OSPO: Object-Centric Self-Improving Preference Optimization for Text-to-Image Generation

Supplementary Material

A. Preliminary Details

This section provides additional explanations of the definitions of Preference-Ambiguous Pairs and Preference-False Pairs introduced in Section 3.1 of the main paper. Along with that, we provide quantitative method used to measure their proportions in the dataset using per-question alignment score s (defined in Section 3.6.1 of the main paper).

Let define the binary indicator $\mathbb{I}_k(y) = \mathbb{1}\{s_k(y) > 0\}$, where $s_k(y)$ denotes per-question score of image y on the k-th atomic question.

- **Preference-Ambiguous Pair**: A pair (y_w, y_ℓ) is a *preference-ambiguous* pair if the two images exhibit the same correctness pattern across all semantic units: $\forall k, \quad \mathbb{I}_k(y_w) = \mathbb{I}_k(y_\ell)$. In this case, both images are either correct or incorrect on the same semantic units, providing no meaningful preference signal. Thus, the distinction between preferred and non-preferred becomes ambiguous.
- **Preference-False Pair**: A pair (y_w, y_ℓ) is a *preference-false* pair if both images are either entirely correct or entirely incorrect across all semantic units: $(\forall k, \quad \mathbb{I}_k(y_w) = \mathbb{I}_k(y_\ell) = 0)$ or $(\forall k, \quad \mathbb{I}_k(y_w) = \mathbb{I}_k(y_\ell) = 1)$. This occurs when prompts are too simple (all correct) or too difficult (all wrong), causing *Best-of-N sampling* to force a label assignment despite no meaningful quality difference, actively introducing contradictory supervision. Preference-false pairs form a strict subset of preference-ambiguous pairs.

B. Framework Details

B.1. Prompt Perturbation

In the prompt perturbation stage, we apply different perturbation strategies depending on the prompt category. For Attribute and Layout prompts that contain a single object (e.g., "a green dress", "two trees"), we restrict perturbations to the Replace strategy and generate K variants by changing only the random seed. For Non-spatial prompts, we similarly limit perturbations to Replace and Drop, since other strategies are less meaningful for this category. For all other prompts, including multi-object Attribute/Layout prompts and Complex prompts, we randomly sample from all three perturbation strategies.

B.2. Object Mask Extraction

Attention Map Extraction. To obtain object masks, we begin by extracting all noun phrases from the input prompt using the spaCy linguistic parser, where each noun phrase is treated as an object candidate. For each object token (or token span), we probe the model's self-attention mechanism during image generation.

Algorithm 1 Object-Centric Token Weighting

Input: Binary object mask M, Gaussian kernel G, Emphasis strength α
Output: Weight vector $\mathbf{w} \in \mathbb{R}^{T^2}$, where T^2 denotes the number of visual tokens per image
$\tilde{M} \leftarrow G * M$ ▷ Gaussian smoothing
$\hat{M} \leftarrow \dfrac{\tilde{M} - \min(\tilde{M})}{\max(\tilde{M}) - \min(\tilde{M})}$ ▷ Normalization
$\mathbf{m} \leftarrow \text{Flatten}(\hat{M})$
$\mathbf{w} \leftarrow \mathbf{1} + \alpha \cdot \mathbf{m}$
return $\mathbf{w}$

Let $\mathcal{L}$ be the set of selected transformer layers—excluding the earliest and latest layers—from which attention maps are extracted. For each layer $\ell \in \mathcal{L}$, we obtain the multi-head self-attention weights $A_\ell \in \mathbb{R}^{|\mathcal{H}| \times (L+T^2) \times (L+T^2)}$, where L and T^2 denotes the number of text tokens and visual tokens, respectively, and $\mathcal{H}$ denotes the set of attention heads with $|\mathcal{H}|$ heads.

