhang, Lumin Xu, Sheng Jin, Zhonghua Wu, Qingyi Tao, Wentao Liu, Wei Li, and Chen Change Loy. Harmonizing visual representations for unified multimodal understanding and generation. *arXiv preprint arXiv:2503.21979*, 2025. [1](#), [6](#)
- [41] Shitao Xiao, Yueze Wang, Junjie Zhou, Huaying Yuan, Xingrun Xing, Ruiran Yan, Chaofan Li, Shuting Wang, Tiejun Huang, and Zheng Liu. Omnigen: Unified image generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 13294–13304, 2025. [1](#), [6](#), [7](#)
- [42] Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv preprint arXiv:2408.12528*, 2024. [1](#), [6](#), [7](#)
- [43] Zhiyuan Yan, Kaiqing Lin, Zongjian Li, Junyan Ye, Hui Han, Zhendong Wang, Hao Liu, Bin Lin, Hao Li, Xue Xu, et al. Can understanding and generation truly benefit together—or just coexist? In *NeurIPS*, 2025. [1](#), [3](#), [6](#), [7](#), [10](#), [15](#)
- [44] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. [10](#)
- [45] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9556–9567, 2024. [6](#), [14](#)
- [46] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986, 2023. [5](#), [7](#), [10](#)

Appendix

| | | |
|---|---|----|
| A | Algorithm Details | 14 |
| B | Training Details | 14 |
| C | Detailed Experimental Results | 15 |
| D | Robustness Results | 16 |
| E | Details on Case Study | 16 |
| F | Convergence and Hyperparameter Analysis | 18 |
| G | Theoretical Insights | 19 |

A Algorithm Details

The complete training algorithm of UniGame is shown in Algorithm 1.

Algorithm 1 UniGame

```

1: Initialize  $\theta_U$  (understanding) and  $\theta_C$  (Perturber);
2: for each training step  $t = 1, 2, \dots$  do
3:   Sample minibatch  $\{(\mathbf{x}_i, q_i, a_i)\}_{i=1}^M \sim \mathcal{D}$  and encode  $\mathbf{z}_i = \text{Proj}(\text{Enc}(\mathbf{x}_i))$ 
4:   Challenge step (update  $C$ ):
5:   Compute perturbations  $\delta_i = C(\mathbf{z}_i; \theta_C)$  and perturbed tokens  $\tilde{\mathbf{z}}_i = \mathbf{z}_i + \delta_i$  with  $\|\delta_i\| \leq \varepsilon_{\max}$ 
6:   Decode candidates  $\tilde{\mathbf{x}}_i = G(\tilde{\mathbf{z}}_i)$ 
7:   Compute  $\mathcal{L}_C(\theta_C; \theta_U)$  as in Eq. (7)
8:   Update  $\theta_C \leftarrow \theta_C + \eta_C \nabla_{\theta_C} \mathcal{L}_C$ 
9:   if  $t \bmod m = 0$  then
10:    Compute scores  $H_j$  and keep candidates passing CLIP threshold  $\tau$  and push hard examples into  $\mathcal{B}$  via Eq. (4)
11:   end if
12:   Understand step (update  $U$ ):
13:   Construct mixed batch: clean samples  $(\mathbf{z}_i, q_i, a_i)$ , and hard samples  $(\hat{\mathbf{z}}_j, \hat{q}_j, \hat{a}_j)$  drawn from  $\mathcal{B}$ 
14:   Compute  $\mathcal{L}_U(\theta_U)$  on the mixed batch as in Eq. (6)
15:   Update  $\theta_U \leftarrow \theta_U - \eta_U \nabla_{\theta_U} \mathcal{L}_U$ 
16: end for
    
```

B Training Details

B.1 Training and Testing Data

Data volumes. Unless otherwise noted, we follow the official training/evaluation splits and report results on the standard benchmarks. Training uses **VQAv2 train-split** [11] is a large-scale visual question answering benchmark (hundreds of thousands of image–question pairs) collected from MS-COCO images with crowd-sourced free-form answers; it emphasizes grounded visual reasoning under natural images. **CC3M** [35] (training only) is a large web-scale image–caption corpus ($\sim 3\text{M}$ pairs in the full set); we use a filtered *subset* of 100k as text–image supervision for the generative branch.

