DeCoRL: Decomposed Consistency Reinforcement Learning for Multi-Image Composition

Zhiqiang Wu* **Shuang Sun** **Jiale Zhang** **Jing Li** **Yibo Chen** **Xu Tang** **Yao Hu**
Xiaohongshu Inc.
{wuzhiqiang,sunshuang1,zhangjiale,lijing5}@xiaohongshu.com
{zhaohaibo, tangshen, xiahou}@xiaohongshu.com

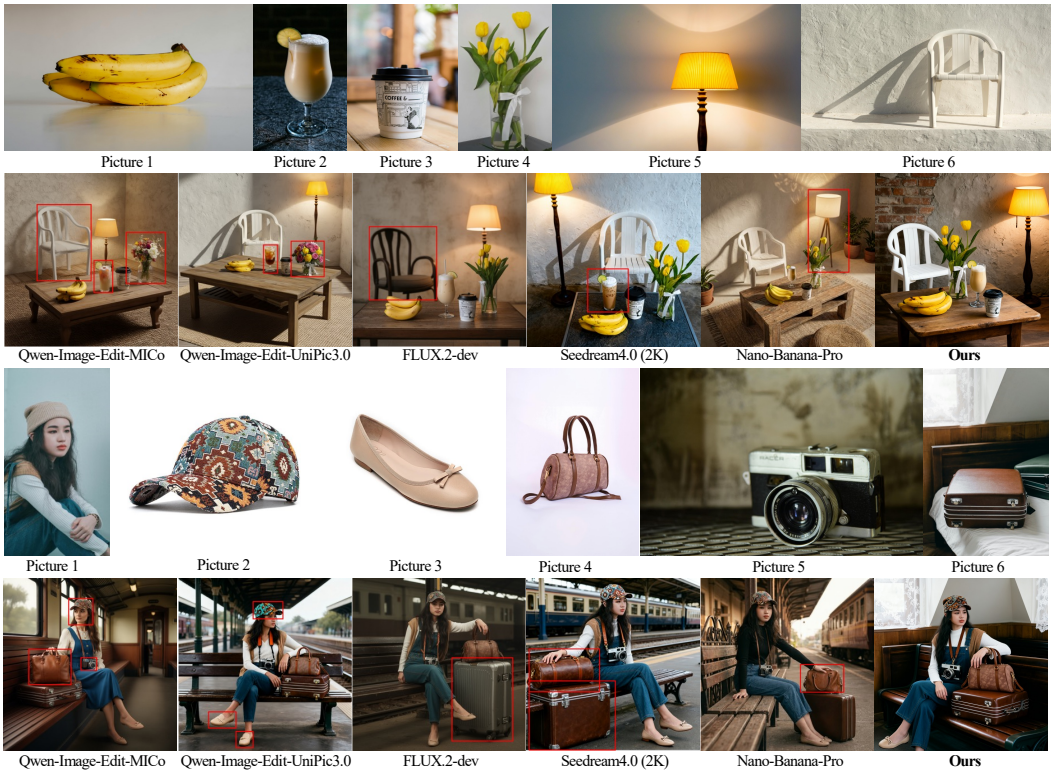

Figure 1: **Qualitative comparisons on two challenging MICo examples, each involving six reference images.** Compared with open-source and proprietary baselines, our method better preserves the identities and visual attributes of multiple referenced objects in the composed image. The red boxes highlight objects with inconsistent appearances.

Abstract

Multi-image composition (MICo) is a critical yet challenging capability for large-scale image editing models, where consistency degrades rapidly as the number of reference images increases. While supervised fine-tuning (SFT) can equip models with basic MICo ability, it provides only implicit supervision on the composed image and often fails to learn reliable object-level correspondence across multiple references. We propose DeCoRL, a reinforcement-learning (RL) framework that improves MICo consistency by decomposing holistic MICo consistency into

*Work done during internship at Xiaohongshu

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

object-level consistency. DeCoRL first establishes instruction-relevant object correspondences across reference images and localizes the corresponding regions in the generated image. It then uses a VLM-based reward model trained on single-object consistency annotations to produce fine-grained, object-wise feedback, which is aggregated into a multi-object consistency reward to guide RL post-training. We further introduce DeCo-Bench, a benchmark covering compositions from 2 to 6 reference images. Experiments on DeCo-Bench and the public MICo-Bench demonstrate consistent improvements in MICo consistency.

1 Introduction

Recent instruction-based editing models [3, 41, 48, 32, 39, 42, 22, 16, 17] have achieved strong performance in instruction following and image quality. However, most open-source systems [22, 41] are designed for single-image editing and cannot naturally support multi-image composition (MICo), where multiple reference images must be jointly composed into a single edited result. Although recent models such as Qwen-Image-Edit-2509/2511 [48] and FLUX.2-dev [21] begin to support this setting, they still struggle to maintain consistency as the number of reference images increases.

A central challenge in MICo is cross-image consistency: the model must integrate multiple visual conditions while preserving the correct identities, attributes, and semantic bindings of multiple entities in the generated image. This is fundamentally harder than single-image editing because the model must resolve many ambiguous many-to-one correspondences: each generated entity may inherit visual properties from a different reference image, and this ambiguity grows quickly with the number of inputs [57, 25, 56].

Despite this challenge, most existing MICo approaches primarily rely on large-scale (often synthetic) multi-image editing data and supervised fine-tuning (SFT) to endow models with multi-reference capability [47, 44]. However, SFT typically provides only implicit supervision on the generated image, without explicitly teaching object-level correspondence and binding across references, which often leads to identity confusion or attribute leakage when the number of conditions grows [57, 25, 56].

These limitations motivate reinforcement learning (RL) as a post-training paradigm for image editing, since it can directly optimize model preferences via reward signals rather than merely imitate paired targets as in SFT. Recent editing works have similarly adopted RL-style post-training to improve alignment and controllability beyond pure SFT [26, 54, 23, 34, 43, 14, 49, 18].

The effectiveness of RL hinges on a reliable reward. However, designing such a reward for MICo consistency is particularly challenging: as the number of reference images grows, holistic scoring must jointly resolve ambiguous correspondence and attribute-to-source binding under many competing visual conditions—an ill-posed judgment that remains unreliable even for strong VLM evaluators (e.g., GPT-4o) due to limited cross-image attention [47].

We propose **DeCoRL** (**De**composed **Co**nsistency **R**einforcement **L**earning), an RL framework built on a central methodological insight: holistic MICo consistency can be decomposed into tractable consistency checks over instruction-relevant, localizable objects. This decomposition is the core reward-design principle, while the specific evaluator, grounding model, and RL optimizer are implementation choices. We first train a single-object consistency reward model using a vision-language model (VLM) [37, 24, 1, 28], with chain-of-thought (CoT) [46, 53, 8, 52] supervision, and use its `yes`/`no` token probabilities [6, 30, 7] as a continuous reward signal. We then build an automatic object alignment pipeline: a VLM identifies instruction-relevant objects across the reference images, and Grounding DINO [31] localizes them. Based on these aligned objects, we define a *multi-object consistency reward* that matches generated objects with their corresponding references, scores each matched pair using *our trained single-object consistency reward model*, and penalizes missing objects (i.e., instruction-relevant entities not present in the generated output). This multi-object reward is then used to post-train the editing model via diffusion-based RL.

We further introduce **DeCo-Bench**, a dedicated benchmark for evaluating MICo consistency. By performing SFT followed by DeCoRL on the Qwen-Image-Edit2511 [48] model, we observe substantial and consistent improvements in MICo consistency on both DeCo-Bench and the public MICo-Bench, demonstrating the effectiveness of our method.

Our main contributions are summarized as follows:

- **Core insight.** We identify object-level decomposition as a practical reward-design principle for MICo: once instruction-relevant object correspondences are established, consistency can be evaluated locally instead of through a single holistic judgment.
- **DeCoRL framework.** We develop an RL post-training framework that aggregates object-level consistency scores into a multi-object reward, including penalties for missing objects. VLM parsing, Grounding DINO localization, and CoT-trained reward modeling provide one modular instantiation of this framework.
- **Benchmark and evaluation.** We introduce DeCo-Bench (2–6 reference images) and demonstrate consistent improvements on both DeCo-Bench and the independently developed MICo-Bench.