Since we are interested in attention from text tokens to visual tokens, we extract the query-to-visual slice for each head $h \in \mathcal{H}$. Let the visual-key index set be $\mathcal{V} = \{L+1, \ldots, L+T^2\}$. For each head h, the attention from any query token i to the visual tokens is: $A_\ell(h)[i, \mathcal{V}] \in \mathbb{R}^{T^2}$. Let $S \subseteq \{1, \ldots, L\}$ be the set of query token indices corresponding to a particular object span (possibly a single token or a multi-token phrase). We compute the object-level attention for head h at layer ℓ by averaging the attention vectors of all query tokens in S:

$$A^{(S)}_{\ell,h} = \frac{1}{|S|} \sum_{i \in S} A_\ell(h)[i, \mathcal{V}] \quad \in \mathbb{R}^{T^2}. \tag{3}$$

This yields a single attention distribution over all T^2 visual tokens for the object, per head and per layer.

Attention Map Aggregation. Next, we average the object-conditioned attention weights across all heads $h \in \mathcal{H}$ and layers $\ell \in \mathcal{L}$ to obtain a single attention distribution over the T^2 visual tokens:

$$\bar{A}^{(S)} = \frac{1}{|\mathcal{L}|\,|\mathcal{H}|} \sum_{\ell \in \mathcal{L}} \sum_{h \in \mathcal{H}} A^{(S)}_{\ell,h} \quad \in \mathbb{R}^{T^2}. \tag{4}$$

The attention distribution is then min–max normalized to $[0, 1]$ and binarized using Otsu's thresholding [28]. The

Table 5. Comparison of training configurations for Janus-Pro-1B, Janus-Pro-7B, and UniTok-MLLM-8B.

Configuration	Janus-Pro-1B	Janus-Pro-7B	UniTok-MLLM-8B
LoRA Rank	64	64	32
LoRA Alpha	128	128	64
Optimizer	AdamW	AdamW	AdamW
Optimizer Hyperparameters	β_1: 0.9, β_2: 0.95, ϵ: 1e-8	β_1: 0.9, β_2: 0.95, ϵ: 1e-8	β_1: 0.9, β_2: 0.95, ϵ: 1e-8
Gradient Clipping	1.0	1.0	1.0
Weight Decay	0.0	0.0	0.0
Learning Rate	1e-5	1e-5	6e-6
Learning Rate Scheduler	Cosine	Cosine	Constant
Total Training Steps	500	800	300
Total Batch Size	128	128	128
SimPO Hyperparameters	β: 5.0, γ: 2.5	β: 5.0, γ: 2.5	β: 3.0 , γ: 2.5
SFT Loss Weight (λ)	2.0	2.0	2.0
Mixed Precision	bf16	bf16	bf16

aggregated attention map is binarized by applying Otsu's threshold τ_{otsu}:

$$M^{(\mathcal{S})} = \begin{cases} 1, & \bar{A}^{(\mathcal{S})} \geq \tau_{\text{otsu}}, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

B.3. Object Mask Application

During training, the binary Otsu mask M is transformed into a continuous per-token mask $\mathbf{m} \in [0,1]^{T^2}$ through Gaussian smoothing and min-max normalization, as described in Algorithm 1. The resulting weight vector $\mathbf{w}$ modulates the contribution of each visual token to the loss: object-relevant tokens receive larger weights while background tokens receive smaller weights, with a smooth transition between them. The weighting is defined solely over visual tokens — text tokens do not participate in the log-probability computation. In effect, the weighted token sequence amplifies gradients on object-relevant visual regions in both preferred and rejected outputs, guiding the preference optimization to focus on correcting object-specific errors.

C. Implementation Details

C.1. Data Construction

From stage 1 to stage 4, all text and image data are generated using a single MLLM. For text generation, stage 1 (Prompt Generation) adopts stochastic sampling with temperature=1.0 and top-p=0.9 rather than greedy decoding, and further encourages diversity by periodically rotating the in-context demonstration set throughout generation. Stage 2 (Prompt Perturbation and Densification) uses the same sampling configuration. For stage 3 (Image and Object Mask Generation), we set temperature=1.0 and cfg-weight=5.0. Lastly for stage 4 (VQA-based Preference Pair Construction), we use a single beam with temperature=1.2 and top-p=0.5 for question generation and answering.