Benchmarks. We briefly introduce the benchmarks:

- **VQAv2 test-dev** [11]: the official VQAv2 test-dev split contains **104 000** questions; evaluation is via the online server.⁵
- **MMMU** [45]: a college-level, multi-discipline benchmark with **11 500** questions in total (we report on the official test set).
- **POPE** [17]: object-hallucination evaluation with a balanced, image-grounded design; the **test split has 9000** QA pairs.
- **MMBench** [20]: curated multiple-choice suite; **dev 1164** and **test 1784** questions (4:6 split of $\sim 3\text{K}$).
- **GenEval** [8]: object/layout/attribute–focused T2I evaluation with **553** prompts (reference-free automatic checks).

⁵Counts from the official VQA site; see also recent reports confirming 104K for test-dev.

- **UnifiedBench** [43]: unification score via caption→reconstruction; Protocol-1 uses **100 source images**.
- **WISE** [23]: knowledge-informed T2I evaluation with **1000** structured prompts across 25 subdomains.
- **NaturalBench** [14]: vision-centric VQA with natural adversarial samples, $\sim 10\,000$ human-verified image-question pairs (2500 *groups* under the 2-image \times 2-question protocol), scored by G-Acc.
- **AdVQA** [16]: human-in-the-loop adversarial VQA; total size reported as $\sim 46\,807$ examples (commonly used splits include ~ 5123 val / $\sim 23\,399$ test).

B.2 Hyperparameter Details

Optimization details. UniGame is like the current UMMs post-training, is an end-to-end method and involves decoding images in each batch, to balance performance and cost. Our optimizations are as followed. We use AdamW optimizers with learning rates for Generation (`gen_lr`) and Understanding (`und_lr`). We conduct extensive ablation on the learning rate ratio between these two components (detailed in Appendix C and Table 8), ultimately finding that a ratio of approximately 250 achieves optimal performance (`gen_lr` = 5×10^{-3} , `und_lr` = 2×10^{-5}).

We implement mixed precision for training, given that Uni-Game only learned and uses small-norm perturbation, insufficient numerical precision can quantize away the perturbation’s gradients and wash out all the supervision. We vary the Generation and understanding update ratio in $\{1:1, 1:5, 1:10\}$. We performed a precision ablation comparing `fp16-all`, `bf16-all`, `tf32-enabled`, `fp32-all`, `fp16(G)+fp32(loss)`, and `bf16(G/D)+fp32(loss)`, and found that our final choice—computing the perturbation update, regularizer, and losses in `float32` while running the remaining forward/backward in `bfloat16`—consistently achieved the best stability–efficiency trade-off and the highest robust accuracy. We force all computations that determine the perturbation and its supervision to `float32`. Gradient norms and per-role clipping are also applied in `FP32`, and optimizer states remain `FP32` (AdamW default). All other forward/backward passes (vision tower, diffusion decoder, and LLM blocks) run under `bfloat16 autocast` for throughput. This preserves the perturbation signal while retaining the speed benefits of mixed precision.

B.3 Perturber

Network architecture of C . We implement the perturber C as a lightweight three-layer MLP that operates on each fused visual token after the language model. The first two layers have the same width as the UMM hidden size and apply non-linear transformations that refine the token representation and extract a direction in the shared visual-token space. The third layer acts as a direction head, mapping the hidden representation back to the token space and indicating along which semantic direction each token should be pushed to maximally challenge the understanding branch. In parallel, C maintains a single learnable scalar gate ε , shared across tokens and constrained within the perturbation budget $[0, \varepsilon_{\max}]$, which controls the overall perturbation strength. In this way, one part of C is responsible for discovering semantically adversarial directions, while the scalar gate ε controls how strongly these directions are applied, keeping the module compact (with $|\theta_C| \ll \min(|\theta_U|, |\theta_G|)$) yet able to generate small but semantically meaningful adversarial perturbations.

B.4 Hard Samples

UniGame added a hard sampler buffer to select only the challenging adversarial samples for training. Figure 6a shows some challenging examples in our experiments.

C Detailed Experimental Results

C.1 Learning Rate Ratio Ablation

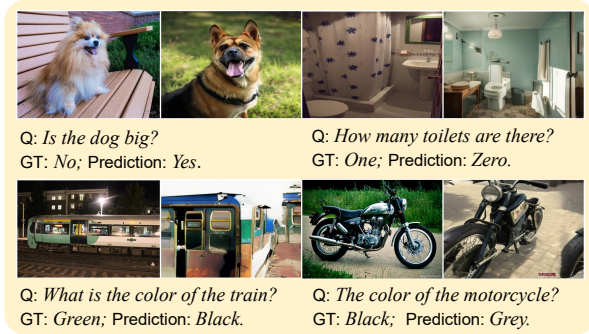
To determine the optimal balance between the generation and understanding branches, we conduct an extensive sweep of learning rate ratios. Table 8 lists the complete set of configurations tested.