2 Related Work

Single-Image Editing. Instruction-based image editing has advanced rapidly with the development of text-to-image (T2I) foundation models [9, 41, 48, 20, 5, 4, 11, 10]. Representative open-source models (LongCat-Image-Edit [41] and FLUX.1-Kontext [22]) achieve strong performance in instruction following and image quality. However, these methods are primarily designed for single-image editing and do not support MICo.

Multi-Image Composition. Multi-image composition (MICo) requires jointly integrating several reference images into a single generated result, making it substantially more challenging than single-image editing. Recent open-source models such as Qwen-Image-Edit [48], FLUX.2-dev [21], and FireRed-Image-Edit [42] have begun to support this setting, while advanced closed-source systems such as Nano-Banana-Pro [13] and Seedream 4.0 [39] further highlight its practical importance. However, maintaining MICo consistency remains difficult, especially as the number of reference images increases.

Most recent MICo progress relies on large-scale (often synthetic) data construction followed by supervised fine-tuning (SFT), which can efficiently imitate demonstrations but typically supervises only the final composed image and provides limited explicit signals for learning cross-image correspondence and consistency relations. As a result, MICo consistency often degrades in harder cases with many reference images. Notably, this issue also appears in strong closed-source systems (e.g., Nano-Banana-Pro [13] and Seedream 4.0 [39]), which sometimes produce inconsistent compositions, e.g., object identities or attributes may change in complex multi-reference settings.

Reinforcement-Learning for Image Editing. Reinforcement learning (RL) has become an effective post-training tool for diffusion-based image editing, and has been increasingly adopted in recent open-source systems. For example, Skywork UniPic 2.0 explicitly incorporates online RL to improve unified multimodal generation and editing capability [45], while LongCat-Image-Edit [41], Qwen-Image-Edit [48], Edit-R1 [26], and FireRed-Image-Edit [42] also leverage preference optimization or reward-driven training (e.g., DPO [38], GRPO [40], Flow-GRPO [29], and Diffusion NFT [55]) to enhance image quality, instruction adherence, and controllability.

These successful practices suggest that RL provides a promising direction for improving editing models beyond standard SFT objectives. Motivated by this, we propose DeCoRL, which leads to substantial improvements in MICo consistency.

3 Method

We present **DeCoRL** (see Figure 2), an RL-based post-training framework for improving consistency in MICo. The key idea of DeCoRL is to avoid directly scoring MICo consistency between multiple reference images and a generated image, which is highly ambiguous and difficult to evaluate reliably (as stated in [47]), and instead introduce a principled decomposition into object-level consistency evaluation. This design is motivated by the observation that, once object correspondence is established, single-object consistency is more tractable to evaluate than global MICo consistency. The decomposed reward primarily targets the identity and visual attributes of localizable, instruction-relevant objects; the particular grounding and scoring components are modular.

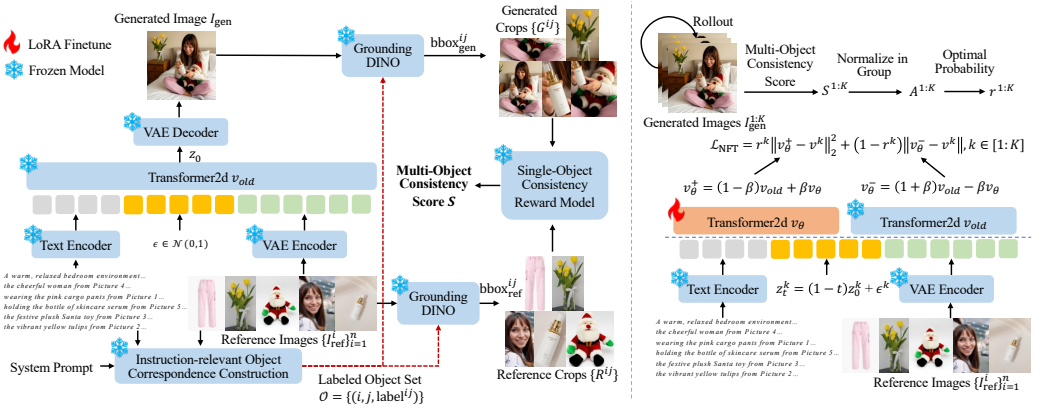

Figure 2: **Overview of DeCoRL.** DeCoRL improves MICo consistency via decomposed, object-level RL. We use a VLM to parse the instruction and identify instruction-relevant objects in each reference image, then localize them with Grounding DINO to obtain object crops. For a generated image, we detect the corresponding objects, score each reference–generated object pair with a trained single-object consistency reward model, aggregate the scores into a multi-object consistency reward, and finally optimize the transformer2d model with Diffusion NFT [55].

3.1 Task Definition

In MICo, the input consists of a textual editing instruction c and a set of reference images $\mathcal{I}_{\text{ref}} = \{I_{\text{ref}}^i\}_{i=1}^n$, and the goal is to generate an edited image I^{gen} that follows the instruction while preserving relevant consistency from the reference images.

3.2 Multi-Image Supervised Fine-Tuning

We adopt Qwen-Image-Edit2511 [48] as the base image editing model. Since the original model does not reliably support generation with more than three reference images, we first perform supervised fine-tuning (SFT) covering 1–6 reference images. For multi-image supervision, we combine MICo-150K [47] and UniPic 3.0 [44], yielding approximately 470K training examples across this reference range.

As the number of reference images increases, the number of visual tokens grows rapidly. To keep the total visual input within the model capacity, we resize all reference images under a shared total-pixel budget while preserving their aspect ratios. Let the i-th reference image I_{ref}^i have width w^i and height h^i. We first compute the total reference area $U = \sum_{i=1}^n w^i h^i$, and define a shared scaling factor $f = \sqrt{\frac{P}{U}}$, where P is the target total area budget (e.g., 1024^2). Each reference image is then resized as

$$\hat{I}_{\text{ref}}^i = \text{Resize}\big(I_{\text{ref}}^i,\ \text{round}(w^i f),\ \text{round}(h^i f)\big)\,, \tag{1}$$

so that aspect ratios and relative scales are preserved.

Latent Encoding. We encode the resized references and the target image I^{tgt} into latent features using a VAE encoder $\mathcal{E}$ [19]:

$$z_{\text{ref}}^i = \mathcal{E}\left(\hat{I}_{\text{ref}}^i\right), \qquad z_{\text{tgt}} = \mathcal{E}(I_{\text{tgt}})\,. \tag{2}$$

We then concatenate each reference latent feature into a whole latent feature:

$$z_{\text{ref}} = [z_{\text{ref}}^1; z_{\text{ref}}^2; \cdots; z_{\text{ref}}^n] \tag{3}$$

Training Objective. We train the transformer2d model with a flow-matching [27] objective. We first sample a scalar time step $t \in (0,1)$ from a logit-normal distribution [9]: $t = \sigma(u), u \sim \mathcal{N}(0,1)$ where $\sigma(\cdot)$ is the sigmoid function. We then sample Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$, with the same shape as z_{tgt}, and construct the noisy latent feature at timestep t:

$$z_t = (1-t)\, z_{\text{tgt}} + t\epsilon. \tag{4}$$

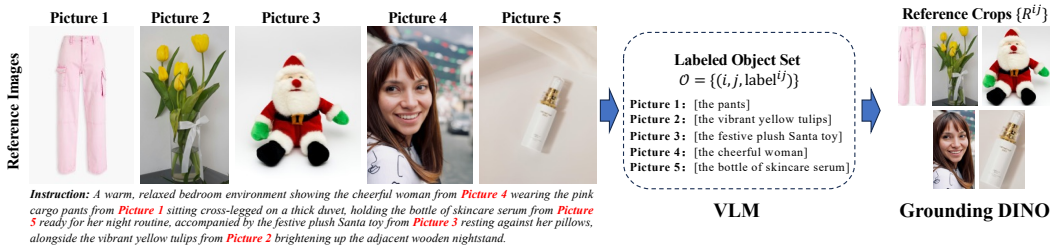

Figure 3: **Instruction-relevant Object Correspondences.** A VLM jointly interprets the instruction and multiple reference images, identifies instruction-relevant objects, and produces a labeled object set for Grounding DINO to obtain the reference crops.