Table 6. Stage-wise wall-clock time breakdown of OSPO for a single iteration.

Stage	Time (h)
(1) Prompt Generation	0.07
(2) Prompt Perturbation and Densification	5.06
(3) Image and Object Mask Generation	6.80
(4) VQA-based Preference Pair Construction	6.45
(5) Preference Optimization	1.83
Total	**20.21**

C.2. Preference Optimization

We provide detailed training configurations for OSPO in this section. We employ Low-Rank Adaptation (LoRA) [16] method for efficient optimization. Table 5 presents the training configurations for the two main experimental model sizes, Janus-Pro-1B and Janus-Pro-7B [3], as well as the additional backbone model Unitok-MLLM-8B [25], which is discussed in Section E.4.

C.3. Stage-wise Time Cost Breakdown

Table 6 presents the wall-clock time for each stage of OSPO over a single iteration, measured on eight NVIDIA A100 (80GB) GPUs. In the default configuration, where 25,000 text prompts are generated in stage 1 and a pair size of 3 is employed in the subsequent stages, the total runtime amounts to 20.21 hours.

D. Evaluation Benchmark Details

This section describes the evaluation benchmarks in detail, focusing on how generated images are assessed under each benchmark's metrics and expert models. Notably, with the exception of T2ICompBench++, the prompt distributions of all other benchmarks do not overlap with our training data, indicating that performance on these benchmarks reflects

out-of-distribution generalization capability.

- **T2I-CompBench++**: Consists of four prompt categories—Attribute, Layout, Non-spatial Relationship, and Complex Composition. Each category is evaluated using an appropriate expert model or metric tailored to its specific requirements. The Attribute category is measured by the Disentangled BLIP-VQA metric, which evaluates image-prompt semantic alignment via binary VQA responses from BLIP [22]. The Layout category is measured by a UniDet-based metric that integrates depth estimation [34] and object detection [57] to assess 2D/3D spatial relationships and object counting. The Non-spatial Relationship category is measured using the CLIP-T metric, and the Complex Composition category is scored by a weighted combination of the three aforementioned metrics, referred to as the 3-in-1 metric. A distinguishing characteristic of this benchmark is deliberate inclusion of unfamiliar or counterintuitive compositional prompts such as *"a white banana"* which poses a greater challenge to current text-to-image generation models and provides a more rigorous evaluation of compositional generalization.
- **DPGBench**: Comprises five prompt categories—Global, Entity, Attribute, Relation, and Other. It is distinguished by its use of notably longer prompts compared to other benchmarks. Evaluation is conducted uniformly across all categories using the mPLUG-large [54] model as the adjudicator, which assesses generated images against a set of designated questions. Following the scoring protocol of DSG [6], the prompt score is computed as the mean of all question scores associated with a single prompt, and the overall DPG-Bench score is defined as the mean of all prompt scores.
- **GenEval**: Evaluates compositional image generation across six categories—Single Object, Two Object, Counting, Colors, Position, and Attribute Binding. A Mask2Former [5] model detects objects to verify presence, count, color (via CLIP [32]-based zero-shot classification), and relative spatial position depending on the category. Each image receives a binary correctness score, averaged per task and then across all tasks to yield the final score.

E. Additional Analysis

To thoroughly evaluate OSPO, we conduct a series of analyses and ablation studies using the Janus-Pro-7B model. We begin by examining broader properties of OSPO, including prompt diversity, object mask quality, and image generation quality. We then evaluate its generalization ability to an alternative backbone model. Next, we analyze each stage of the OSPO framework. In the data construction stage, we examine the effect of prompt densification and the influence of the alignment-score threshold used for preference-pair filtering. In the training stage, we examine the effect of the key hyperparameter β, compare SimPO with DPO, and evaluate iterative training to verify OSPO's consistent self-boosting capability. Beyond these components, we investigate the presence of confirmation bias—a commonly noted limitation in self-improving systems—by comparing the self-rewarding model against an external VQA-based reward model. Finally, we explore changes in the model's internal dynamics to provide further interpretability.