C.2 Motivation Experiments

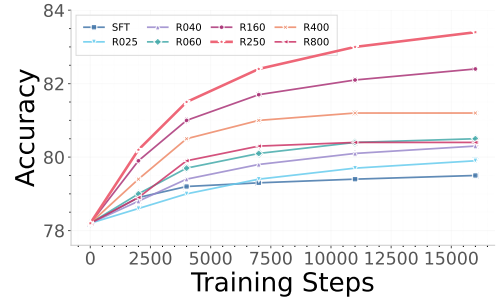
To find an Optimal noise level, we inject i.i.d. Gaussian noise into the projected visual tokens with $\sigma \in \{0, 0.005, 0.01, 0.015, 0.02, 0.05, 0.1\}$. We observe a *sweet spot* near $\sigma \approx 0.01$ where VQAv2 soft accuracy slightly increases (74.50 \rightarrow 75.58) before degrading at larger noise, see in Figure 7. This indicates that small, structured embedding perturbations can beneficially modulate the shared representation.

Table 8: Learning-rate configurations for the adversarial ratio sweep. Each row (ID Rxxx) specifies a pair of learning rates for the generation (gen_lr) and understanding module (und_lr); the last column reports their ratio Gen/Und . For example, R250 corresponds to $gen_lr = 5 \times 10^{-3}$ and $und_lr = 2 \times 10^{-5}$, i.e., a 250:1 ratio. These IDs (R025–R800) are used in Fig. 6b(b) to plot validation performance as a function of the adversarial ratio.

| ID | gen_lr | und_lr | Gen/Und |
|------|----------------------|----------------------|----------------|
| R025 | 1.6×10^{-3} | 6.3×10^{-5} | ≈ 25.4 |
| R040 | 2×10^{-3} | 5×10^{-5} | 40 |
| R060 | 2.4×10^{-3} | 4.1×10^{-5} | ≈ 58.5 |
| R100 | 3.2×10^{-3} | 3.2×10^{-5} | 100 |
| R160 | 4×10^{-3} | 2.5×10^{-5} | 160 |
| R250 | 5×10^{-3} | 2×10^{-5} | 250 |
| R400 | 6.3×10^{-3} | 1.6×10^{-5} | ≈ 394 |
| R600 | 7.7×10^{-3} | 1.3×10^{-5} | ≈ 592 |
| R800 | 8.9×10^{-3} | 1.1×10^{-5} | ≈ 809 |



(a) Cases drawn from the hard-sample buffer that successfully challenged the model.



(b) VQA accuracy across different adversarial ratios, with the best reaching 83.4%.

Figure 6: **Hard examples and learning-rate ratio ablation.** (a) Representative hard cases mined into the replay buffer. (b) Training dynamics under different adversarial update ratios.

D Robustness Results

The details results on OOD and adversarial robustness are shown in Table 9, indicating that UniGame significantly improves the robustness of the models.

E Details on Case Study

E.1 Case Study on Understanding Tasks

We offer more interpretations to Figure 5.

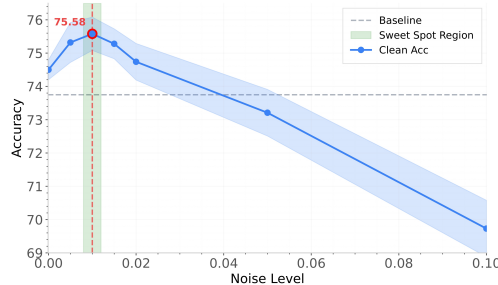


Figure 7: Perturbation Sweetspot

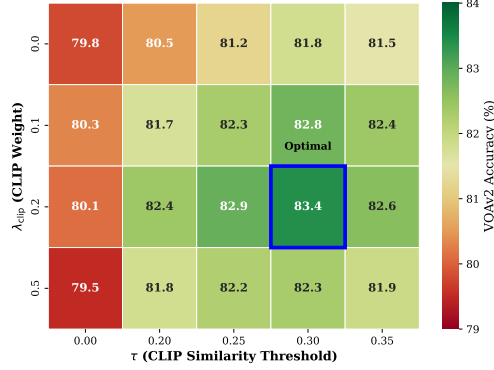


Figure 8: heatmap Ablation study on CLIP constraint configurations. We report VQAv2 accuracy for different combinations of CLIP weight and CLIP similarity threshold