The transformer2d model [9] predicts the velocity field for the target latent feature:

$$v_\theta = \text{Transformer2d}(z_t, z_{\text{ref}}, t, c). \tag{5}$$

Under the linear interpolation in Eq. (4), the ground-truth velocity is $v_t = \frac{dz_t}{dt} = \epsilon - z_{\text{tgt}}$. We optimize the flow-matching loss

$$\mathcal{L}_{\text{FM}} = \|v_\theta - v_t\|_2^2 . \tag{6}$$

Role of SFT. This stage provides a strong initialization for multi-reference editing, but it does not explicitly supervise object correspondences across reference images. Consequently, the editor may still omit requested objects, confuse their identities, or fail to preserve local attributes, motivating the object-level consistency rewards introduced next.

3.3 Single-Object Consistency Reward model

We train a single-object consistency reward model on PaCO-RL-Dataset [35] under a chain-of-thought (CoT) evaluation protocol, since `yes/no`-only supervision is low-entropy and prone to shortcut overfitting [46, 53, 8, 52]. We adopt Qwen3-VL-32B [2] as the reward backbone and fine-tune it to answer binary consistency questions.

Training. Each training example contains a reference crop $\mathbf{x}_{\text{ref}}$, a generated crop $\mathbf{x}_{\text{gen}}$, and a textual query q about their identity and visual-attribute consistency. We fine-tune the VLM with standard autoregressive cross-entropy on annotated CoT responses containing an explanation and a binary decision (`yes/no`).

Inference. We decode up to T_{check} steps and identify the first step $t^\star$ at which the highest-logit vocabulary token is `yes` or `no`. Denoting their logits by $\ell_{\texttt{yes}}^{(t^\star)}$ and $\ell_{\texttt{no}}^{(t^\star)}$, we compute the single-object consistency score as

$$s_{\text{obj}} = \begin{cases} \sigma\left(\ell_{\texttt{yes}}^{(t^\star)} - \ell_{\texttt{no}}^{(t^\star)}\right), & \text{if } t^\star \text{ exists}, \\ 0, & \text{otherwise}, \end{cases} \tag{7}$$

where $\sigma(\cdot)$ is the sigmoid function. The resulting score in $[0, 1]$ provides continuous object-level feedback.

3.4 Instruction-Relevant Object Correspondences

To make object-level rewards applicable to MICo, we first construct instruction-relevant object correspondences from the instruction and reference images, as illustrated in Figure 3. This step converts holistic MICo into a set of aligned object instances, enabling reliable evaluation with our trained single-object consistency reward model.

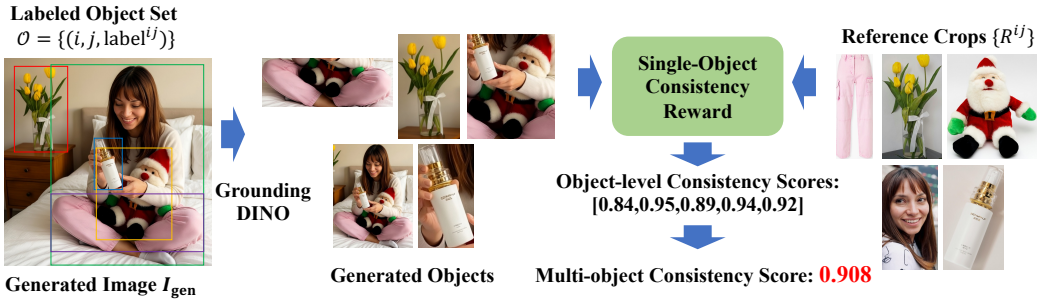

Figure 4: **Multi-object Consistency Reward.** Object-level consistency scores are computed for aligned reference-generated object pairs and then aggregated into a single-object consistency reward.

Instruction-aware Object Set. Given an instruction c and reference images $\mathcal{I}_{\text{ref}} = \{I^i_{\text{ref}}\}_{i=1}^n$, we use a VLM to identify the objects explicitly required by c and to assign an open-vocabulary label to each object instance in each reference image. We denote the resulting labeled object set as

$$\mathcal{O} = \{(i, j, \text{label}^{ij})\},$$

where (i, j) indexes the j-th instruction-relevant object in the i-th reference image. Focusing on $\mathcal{O}$ prevents background clutter or instruction-irrelevant entities from affecting reward computation.

Open-vocabulary Grounding in References. For each $(i, j, \text{label}^{ij}) \in \mathcal{O}$, we apply Grounding DINO on I^i_{ref} with the text prompt label^{ij} and obtain a set of candidate detections with confidence scores. We keep only the highest-confidence (top-1) detection (after standard post-processing such as thresholding and NMS), and denote its bounding box as $\text{bbox}^{ij}_{\text{ref}}$. The corresponding reference crop is

$$R^{ij} = \text{Crop}(I^i_{\text{ref}}, \text{bbox}^{ij}_{\text{ref}}). \tag{8}$$

These grounded crops serve as the spatial anchors for subsequent object-level consistency scoring.

3.5 Multi-Object Consistency Reward

We now construct a multi-object consistency reward for MICo by composing the single-object consistency scores, as shown in 4.

Generated-object Grounding and Matching. For each object label label^{ij} identified in the reference images, we also apply Grounding DINO on I_{gen} with the text prompt label^{ij} and keep the highest-confidence (top-1) detected bounding box $\text{bbox}^{ij}_{\text{gen}}$. The corresponding generated crop is

$$G^{ij} = \text{Crop}(I_{\text{gen}}, \text{bbox}^{ij}_{\text{gen}}). \tag{9}$$

Per-Object Reward. For each pair (R^{ij}, G^{ij}), we resize both crops while preserving aspect ratio and feed them into the single-object consistency reward model. If an object exists in the reference image but is not detected in the generated image, we assign a reward of zero to explicitly penalize missing objects:

$$s^{ij} = \begin{cases} s_{\text{obj}}\big(\text{Resize}(R^{ij}), \text{Resize}(G^{ij})\big), & G^{ij} \neq \emptyset, \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

Reward Aggregation. Let N denote the number of valid reference objects in the sample. We aggregate the object-level consistency scores into a multi-object consistency score:

$$S = \frac{1}{N} \sum_{i,j} s^{ij}, \tag{11}$$

where $N > 0$ denotes the number of valid instruction-relevant reference objects. The decomposition makes each required object's consistency explicitly measurable and assigns zero reward to missing

detections, rather than relying on a single undifferentiated whole-image judgment. The mean is deliberately simple; it does not by itself guarantee balanced improvement across objects.

3.6 Diffusion NFT Post-Training

Using the multi-object reward defined above, we further optimize the editing model through Diffusion NFT [55], a practical choice for diffusion RL post-training. The decomposition-based reward is not specific to this optimizer. The objective is to increase the likelihood of reward-improving predictions while keeping the updated policy close to a reference model for stability.

Reward Normalization and Policy Optimization. Following Diffusion NFT [55], for K samples generated under the same condition c, let S^k denote the multi-object reward of sample k, and let μ and σ be the group mean and standard deviation. We normalize and clip the relative advantages, mapping them to weights in $[0, 1]$:

$$r^k = \frac{1}{2} + \frac{1}{2A_{\max}} \operatorname{clip}\left(\frac{S^k - \mu}{\sigma + \epsilon}, -A_{\max}, A_{\max}\right), \tag{12}$$

where $\epsilon > 0$ ensures numerical stability and $A_{\max} > 0$ sets the clipping bound. These weights balance the positive and negative optimization directions:

$$\mathcal{L}_{\mathrm{NFT}} = \mathbb{E}_{k,t}\left[r^k \|v_\theta^+ - v^k\|_2^2 + (1 - r^k)\|v_\theta^- - v^k\|_2^2\right], \tag{13}$$

where $v^k = \epsilon^k - z^k_{\mathrm{gen}}$ with $z^k_{\mathrm{gen}} = \mathcal{E}(I^k_{\mathrm{gen}})$, and $v_\theta^+ = (1-\beta)v_{\mathrm{old}} + \beta v_\theta$ and $v_\theta^- = (1+\beta)v_{\mathrm{old}} - \beta v_\theta$ denote the positive and negative velocities, respectively. Here β is a hyperparameter, and the loss is evaluated over sampling timesteps t for samples generated by the old policy.

To prevent the policy from drifting too far from the initial model, we use reference regularization [33, 51] that penalizes differences between the current velocity prediction v_θ and the reference prediction v_{ref}. The final DeCoRL objective is

$$\mathcal{L} = \lambda_{\mathrm{NFT}}\mathcal{L}_{\mathrm{NFT}} + \lambda_{\mathrm{Reg}} \underbrace{\mathbb{E}_{k,t}\left[\|v_\theta - v_{\mathrm{ref}}\|_2^2\right]}_{\mathcal{L}_{\mathrm{Reg}}}, \tag{14}$$

where λ_{NFT} and λ_{Reg} are hyperparameters. In practice, to mitigate reward hacking, we additionally incorporate a general-purpose VLM (Qwen3.5-122B-10A) [36] reward that penalizes unnatural outputs, structural distortions, and compositing artifacts. This complementary signal also discourages globally implausible object–subject interactions that might be missed by local crop-based consistency scoring.