Metric	Category				Overall
	Attribute	Layout	Non-Spatial	Complex	
Self-BLEU	0.176	0.089	0.190	0.313	0.067

Table 7. Self-BLEU across categories, measuring lexical diversity of generated prompts. Lower is better.

E.1. Prompt Diversity under Self-Generation

To quantify prompt diversity, we report the lexical diversity of our generated prompts across categories by Self-BLEU [58] which is a metric to evaluate the diversity of the generated data. As shown in Table 7, the overall Self-BLEU indicates low n-gram overlap across prompts, confirming that the generated prompts are not templated or repetitive. At the category level, the Layout category achieves the lowest Self-BLEU, reflecting high structural variety in spatial descriptions, while the Complex category shows the highest Self-BLEU, which is expected given that complex prompts inherently share more compositional patterns. Overall, these results demonstrate that our prompt generation produces lexically diverse training prompts across all categories.

E.2. mIoU of Object Mask against GT

To evaluate whether our attention-derived object masks accurately localize the target objects, we measure mIoU against reference segmentation masks produced by Grounded-SAM-2 [35] (GroundingDINO-Base + SAM2). We sample 500 training instances per category from iteration 1 and compute mIoU between OSPO masks and the corresponding Grounded-SAM-2 masks generated using the same set of object words. OSPO masks achieve an average mIoU of **0.5475** for preferred images and **0.5284** for non-preferred images, indicating that the attention-derived masks substantially overlap with detector-guided segmentations and capture meaningful object regions rather than arbitrarily covering background areas. Furthermore, as shown in Table 8, the downstream performance gap between OSPO with self-generated masks and OSPO with Grounded-SAM-2 masks is negligible across all benchmarks, demonstrating that our self-supervised masking approach is a practical and effective alternative to external segmentation models.

Table 8. Downstream benchmark performance of OSPO with self-generated masks and Grounded-SAM-2 masks. ↑ indicates higher is better, with bold highlighting the best score.

Model	Size	T2I-CompBench++↑				DPGBench↑	GenEval↑
		Attribute	Layout	Non-Spatial	Complex	Overall	Overall
Janus-Pro	7B	0.65	0.41	0.31	0.36	81.87	0.796
+ OSPO w/ self mask (default)	7B	0.756	0.447	0.316	**0.415**	**85.61**	0.830
+ OSPO w/ GroundedSAM2 mask	7B	**0.769**	**0.449**	**0.317**	0.407	85.45	**0.831**

Table 9. Performance comparison of the Unitok-MLLM-8B on T2I-CompBench++, DPGBench, and GenEval. ↑ indicates higher is better, with bold highlighting the best score.

Model	Size	T2I-CompBench++↑				DPGBench↑	GenEval↑
		Attribute	Layout	Non-Spatial	Complex	Overall	Overall
Unitok-MLLM	8B	0.646	**0.414**	0.313	0.362	81.87	0.590
+ OSPO	8B	**0.677**	**0.414**	**0.316**	**0.379**	**83.34**	**0.620**

Table 10. Comparison on MSCOCO-30K [2]. ↑ indicates higher is better, with bold highlighting the best score.