Table 9: Results for OOD and adversarial robustness.

| Model | NaturalBench | AdVQA |
|-----------|--------------|-------|
| Janus-Pro | 73.8 | 34.2 |
| +SFT | 73.9 | 36.4 |
| +Ours | 78.6 | 40.4 |

- **Object counting (C1):** The baseline model fails to accurately count objects in cluttered scenes, often confusing similar-looking items or missing partially visible objects. After UniGame training, the model correctly identifies the precise count, demonstrating improved fine-grained visual attention.
- **Object interaction (C2):** Understanding relational semantics between objects (e.g., "person holding umbrella" vs. "umbrella next to person") requires compositional reasoning. The baseline misinterprets spatial relationships, while UniGame correctly recognizes the interaction pattern.
- **Spatial relation and location (C3):** Queries about relative positions (e.g., "left of", "behind") expose fragile spatial understanding in the baseline. UniGame’s adversarial training—which systematically perturbs spatial layouts during decoding—hardens the model against such failures.
- **Crowd object detection (C4):** dense and overlapping objects in crowded scenes challenge both localization and recognition. The baseline produces vague or incorrect answers, whereas UniGame maintains accuracy by learning from decoded adversarial samples that emphasize occlusion and clutter.

These qualitative improvements align with our quantitative gains, confirming that UniGame systematically addresses decision-critical reasoning failures rather than merely fitting to benchmark statistics.

In addition, we also present detailed analysis to the open-ended understanding tasks:

- **Open-ended understanding.** As illustrated in Figure 5a, UniGame produces more fine-grained and visually grounded captions than the baseline. The model not only recognizes the overall scene (e.g., pizza, street sign, animals) but also reliably captures details such as a missing pizza slice, vegetables on display at a farmers market, a cat sitting on a windowsill looking out the window, or two sheep wearing coats standing in a field. These examples show that adversarial self-play improves open-ended descriptions by encouraging the model to focus on decision-critical visual evidence rather than hallucinated or overly generic content.

E.2 Case Study on Generation Tasks

We offer more detailed explanation of the text-to-image generations in Figure 5b.

- On the synthetic shapes example, the baseline model already produces plausible objects but often violates fine-grained layout constraints (e.g., incorrect left/right ordering or cube–sphere counts), whereas UniGame yields images that respect the specified 2×2 red cube stack, the correct number of blue spheres, and the spatial relations such as "on the left / on the right" and "between".

- In the “broccoli in a glass bowl” example, UniGame more faithfully binds multiple attributes—three pieces of broccoli, two carrots on the side, and a clearly visible red sticker with the number “5” attached to the bowl—demonstrating stronger compositional control.
- For the Grand Canyon scene, the baseline sometimes collapses the layered rock formations into a flatter composition, while UniGame better preserves depth and lighting that match the prompt description.
- Finally, for the “blue-eyed Siamese cat sitting on a green velvet armchair”, UniGame produces a sharper Siamese appearance and a more coherent green velvet texture, indicating that self-play training can improve both semantic alignment and visual fidelity.

F Convergence and Hyperparameter Analysis

F.1 Convergence

Convergence of the minimax training. The minmax setup raises the practical question: when does the game converge and what schedules keep it stable? In our setup, only the Perturber C and LoRA adapters on the understanding branch U are trainable; due to U ’s larger capacity, it can dominate and degrade the generation module. We restore stability by giving C a higher learning rate and using short, interleaved updates. We conducted an extensive sweep of the Generation/Understanding update ratio in Table 8, shows $\text{gen_lr} = 5 \times 10^{-3}$, $\text{und_lr} = 2 \times 10^{-5}$, provides the best clean–robust trade-off; prolonged generation phases saturate the attack success rate (ASR) before U adapts and induce catch-up oscillations (see Figure 12 Figure 10). Conversely, when the generation overpowers U , decoded candidates drift off-manifold and hurt clean accuracy. Thus, *balance progression speeds*: (i) use a slightly larger learning rate for C than for U ’s adapters, and (ii) prefer short alternations over long unilateral bursts. Full grids, curves, and ablations are shown in Section C.