4 Experiments

We provide further implementation details, including our trained single-object consistency reward model, in Appendix A.1. Additional diagnostic evaluations and a discussion of scope are provided in Appendix B.

4.1 Training Setup.

SFT. We fine-tune Qwen-Image-Edit2511 on the multi-image datasets MICo-150K [47] and UniPic 3.0 [44] described in Section 3.2, together with Nano-consistent-150K [50] for single-image editing. We train for 24,000 steps on 32 H800 GPUs with a total batch size of 128, using AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, weight decay 10^{-2}) and a learning rate of 2×10^{-5}.

RL. We initialize from the SFT checkpoint and perform Diffusion NFT post-training on MICo-150K [47] using LoRA [15] ($r = 32$). Training uses 32 H800 GPUs, a group size of 16, and AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-10}$, weight decay 10^{-4}) with a learning rate of 3×10^{-4}.

4.2 Benchmarks

DeCo-Bench. To evaluate consistency in MICo, we construct **DeCo-Bench**, a benchmark covering MICo tasks with 2–6 reference images. Each setting contains 40 test cases (a total of 200 cases), and all reference images are real-world photographs. We provide details of DeCo-Bench in Appendix A.2.1.

Table 1: Results and ablations on DeCo-Bench.

Model	2 images↑	3 images↑	4 images↑	5 images↑	6 images↑	Overall↑
Qwen-Image-Edit-MICo [47]	6.90	5.27	4.85	4.79	4.06	5.17
Qwen-Image-Edit-UniPic3.0 [44]	7.82	6.23	5.73	5.24	5.11	6.02
FLUX.2-dev [21]	8.10	7.26	7.10	6.76	6.28	7.10
Seedream4.0 (2K) [39]	8.17	8.17	7.50	7.49	7.47	7.76
Nano-Banana-Pro [13]	**9.26**	**8.67**	**8.77**	**8.58**	**8.18**	**8.69**
Qwen-Image-Edit2511 (baseline) [48]	8.41	6.99	7.09	4.98	3.64	6.22
+ SFT	8.34	7.35	6.80	6.60	6.15	7.05
+ DeCoRL (ours)	<u>8.91</u>	<u>8.36</u>	<u>8.05</u>	<u>7.88</u>	<u>7.61</u>	<u>8.16</u>

Table 2: Results and ablations on MICo-Bench.

Model	Object Centric↑	Human Centric↑	HOI↑	De&Re↑	Overall↑
Qwen-Image-Edit-MICo [47]	52.38	21.11	34.95	37.42	35.86
Qwen-Image-Edit-UniPic3.0 [44]	**67.94**	44.79	28.24	<u>59.28</u>	47.90
FLUX.2-dev [21]	41.21	45.10	<u>49.86</u>	55.71	49.55
Nano-Banana [12]	48.01	41.79	49.64	49.44	47.83
Nano-Banana-Pro [13]	50.59	<u>54.75</u>	**50.21**	52.13	<u>51.76</u>
Qwen-Image-Edit2511 (baseline) [48]	27.41	28.89	19.59	20.60	22.90
+ SFT	39.60	30.49	32.41	45.50	37.58
+ DeCoRL (ours)	<u>53.47</u>	**56.21**	42.49	**64.42**	**54.03**

MICo-Bench. To further assess overall performance, we also evaluate our method on the public MICo-Bench. Following its standard protocol, we additionally report category-level results under Object Centric (138 cases), Human Centric (168 cases), HOI (291 cases), and De&Re (300 cases). We additionally report results on Semantic Consistency (SC), including Prompt Following (PF) and Subject Resemblance (SR), as well as Perceptual Quality (PQ), including Naturalness and Artifact. We provide details of MICo-Bench in Appendix A.2.2.

4.3 Comparative Results

Results on DeCo-Bench. As shown in Table 1, performance generally drops as the number of reference images increases (2→6), consistent with the increased difficulty of MICo consistency. Our method achieves an overall score of **8.16**, clearly outperforming all open-source baselines, including Qwen-Image-Edit-MICo (5.17), Qwen-Image-Edit-UniPic (6.02), and FLUX.2-dev (7.10), and also surpassing the proprietary Seedream4.0 (2K) (7.76). While Nano-Banana-Pro remains the strongest model overall (8.69), our results indicate that we are closing the gap toward it. Despite the remaining difference, the consistent improvements over all other models validate the effectiveness of our method.

Results on MICo-Bench. Table 2 reports results on the public MICo-Bench. Our method achieves a state-of-the-art overall score of **54.03**, outperforming both open-source baselines (Qwen-Image-Edit-MICo/UniPic3.0 and FLUX.2-dev) and strong proprietary systems (Nano-Banana and Nano-Banana-Pro).

4.4 Ablation Study

Ablation Study on DeCo-Bench. Table 1 shows that the baseline model struggles severely when the number of reference images increases, with a sharp drop at 5–6 images (4.98 and 3.64). In our tests, this often corresponds to qualitative "collapse" cases where the model fails to preserve key

Table 3: Ablations on four MICo-Bench metrics.

Model	Prompt Following (PF)↑	Subject Resemblance (SR)↑	Naturalness↑	Artifacts↑
Qwen-Image-Edit2511 (baseline)	6.1495	5.2378	5.0817	4.1208
+ SFT	**8.7586**	<u>7.0080</u>	<u>6.5355</u>	<u>5.7574</u>
+ DeCoRL	<u>8.6865</u>	**7.6393**	**8.0123**	**6.6362**

Table 4: Ablations of vanilla vs. trained VLM rewards on DeCo-Bench.

Model	2 images↑	3 images↑	4 images↑	5 images↑	6 images↑	Overall↑
Qwen-Image-Edit2511 (baseline) [48] + SFT	8.34	7.35	6.80	6.60	6.15	7.05
w/ **vanilla** VLM reward	8.54	7.49	6.42	6.90	6.78	7.21
w/ **trained** VLM reward (ours)	**8.91**	**8.36**	**8.05**	**7.88**	**7.61**	**8.16**

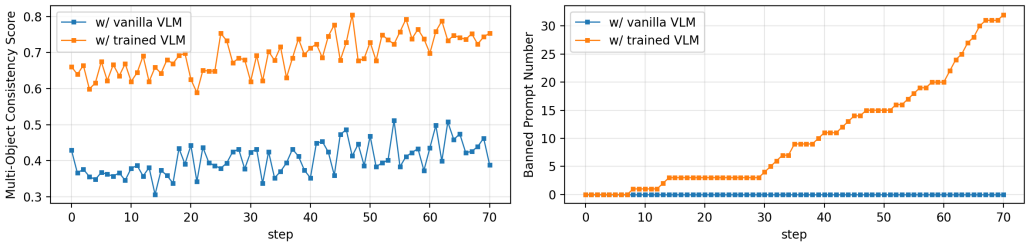

Figure 5: Training dynamics of multi-object consistency score and banned prompt count (vanilla vs. trained VLM).

attributes consistently. After SFT, consistency improves across all settings (overall: 6.22→7.05), indicating that supervised training helps the model acquire basic MICo capability.

Applying DeCoRL further yields consistent and substantial gains from 2 to 6 images (overall: 7.05→**8.16**), with especially notable improvements under harder conditions (e.g., 6 images: 6.15→**7.61**). These results suggest that while SFT can teach the model to perform MICo, it remains difficult for the model to robustly learn MICo consistency; our DeCoRL is effective in strengthening this consistency across varying numbers of inputs.

Ablation Study on MICo-Bench. As shown in Table 2, our method brings substantial improvements over both the baseline and baseline+SFT on MICo-Bench. Specifically, DeCoRL boosts the overall score from 22.90 (baseline) and 37.58 (+SFT) to **54.03**, with the most significant gain on Human Centric (28.89→30.49→**56.21**) and De&Re (20.60→45.50→**64.42**).