Model	Size	CLIP-T↑	CLIP-I↑	LPIPS↓
Janus-Pro	1B	0.222	0.661	0.7137
+ SILMM	1B	0.227	0.667	0.7159
+ OSPO (ours)	1B	**0.235**	**0.684**	**0.6996**
Janus-Pro	7B	0.235	0.704	0.7060
+ SILMM	7B	0.237	**0.707**	0.7023
+ FocusDiff	7B	0.232	0.672	-
+ OSPO (ours)	7B	**0.239**	**0.707**	**0.6959**

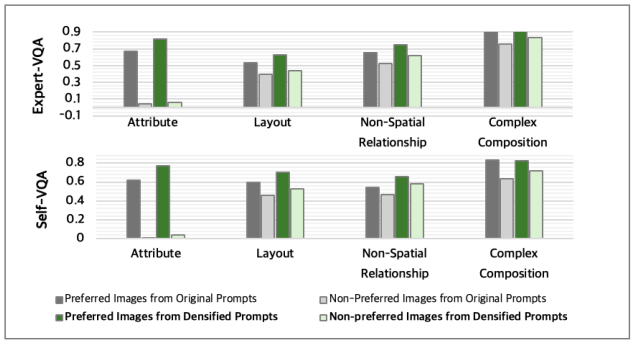


Figure 6. Mean alignment score computed over all alignment scores S for preference pairs generated from original prompts (gray) and densified prompts (green), reported separately for each category. "Expert-VQA" reports scores estimated by the expert model, where as "Self-VQA" reports scores estimated by the model itself under the OSPO framework.

E.3. General T2I Generation Quality

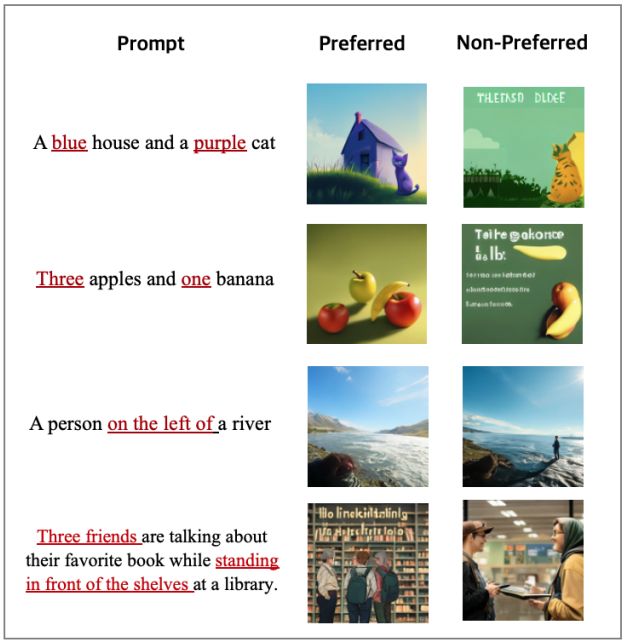
To assess the robustness of our framework under various input formats, we evaluate it on extensive compositional T2I benchmarks. Furthermore, we also measure CLIP-T, CLIP-I [32], and LPIPS [55] score on the MSCOCO-30K [2] dataset to examine its general T2I generation capability. As reported in Table 10, despite being designed for fine-grained semantic supervision, OSPO exhibits improvements in general T2I performance as well. This stands in contrast to FocusDiff [29], another fine-grained approach, which exhibits a performance decline on the same task.

E.4. Effect of Backbone Model

To investigate the model-agnostic applicability of our framework, we conduct additional experiments using alternative MLLMs: UniTok-MLLM-8B [25] with a different architecture. The model exhibits notably weaker image understanding and reasoning capabilities compared to Janus-Pro-7B, which are critical for our data generation pipeline. As shown in Table 9, applying OSPO consistently improves performance across different model architectures, demonstrating the framework's generalizability and effectiveness.