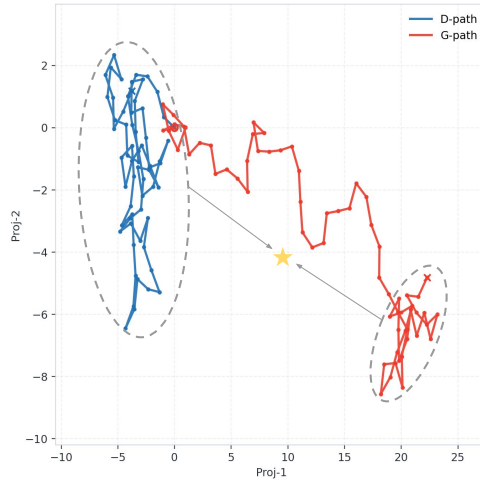


Figure 9: The best result of all of our runs, optimization path are projected to a two dimension axis.

Perturbation budget. The budget constraint ε_{\max} controls the perturbation magnitude in the token space. The results in Appendix C show a sweetspot that inverted U-shaped performance curve Figure 7, setting ε_{\max} too small (e.g., 0.005) produces weak perturbations that fail to expose critical reasoning failures, yielding limited robustness gains (+1.7% on NaturalBench).

F.2 Hyperparameter Sensitivity Analysis

Unless otherwise noted, we fix the perturbation budget to $\delta = \varepsilon_{\max} = 0.02$ in all main experiments, which we found to provide a good clean–robust trade-off after sweeping $\delta \in \{0.005, 0.01, 0.015, 0.02, 0.05, 0.10\}$ Figure 11. For hard-example mining, we define the hardness score H as the cross-entropy loss of the understanding branch on decoded candidates plus a CLIP-based hinge term, and select hard samples using a quantile-based threshold: the buffer threshold τ is set to the 60-th percentile of H within each mining batch, while additionally enforcing a minimum text–image CLIP similarity of 0.6 to filter out semantically off-manifold generations. The trade-off coefficient β in Eq. (6), which weights the contribution of buffer samples relative to clean examples, is set to $\beta = 0.5$ by default so that roughly

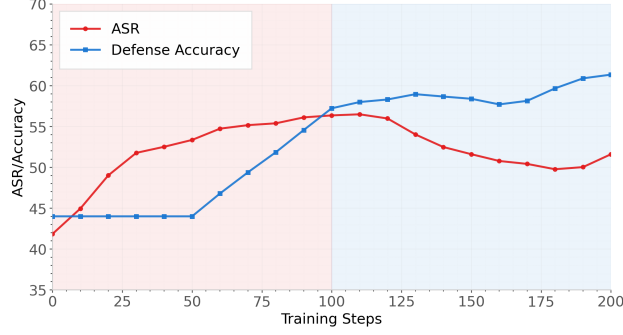


Figure 10: Self-play dynamics between the generation and the understanding . The two branches alternately dominate the training objective, exhibiting a stable tug-of-war behavior.

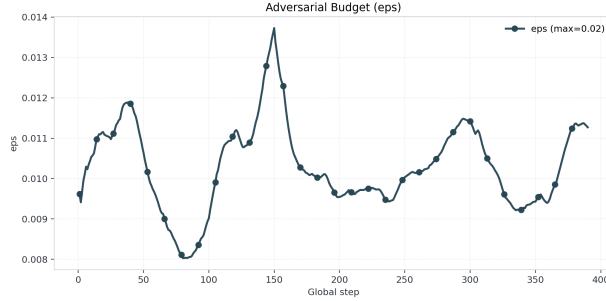


Figure 11: Perturbation Budget

half of the understanding gradient comes from adversarial or hard instances; we observed that UniGame is numerically stable for a broad range of $\beta \in [0.3, 1.0]$. The hard-sample replay buffer stores up to 50 decoded images ranked by H . We deliberately keep the capacity moderate, as substantially larger buffers (e.g., $\gg 10^4$ entries) would store many full-resolution decoded images and quickly lead to a steep increase in GPU and host memory usage, without providing noticeable additional benefits in practice.

G Theoretical Insights

In this section, we provide preliminary theoretical justification for why the proposed minimax self-play procedure improves (i) the stability of the understanding branch, (ii) convergence of the alternating optimization, and (iii) coverage of the shared generative manifold. The analysis is intentionally model-agnostic and applies to a broad class of unified multimodal architectures.