We further analyze MICo-Bench using four metrics in Table 3. Compared with the SFT model, DeCoRL yields clear gains in Subject Resemblance (SR) (7.01→**7.64**), Naturalness (6.54→**8.01**), and Artifacts (5.76→**6.64**), indicating better identity preservation and visual quality with fewer artifacts. Meanwhile, Prompt Following (PF) is slightly lower than the SFT variant (8.76→8.69), suggesting that the improvements are not merely due to reward hacking that over-optimizes a single criterion. Notably, SFT consistently improves all four metrics over the baseline, largely because it strengthens basic multi-image editing ability, reducing failure cases and improving instruction following in multi-image settings.

Ablation Study of Single-Object Consistency Reward Model on DeCo-Bench. We provide an ablation study of our single-object consistency reward model mentioned in Section 3.3, as presented in Table 4. Using a vanilla VLM yields only marginal gains over SFT (overall 7.05→7.21), suggesting limited ability to provide reliable supervision for MICo consistency. In contrast, our CoT-trained VLM substantially improves performance across 2–6 images, reaching **8.16** overall.

Figure 5 further supports this finding. With the vanilla VLM reward, the multi-object consistency score remains mostly below 0.5 with limited improvement, indicating a weak optimization signal. In contrast, our CoT-trained VLM reward stays above 0.6 and steadily increases during training. The banned prompt count is also informative: a prompt is banned when its samples already obtain high and similar rewards, meaning the model has learned this reward signal. It remains zero for the vanilla VLM, while it increases over time for the CoT-trained VLM, showing that more prompts receive stable and well-learned reward signals. Overall, these results show that the CoT-trained VLM provides a more reliable and informative reward than a vanilla VLM, enabling effective DeCoRL optimization and improving MICo consistency.

4.5 Visualization

Figure 1 presents two examples of MICo, each consisting of six images. It can be observed that both the open-source model and the closed-source Seedream 4.0 exhibit relatively poor consistency. Even the strongest closed-source model, Nano-Banana-Pro, shows objects with noticeable inconsistency under multi-reference settings. In contrast, our model maintains better consistency in both cases. We also provide more visualizations of our method on DeCo-Bench and MICo-Bench in Appendix A.3.

5 Conclusion

5.1 Summary

In this work, we introduce DeCoRL, a reinforcement-learning (RL) post-training framework that improves multi-image composition (MICo) consistency through object-level reward decomposition. DeCoRL establishes instruction-relevant object correspondences, localizes matched regions, and aggregates fine-grained single-object consistency scores into a multi-object reward for optimization. We further propose DeCo-Bench, covering MICo tasks with 2–6 reference images. Experiments on DeCo-Bench and MICo-Bench demonstrate consistent improvements, particularly in challenging many-reference settings, showing the effectiveness of decomposed consistency rewards for scalable MICo post-training.

5.2 Discussion and Future Directions

From many references to object-wise evaluation. DeCoRL is built on the principle that a difficult holistic MICo consistency judgment can be decomposed into more tractable checks. Instead of evaluating all references against the generated image jointly, we first establish instruction-relevant object correspondences, then score each matched reference–generated object pair individually, and finally aggregate these scores into a multi-object reward. This turns a many-reference judgment into a collection of simpler object-level evaluations while retaining a joint objective of preserving multiple subjects. The central methodological contribution is this decomposition-based reward design, rather than a particular VLM, grounding model, or RL optimizer.

Generality and future directions. Our CoT-trained VLM is one instantiation of the single-object evaluator. In principle, any reliable evaluator that produces comparable object-level consistency scores can be integrated into the same framework; the reward design is also conceptually independent of Diffusion NFT. This modularity suggests opportunities for other evaluators and editing backbones, although such extensions require empirical validation. The current method targets localizable objects and depends on accurate grounding and matching. Future directions include improving correspondence and robustness to imperfect crops, exploring adaptive aggregation to better protect low-scoring objects, and extending the reward to relational, interaction-level, and other non-localizable consistency. Scaling beyond six references additionally requires suitable training data and further evaluation.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

[2] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[3] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18392–18402, 2023.

[4] Huanqia Cai, Sihan Cao, Ruoyi Du, Peng Gao, Steven Hoi, Zhaohui Hou, Shijie Huang, Dengyang Jiang, Xin Jin, Liangchen Li, et al. Z-image: An efficient image generation foundation model with single-stream diffusion transformer. *arXiv preprint arXiv:2511.22699*, 2025.

[5] Siyu Cao, Hangting Chen, Peng Chen, Yiji Cheng, Yutao Cui, Xinchi Deng, Ying Dong, Kipper Gong, Tianpeng Gu, Xiusen Gu, et al. Hunyuanimage 3.0 technical report. *arXiv preprint arXiv:2509.23951*, 2025.

[6] Kejia Chen, Jiawen Zhang, Jiacong Hu, Kewei Gao, Jian Lou, Zunlei Feng, and Mingli Song. Token-level inference-time alignment for vision-language models. *arXiv preprint arXiv:2510.21794*, 2025.

[7] Shirui Chen, Cole Harrison, Ying-Chun Lee, Angela Jin Yang, Zhongzheng Ren, Lillian J Ratliff, Jiafei Duan, Dieter Fox, and Ranjay Krishna. Topreward: Token probabilities as hidden zero-shot rewards for robotics. *arXiv preprint arXiv:2602.19313*, 2026.

[8] Yangyi Chen, Karan Sikka, Michael Cogswell, Heng Ji, and Ajay Divakaran. Measuring and improving chain-of-thought reasoning in vision-language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 192–210, 2024.

[9] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

[10] Yu Gao, Lixue Gong, Qiushan Guo, Xiaoxia Hou, Zhichao Lai, Fanshi Li, Liang Li, Xiaochen Lian, Chao Liao, Liyang Liu, et al. Seedream 3.0 technical report. *arXiv preprint arXiv:2504.11346*, 2025.

[11] Lixue Gong, Xiaoxia Hou, Fanshi Li, Liang Li, Xiaochen Lian, Fei Liu, Liyang Liu, Wei Liu, Wei Lu, Yichun Shi, et al. Seedream 2.0: A native chinese-english bilingual image generation foundation model. *arXiv preprint arXiv:2503.07703*, 2025.

[12] Google DeepMind. Introducing Gemini 2.5 Flash Image, our state-of-the-art image model. `https://developers.googleblog.com/en/introducing-gemini-2-5-flash-image/`, 2024. Accessed: 2025-10-18.

[13] Google DeepMind. Introducing Nano Banana Pro. `https://blog.google/innovation-and-ai/products/nano-banana-pro/`, 2025. Accessed: 2026-05-07.

[14] Hanzhong Guo, Jie Wu, Jie Liu, Yu Gao, Zilyu Ye, Linxiao Yuan, Xionghui Wang, Yizhou Yu, and Weilin Huang. Leveraging verifier-based reinforcement learning in image editing. *arXiv preprint arXiv:2604.27505*, 2026.

[15] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

[16] Yuzhou Huang, Liangbin Xie, Xintao Wang, Ziyang Yuan, Xiaodong Cun, Yixiao Ge, Jiantao Zhou, Chao Dong, Rui Huang, Ruimao Zhang, et al. Smartedit: Exploring complex instruction-based image editing with multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8362–8371, 2024.

[17] Liya Ji, Chenyang Qi, and Qifeng Chen. Instruction-based image editing with planning, reasoning, and generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17506–17515, 2025.

[18] 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.

[19] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[20] Black Forest Labs. Flux. `https://github.com/black-forest-labs/flux`, 2024.

[21] Black Forest Labs. FLUX.2: Frontier Visual Intelligence. `https://bfl.ai/blog/flux-2`, 2025.

[22] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025.

[23] Hengjia Li, Liming Jiang, Qing Yan, Yizhi Song, Hao Kang, Zichuan Liu, Xin Lu, Boxi Wu, and Deng Cai. Thinkrl-edit: Thinking in reinforcement learning for reasoning-centric image editing. *arXiv preprint arXiv:2601.03467*, 2026.

[24] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022.

[25] Yanfeng Li, Kahou Chan, Yue Sun, Chantong Lam, Tong Tong, Zitong Yu, Keren Fu, Xiaohong Liu, and Tao Tan. Moedit: On learning quantity perception for multi-object image editing. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2683–2693, 2025.

[26] Zongjian Li, Zheyuan Liu, Qihui Zhang, Bin Lin, Feize Wu, Shenghai Yuan, Zhiyuan Yan, Yang Ye, Wangbo Yu, Yuwei Niu, et al. Uniworld-v2: Reinforce image editing with diffusion negative-aware finetuning and mllm implicit feedback. *arXiv preprint arXiv:2510.16888*, 2025.

[27] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[28] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

[29] Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. *arXiv preprint arXiv:2505.05470*, 2025.

[30] Jinlong Liu, Wanggui He, Peng Zhang, Mushui Liu, Hao Jiang, and Pipei Huang. Promptecho: Annotation-free reward from vision-language models for text-to-image reinforcement learning. *arXiv preprint arXiv:2604.12652*, 2026.

[31] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pages 38–55. Springer, 2024.

[32] Shiyu Liu, Yucheng Han, Peng Xing, Fukun Yin, Rui Wang, Wei Cheng, Jiaqi Liao, Yingming Wang, Honghao Fu, Chunrui Han, et al. Step1x-edit: A practical framework for general image editing. *arXiv preprint arXiv:2504.17761*, 2025.

[33] Weijian Luo, Tianyang Hu, Shifeng Zhang, Jiacheng Sun, Zhenguo Li, and Zhihua Zhang. Diff-instruct: A universal approach for transferring knowledge from pre-trained diffusion models. *Advances in Neural Information Processing Systems*, 36:76525–76546, 2023.

[34] Xin Luo, Jiahao Wang, Chenyuan Wu, Shitao Xiao, Xiyan Jiang, Defu Lian, Jiajun Zhang, Dong Liu, et al. Editscore: Unlocking online rl for image editing via high-fidelity reward modeling. *arXiv preprint arXiv:2509.23909*, 2025.

[35] Bowen Ping, Chengyou Jia, Minnan Luo, Changliang Xia, Xin Shen, Zhuohang Dang, and Hangwei Qian. Paco-rl: Advancing reinforcement learning for consistent image generation with pairwise reward modeling. *arXiv preprint arXiv:2512.04784*, 2025.

[36] Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026.

[37] 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.

[38] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

[39] Team Seedream, Yunpeng Chen, Yu Gao, Lixue Gong, Meng Guo, Qiushan Guo, Zhiyao Guo, Xiaoxia Hou, Weilin Huang, Yixuan Huang, et al. Seedream 4.0: Toward next-generation multimodal image generation, 2025. *URL https://arxiv. org/abs/2509.20427*, 6.

[40] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[41] Meituan LongCat Team, Hanghang Ma, Haoxian Tan, Jiale Huang, Junqiang Wu, Jun-Yan He, Lishuai Gao, Songlin Xiao, Xiaoming Wei, Xiaoqi Ma, et al. Longcat-image technical report. *arXiv preprint arXiv:2512.07584*, 2025.

[42] Super Intelligence Team, Changhao Qiao, Chao Hui, Chen Li, Cunzheng Wang, Dejia Song, Jiale Zhang, Jing Li, Qiang Xiang, Runqi Wang, et al. Firered-image-edit-1.0 technical report. *arXiv preprint arXiv:2602.13344*, 2026.

[43] Rui Tian, Mingfei Gao, Haiming Gang, Jiasen Lu, Zhe Gan, Yinfei Yang, Zuxuan Wu, and Afshin Dehghan. Unigen-1.5: Enhancing image generation and editing through reward unification in reinforcement learning. *arXiv preprint arXiv:2511.14760*, 2025.

[44] Hongyang Wei, Hongbo Liu, Zidong Wang, Yi Peng, Baixin Xu, Size Wu, Xuying Zhang, Xianglong He, Zexiang Liu, Peiyu Wang, et al. Skywork unipic 3.0: Unified multi-image composition via sequence modeling. *arXiv preprint arXiv:2601.15664*, 2026.

[45] Hongyang Wei, Baixin Xu, Hongbo Liu, Size Wu, Jie Liu, Yi Peng, Peiyu Wang, Zexiang Liu, Jingwen He, Yidan Xietian, et al. Skywork unipic 2.0: Building kontext model with online rl for unified multimodal model. *arXiv preprint arXiv:2509.04548*, 2025.

[46] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[47] Xinyu Wei, Kangrui Cen, Hongyang Wei, Zhen Guo, Bairui Li, Zeqing Wang, Jinrui Zhang, and Lei Zhang. Mico-150k: A comprehensive dataset advancing multi-image composition. *arXiv preprint arXiv:2512.07348*, 2025.

[48] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

[49] Yuhui Wu, Chenxi Xie, Ruibin Li, Liyi Chen, Qiaosi Yi, and Lei Zhang. Cocoedit: Content-consistent image editing via region regularized reinforcement learning. *arXiv preprint arXiv:2602.14068*, 2026.

[50] Junyan Ye, Dongzhi Jiang, Zihao Wang, Leqi Zhu, Zhenghao Hu, Zilong Huang, Jun He, Zhiyuan Yan, Jinghua Yu, Hongsheng Li, et al. Echo-4o: Harnessing the power of gpt-4o synthetic images for improved image generation. *arXiv preprint arXiv:2508.09987*, 2025.

[51] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6613–6623, 2024.

[52] Ruohong Zhang, Bowen Zhang, Yanghao Li, Haotian Zhang, Zhiqing Sun, Zhe Gan, Yinfei Yang, Ruoming Pang, and Yiming Yang. Improve vision language model chain-of-thought reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1662, 2025.

[53] Rosie Zhao, Anshul Shah, Xiaoyu Zhu, Xinke Deng, Zhongyu Jiang, Yang Yang, Joerg Liebelt, and Arnab Mondal. On robustness and chain-of-thought consistency of rl-finetuned vlms. *arXiv preprint arXiv:2602.12506*, 2026.

[54] Yiran Zhao, Yaoqi Ye, Xiang Liu, Michael Qizhe Shieh, and Trung Bui. Imageedit-r1: Boosting multi-agent image editing via reinforcement learning. *arXiv preprint arXiv:2603.08059*, 2026.

[55] Kaiwen Zheng, Huayu Chen, Haotian Ye, Haoxiang Wang, Qinsheng Zhang, Kai Jiang, Hang Su, Stefano Ermon, Jun Zhu, and Ming-Yu Liu. DiffusionNFT: Online diffusion reinforcement with forward process. In *The Fourteenth International Conference on Learning Representations*, 2026.

[56] Zijun Zhou, Yingying Deng, Xiangyu He, Weiming Dong, and Fan Tang. Multi-turn consistent image editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15792–15801, 2025.

[57] Hongyang Zhu, Haipeng Liu, Bo Fu, and Yang Wang. Mde-edit: Masked dual-editing for multi-object image editing via diffusion models. *arXiv preprint arXiv:2505.05101*, 2025.

A Appendix

A.1 Training Details

Single-object consistency reward model. We train Qwen3-VL-32B-Instruct [2] on PaCO-RL-Dataset [35] with LoRA rank 32 and a learning rate of 1×10^{-4} for 2,200 steps on 8 H800 GPUs.

Supervised fine-tuning. We fine-tune Qwen-Image-Edit-2511, our baseline model, on Nano-consistent-150k [50], MICo-150K [47], and UniPic 3.0 [44]. We use AdamW with a learning rate of 2×10^{-5}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1 \times 10^{-8}$, and weight decay 1×10^{-2}. Training uses 32 H800 GPUs, a total batch size of 128, and runs for 24,000 steps.

RL post-training. Starting from the SFT checkpoint, we further train on MICo-150K with LoRA [15] rank $r = 32$. RL uses 32 H800 GPUs, group size 16, 8 prompts per sample, and AdamW with learning rate 3×10^{-4}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1 \times 10^{-10}$, and weight decay 1×10^{-4}. To reduce reward hacking, we add a general-purpose VLM reward from Qwen3.5-122B-10A [36] to penalize unnatural outputs, structural distortions, and compositing artifacts. We set $\lambda_{\mathrm{NFT}} = 1$, $\lambda_{\mathrm{Reg}} = 10^{-4}$, and $\beta = 1$. See Algorithm 1 for pseudocode.