G.1 Convergence of the Minimax Self-Play Dynamics

Recall the UniGame objective

$$\min_{\theta_U} \max_{\theta_C} \mathcal{L}(\theta_U, \theta_C) = \mathbb{E}[\ell_U(\theta_U)] + \lambda \mathbb{E}[\ell_C(\theta_C; \theta_U)], \quad (8)$$

where the perturber maximizes the understanding loss while the understanding head minimizes both clean and adversarial losses, subject to a bounded perturbation $\|\delta\| \leq \varepsilon_{\max}$ in the shared token space. In this subsection, we analyze an *idealized* version of this minimax problem to provide theoretical intuition, rather than a full convergence proof for the actual deep network implementation.

Assumption 1 (Lipschitz continuity). The understanding loss $\ell_U(a \mid z, q)$ is L -Lipschitz continuous in the token embedding z and continuously differentiable in θ_U .

Assumption 2 (Bounded perturbation set and parameter domain). The perturber operates within a compact, convex set

$$\mathcal{D} = \{\delta : \|\delta\| \leq \varepsilon_{\max}\}. \quad (9)$$

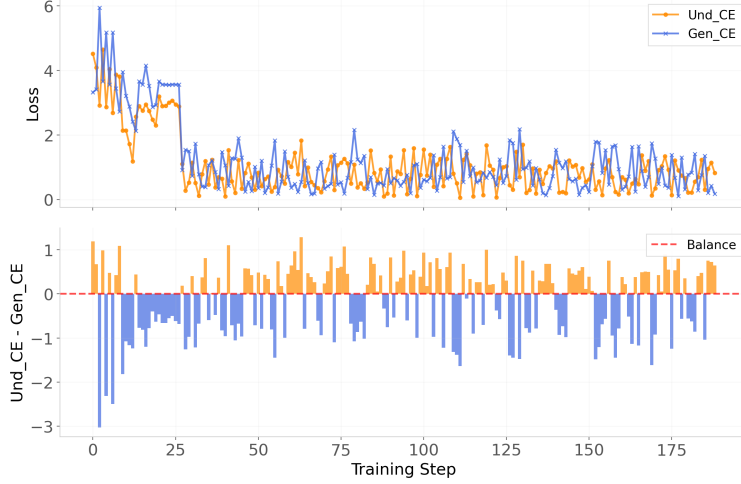


Figure 12: Dominance timeline. The trajectory alternates between understanding and generation phases, illustrating a stable tug-of-war rather than collapse to either side during training.

Moreover, the parameter sets Θ_U and Θ_C for θ_U and θ_C are assumed to be compact and convex.

Assumption 3 (Local nonconvex–concave structure). For any fixed $\theta_U \in \Theta_U$, the function $\theta_C \mapsto \mathcal{L}(\theta_U, \theta_C)$ is (locally) concave on Θ_C in a neighborhood of the stationary points of interest. Equivalently, the game is nonconvex in θ_U and (locally) concave in θ_C around those points.

Proposition 1 (First-order stationary point and stability). Under Assumptions 1–3, the minimax problem in Eq. (8) admits at least one first-order stationary point (θ_U^*, θ_C^*) , i.e.,

$$\nabla_{\theta_U} \mathcal{L}(\theta_U^*, \theta_C^*) = 0, \quad \nabla_{\theta_C} \mathcal{L}(\theta_U^*, \theta_C^*) = 0.$$

Moreover, for sufficiently small learning rates (η_U, η_C) , gradient descent–ascent generates a bounded sequence and converges to a neighborhood of a first-order stationary point of \mathcal{L} .

Sketch of proof. By Assumption 2, the feasible set in $(\theta_U, \theta_C, \delta)$ is compact and convex, so a minimax solution and hence a first-order stationary point exist. Assumption 1 guarantees that the loss is smooth in θ_U , and Assumption 3 provides a local nonconvex–concave structure: for each fixed θ_U , the objective is (locally) concave in θ_C . Under such smooth nonconvex–concave conditions, standard results for two-player minimax optimization show that gradient descent–ascent with sufficiently small step sizes (η_U, η_C) generates a bounded sequence and converges to an $\mathcal{O}(\eta_U + \eta_C)$ neighborhood of a first-order stationary point of \mathcal{L} .

Implication. These assumptions suggest that the adversarial self-play dynamics are stable and tend not to diverge, even though the perturber and understanding branches pursue opposing objectives.