Algorithm 1: DeCoRL Diffusion NFT Post-Training

Input: Dataset $\mathcal{D}$, current policy v_θ, old policy v_{old}, reference policy v_{ref}
while *not converged* **do**
  Sample $(c, \mathcal{I}_{\mathrm{ref}}) \sim \mathcal{D}$;
  Generate a group of samples $\{I_{\mathrm{gen}}^k\}_{k=1}^K$ using v_{old};
  Compute rewards $\{S_k\}_{k=1}^K$ with the multi-object consistency reward in Eq. (11) and the general VLM reward;
  Normalize rewards into weights $\{r_k\}_{k=1}^K$ using Eq. (12);
  for *each sample* k **do**
    for *each sample timestep* t **do**
      Sample (z_t^k, t) using Eq. (4);
      Compute v_θ, v_{old}, and v_{ref};
      Construct v_θ^+ and v_θ^- using Eqs. (13);
      Compute $\mathcal{L}_{\mathrm{NFT}}$ using Eq. (13);
      Compute $\mathcal{L}_{\mathrm{Reg}}$ using Eq. (14);
  Update θ by minimizing the final objective in Eq. (14);

A.2 Benchmarks

A.2.1 DeCo-Bench.

Scores. DeCo-Bench evaluates multi-image composition primarily from the perspective of *cross-image consistency*, i.e., whether instruction-relevant entities from multiple reference images are correctly and consistently realized in the generated result. It contains settings with $N \in \{2, 3, 4, 5, 6\}$ reference images; each setting includes 40 test cases (200 cases in total), and all reference images are real-world photographs.

For each test case, we first crop the instruction-relevant objects from every reference image, and crop the corresponding objects from the generated image. We then score each cropped reference–generation object pair using Gemini-3.1-Pro (the evaluation prompt is provided in Figure 8). The final score for a sample is computed by averaging all object-level scores:

$$\mathrm{Score}(c) = \frac{1}{|\mathcal{O}_c|} \sum_{o \in \mathcal{O}_c} s(o), \tag{15}$$

where $\mathcal{O}_c$ denotes the set of all instruction-relevant objects in case c, and $s(o)$ is the Gemini-3.1-Pro score for object o computed from the corresponding cropped reference and generated regions.

Aggregation. Let $\mathcal{C}_N$ denote the set of cases with N reference images, and let $\mathcal{C} = \bigcup_{N \in \{2,3,4,5,6\}} \mathcal{C}_N$ denote all cases. We report:

$$\mathrm{Score}_N = \frac{1}{|\mathcal{C}_N|} \sum_{c \in \mathcal{C}_N} \mathrm{Score}(c), \quad \mathrm{Score}_{\mathrm{overall}} = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \mathrm{Score}(c). \tag{16}$$

A.2.2 MICo-Bench

Scores. MICo-Bench evaluates each multi-image composition case by jointly considering (i) *source preservation* and (ii) *instruction alignment*. Concretely, the final score is computed as:

$$\text{Score} = W \times \text{SC} \times \text{PQ}. \tag{17}$$

Tasks. MICo-Bench consists of four tasks:

1. **Object-Centric (138 cases):** compose 1–4 distinct objects, or insert 1–2 objects into a semantically compatible scene.
2. **Human-Centric (168 cases):** compose multiple persons, or place 1–2 persons into a compatible scene (with explicit gender control).
3. **HOI (291 cases):** evaluate human–object interactions, including Person+Object, Person+Clothes, and their combinations.
4. **De&Re (300 cases):** decompose real portraits into human/clothing/object/scene components and then recompose them to assess holistic controllability.

Weight W. The fraction of source images correctly preserved in the generated output. Faces are verified via ArcFace cosine similarity; non-face elements (objects, clothes, scenes) are verified via a VLM (i.e., Qwen3-VL-30B-A3B-Instruct).

Semantic Consistency (SC). To avoid cross-image attention overload, the evaluator compares the generated image against a *human-verified reference image* (instead of jointly attending to all sources) and outputs two sub-scores:

1. **Prompt Following** (PF), measuring compliance with the text instruction (required entities, relations, and scene constraints),
2. **Subject Resemblance** (SR), measuring how well the generation matches the reference in subject identity/appearance and semantic content.

We compute:

$$\text{SC} = \sqrt{\text{PF} \times \text{SR}}. \tag{18}$$

Perceptual Quality (PQ). PQ is evaluated from the generated image alone, including:

1. **Naturalness** (overall realism and coherence of lighting, shadows, geometry, and materials),
2. **Artifacts** (absence of visible defects such as distortions, duplicated limbs, and unnatural textures).

We compute:

$$\text{PQ} = \sqrt{\text{Naturalness} \times \text{Artifacts}}. \tag{19}$$

Aggregation. Let $\mathcal{C}_t$ denote the set of evaluation cases in task t, and let $\mathcal{C} = \bigcup_t \mathcal{C}_t$ denote the set of all cases across tasks. We report the task-wise and overall scores as:

$$\text{Score}_t = \frac{1}{|\mathcal{C}_t|} \sum_{c \in \mathcal{C}_t} \text{Score}(c), \quad \text{Score}_{\text{overall}} = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \text{Score}(c). \tag{20}$$

A.3 More Visualizations

We provide additional visualization examples on DeCo-Bench (Figure 6) and MICo-Bench (Figure 7). As can be seen from these visual results, our method achieves excellent MICo consistency and visual quality.

B Additional Validation and Discussion

Independent validation of the object-level reward evaluator. To assess the trained reward model separately from downstream RL, we evaluated its agreement with human preference labels on EditReward-Bench, which contains approximately 3,000 original/edited image pairs. Table 5 reports the accuracy values from this evaluation. The trained evaluator improves upon its untuned counterpart (0.721 versus 0.354), supporting the importance of using a reliable object-level evaluator within the decomposed reward framework. This evaluation measures agreement with the benchmark labels, not reward calibration.



Figure 6: Visualization examples from the DeCo-Bench (6 images).

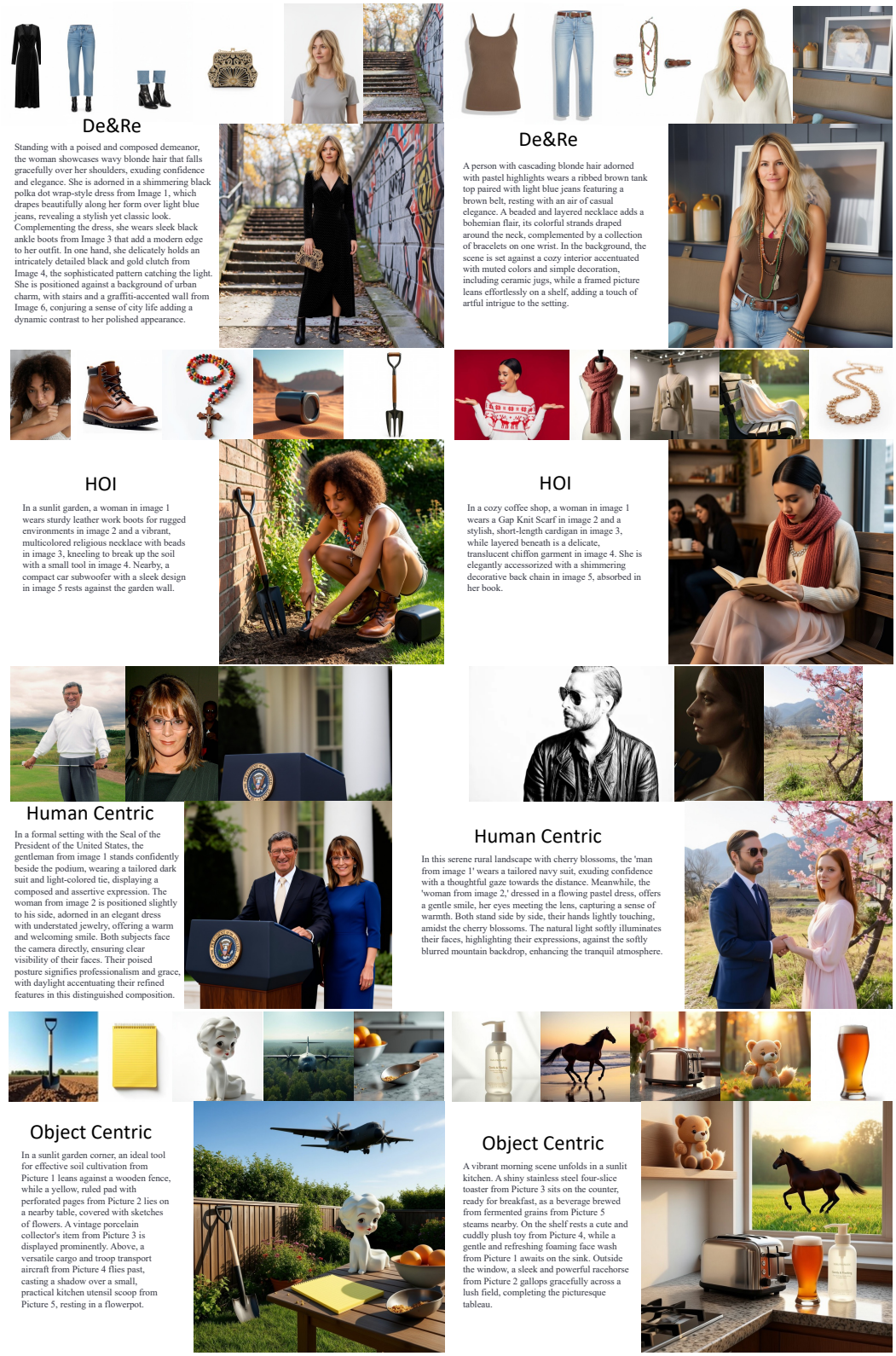

Figure 7: Visualization examples from the MICo-Bench (De&Re, HOI, Human Centric and Object Centric).

```
# Role
You are a senior expert in computer vision and image quality assessment, with deep
     expertise in semantic feature extraction and target consistency analysis.

# Task
Target evaluation label (Label): [obj_label]
Please compare Image A (reference image) and Image B (generated image), and focus your
     attention exclusively on the entity indicated by the target evaluation label above.
     Assess the consistency of Image B relative to Image A only for that target.

# Core Constraints
You must strictly follow the principles below for analysis and scoring:

1. Absolute focus (exclusivity): Evaluate strictly and only the target [obj_label]
     itself. You must completely ignore the background, environment, unrelated people,
     or any other distracting objects in the scene. Never deduct points because of
     additions, removals, or substitutions of irrelevant elements.
2. Comparison of major color regions (critically important): No matter what category the
     target belongs to, you must strictly compare its major color regions.
   - Carefully compare the target's intrinsic dominant colors, color distribution,
       proportion of color blocks, and key color characteristics.
   - Color changes in core regions, loss of key color blocks, or color inversion must be
       treated as severe inconsistencies.
3. Generalized feature evaluation dimensions: Based on the semantic category of
     [obj_label], extract its core features for consistency evaluation:
   - If the target is a living being (such as a person or animal): focus absolutely on
       biometric identity consistency. Compare only facial feature proportions, facial
       characteristics, hairstyle/fur texture, face shape/contour, bone structure, and
       distinctive biological details (such as moles or markings). Ignore clothing,
       accessories, and body pose changes.
   - If the target is a non-living object (such as an item, product, or building): focus
       absolutely on physical and design attributes. Compare shape contour, physical
       structure, material texture, surface patterns, and iconic design elements (such
       as logos, text, or specific constructions).
4. Exclude physical and viewpoint interference: You must cognitively decouple the target
     itself from its interaction with the external environment. Compare only the
     target's intrinsic identity or attributes, and strictly ignore visual differences
     caused by viewpoint (distance, front/side angle, pitch), pose variation,
     environmental lighting (brightness, warm/cool light, reflected ambient color), or
     image noise.

# Output Format
You must output one and only one valid JSON object. Do not include any extra explanation
     text or Markdown formatting. You must provide the reason first (including analysis
     of shape/features and major color regions), and then the score (an integer from 0
     to 10, where 10 means perfectly consistent and 0 means completely inconsistent).
     The format must be:

'''json
{
  "reason": "Brief explanation of the evaluation basis (must include comparison of core
       features and major color regions, while excluding viewpoint/lighting
       interference)",
  "score": "consistency score"
}
```

Figure 8: The evaluation prompt (Gemini-3.1-Pro) of DeCo-Bench.

Table 5: Human-label agreement accuracy on EditReward-Bench.

Evaluator	Accuracy↑
GPT-4.1	0.602
GPT-5	0.669
Vanilla reward VLM	0.354
Trained reward VLM (ours)	**0.721**

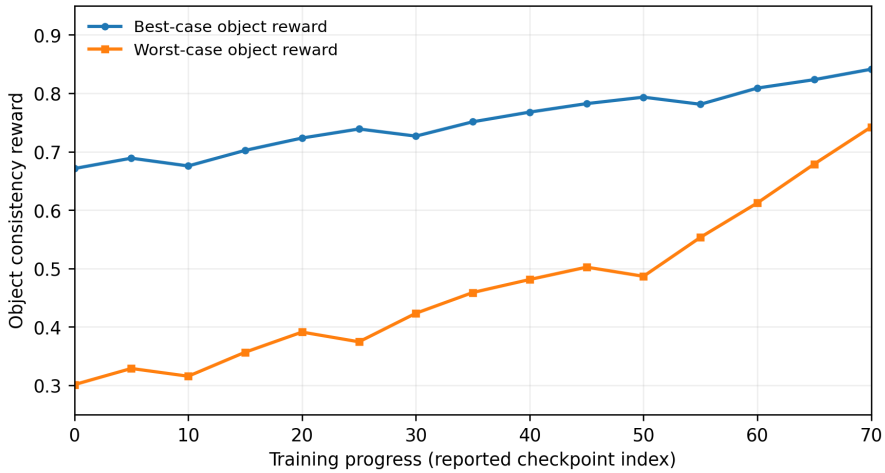

Figure 9: Best-case and worst-case object consistency rewards over the reported training checkpoints.

Grounding and object matching diagnostics. We sampled 10,000 examples from the training set, applied Grounding DINO to identify the relevant object regions, and used a VLM to judge whether the detected crops matched the assigned object labels. Under this VLM-based verification protocol, the resulting label–crop matching accuracy was 94.6%. This result is a diagnostic proxy and should not be interpreted as human-annotated bounding-box localization accuracy or same-category instance-matching accuracy.

Behavior of best- and worst-case object rewards. Figure 9 reports the best-case and worst-case object rewards over the reported training checkpoints. The worst-case reward increases from 0.3012 to 0.7423, while the best-case reward increases from 0.6712 to 0.8412. Thus, both increase under mean aggregation, and the worst-case object shows a larger absolute increase over these checkpoints. These observations are consistent with improvements beyond the easiest objects, although they do not establish that mean aggregation is optimal among possible alternatives.

Evaluation scope and potential reward–metric alignment. DeCo-Bench covers 2–6 reference images with 40 test cases per setting (200 cases total). It is designed to assess the localizable object-level consistency targeted by our reward and is therefore an in-domain evaluation. The separately developed MICo-Bench provides complementary evidence beyond this reward-aligned benchmark, including improvements in subject resemblance and perceptual-quality metrics. Nevertheless, this does not replace a dedicated study of possible evaluator bias. The current model was trained for settings with up to six reference images; reliability beyond this range has not been demonstrated.

Choice of RL optimization and auxiliary reward. DeCoRL employs Diffusion NFT as a practical post-training optimizer; the object-level reward design is conceptually orthogonal to the choice of diffusion RL algorithm. Early investigations of flow-based GRPO variants suggested higher resource demands and sensitivity to optimization settings, but no matched-training controlled comparison is reported here. The general-purpose auxiliary VLM reward described in Section 3.6 is used to discourage implausible global compositions, including artifacts and incorrect object–subject interactions that a local crop-based score may miss.

Limitations. Our decomposed reward primarily captures localizable object identity and appearance. Cropped objects cannot by themselves fully represent abstract styles, spatial relations, or interaction and binding constraints. Grounding and same-category object matching can fail, while VLM-based evaluation may introduce bias or be vulnerable to reward hacking. DeCo-Bench is relatively small, and the main experiments use one

editing backbone and 2–6 reference images. The pipeline also relies on multiple large models, increasing computational and reproducibility costs. A controlled holistic-versus-object-level reward comparison under matched RL training remains an important avenue for future work.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [N/A]

   Justification: The paper has no limitation.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not include theoretical results.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.

- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper provides relevant information, and even if there are random factors, they do not affect the main claims and/or conclusions of the paper.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: We will consider releasing the relevant data and code if the paper is accepted.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.

- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: In the experimental section, we provide the training and test details, and full details are presented in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: All experiments were conducted with the same equipment and random seed. The costs of training and metric evaluation (via closed-source model API calls) were too high to allow multiple runs, so error bars, confidence intervals, and statistical significance tests are not provided.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: In the experimental section, we provide the training resources and some training parameters.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: Our topic (Multi-Image Consistency) does not involve societal impact.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the original paper that produced the code package or dataset.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.