G.2 Robustness Improvement via Worst-Case Regularization

For a fixed sample z from the shared representation space, the creator seeks a worst-case perturbation

$$\max_{\|\delta\| \leq \varepsilon_{\max}} \ell_U(z + \delta). \quad (10)$$

Using a first-order Taylor expansion around z , we obtain

$$\ell_U(z + \delta) \approx \ell_U(z) + \delta^\top \nabla_z \ell_U(z). \quad (11)$$

The optimal perturbation under the norm constraint is

$$\delta^* = \varepsilon_{\max} \frac{\nabla_z \ell_U(z)}{\|\nabla_z \ell_U(z)\|}. \quad (12)$$

Substituting δ^* into Eq. (11) and taking expectation over the data distribution yields the expected adversarial loss

$$\mathbb{E}[\ell_U(z) + \varepsilon_{\max} \|\nabla_z \ell_U(z)\|]. \quad (13)$$

Proposition 2 (Implicit gradient regularization). Adversarial self-play is equivalent, to first order, to adding a Jacobian-norm penalty:

$$\mathcal{L}_{U,\text{adv}} = \mathcal{L}_U + \lambda \varepsilon_{\max} \mathbb{E}[\|\nabla_z \ell_U(z)\|]. \quad (14)$$

Consequently, the understanding branch is encouraged to reduce its sensitivity to small perturbations in z , leading to locally flatter decision boundaries.

Implication. This explains the empirically observed improvements in robustness: the understanding head learns to be less sensitive to challenging input variations, improving both in-distribution and out-of-distribution performance as well as adversarial robustness.

G.3 Manifold-Expanding Effect of Decoder-Constrained Perturbations

Unlike conventional pixel-space adversarial training, UniGame produces *decoder-constrained* adversarial examples

$$\tilde{x} = G(z + \delta), \quad \tilde{x} \in \mathcal{M}, \quad (15)$$

where G is the decoder and \mathcal{M} is the decodable image manifold. This architecture ensures adversarial samples are:

1. **On-manifold:** \tilde{x} remains realistic and visually plausible;
2. **Semantically valid:** filtered by CLIP-based or similar consistency criteria;
3. **Near boundary regions:** targeted towards regions where the understanding model is fragile.

Assumption 3 (Local bi-Lipschitz decoder). The decoder G is locally bi-Lipschitz on the relevant region of the token space, i.e., there exist constants $0 < m \leq M < \infty$ such that for all z_1, z_2 in a neighborhood \mathcal{Z} ,

$$m\|z_1 - z_2\| \leq \|G(z_1) - G(z_2)\| \leq M\|z_1 - z_2\|. \quad (16)$$

Lemma 1 (Adversarial manifold expansion). Under Assumption 3, for any $z \in \mathcal{Z}$ the support of the perturbed output distribution satisfies

$$\text{supp}(G(z + \mathcal{D})) \supseteq \text{supp}(G(z)), \quad (17)$$

and expands the empirical training distribution toward regions where $\|\nabla_z \ell_U(z)\|$ is large.

Implication. The decoder-constrained perturbations induce a structured “inflation” of the data manifold towards decision boundary regions where the understanding head is uncertain. The hard-sample buffer \mathcal{B} collects such samples, which are approximately located near the understanding decision boundary. Training on \mathcal{B} reduces the empirical risk in these critical regions:

$$\hat{R}_{\text{adv}} = \frac{1}{|\mathcal{B}|} \sum_{x \in \mathcal{B}} \ell_U(x) \quad (18)$$

acts as a surrogate for minimizing the out-of-distribution risk R_{OOD} .

G.4 Summary of Theoretical Insights

The above analysis provides a theoretical lens on the benefits of the UniGame framework:

1. **Convergence of self-play:** Alternating gradient descent–ascent admits a stationary saddle point under mild smoothness and compactness assumptions.
2. **Robust optimization view:** The adversarial creator implicitly enforces a gradient-norm penalty (Eq. (14)), flattening the understanding decision boundary.
3. **Manifold expansion:** Decoder-constrained perturbations generate semantically valid hard samples that expand coverage of the decodable manifold towards challenging regions.
4. **Alignment with empirical gains:** These properties theoretically support the empirical improvements in understanding, consistency, out-of-distribution robustness, and adversarial robustness observed in our experiments.

WARNING: do not forget to delete the supplementary pages from your submission