E.5. Effect of Prompt Densification

We have examined the effect of prompt densification on benchmark performance in Section 5.1.3 of the main paper. Here, we additionally evaluate the intrinsic image quality by examining how faithfully the generated outputs capture the input text, comparing images synthesized from the original prompts with those synthesized from densified prompts. To measure fidelity at a fine-grained level, we use the same decompositional VQA protocol described in Section 3.6.1 of the main paper, querying a Vision-Language Model with atomic semantic questions. Using the two preference-pair datasets constructed in Section 5.1.3—(1) the baseline set whose images are generated from the original prompts, and (2) our OSPO-constructed set generated from densified prompts. We then evaluate their text–image alignment using the alignment score S assigned by the expert VQA model



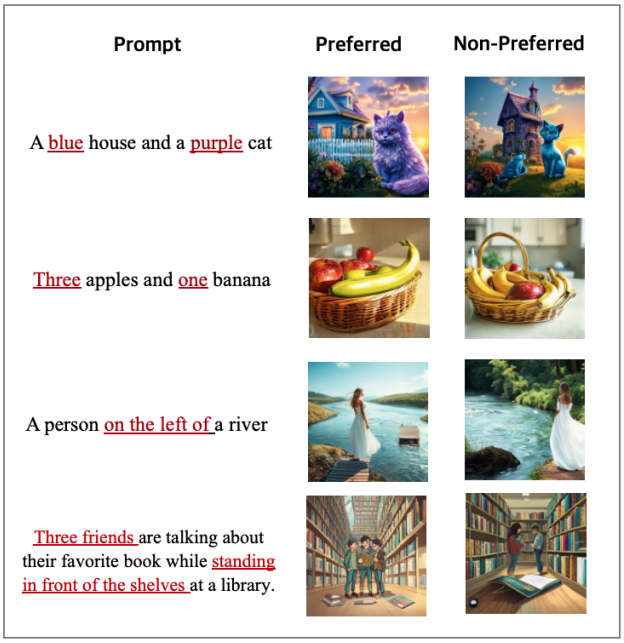


Figure 7. (Left) Training data examples whose images are generated from non-densified (original) prompts. (Right) Training data examples whose images are generated from densified prompts.

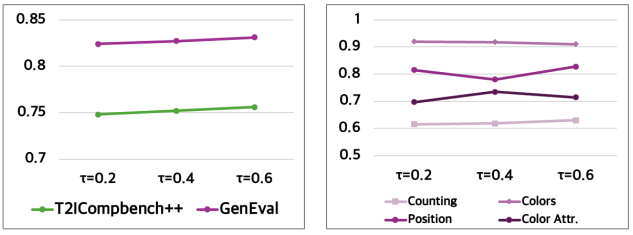


Figure 8. Effect of different filtering threshold (τ) values. (Left) Performance comparison on T2I-CompBench++ (attribute) and GenEval (overall). (Right) Category-wise GenEval results. Higher is better for all benchmarks.

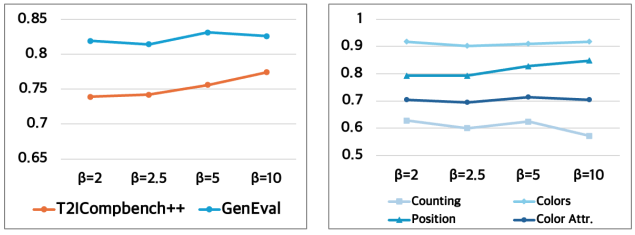


Figure 9. Effect of different β values in SimPO. (Left) Performance comparison on T2I-CompBench++ (attribute) and GenEval (overall). (Right) Category-wise GenEval results. Higher is better for all benchmarks.

InternVL3.5-14B [44]. As shown in Figure 6, prompt densification consistently improves the fidelity of preferred images across all categories, and even non-preferred images exhibit modest gains. These results indicate that our densification enhances the overall semantic consistency of generated images. Qualitative examples illustrating these visual differences are shown in Figure 7.

E.6. Effect of Filtering Threshold

In the preference-pair construction stage, we filter samples by requiring the preferred image to have an alignment score S above a threshold τ. To evaluate the influence of this filtering criterion, we vary τ and compare the resulting training performance. Our default threshold is $\tau = 0.6$. As shown in Figure 8, increasing the threshold slightly improves benchmark performance. However, this must be considered alongside the fact that a higher threshold reduces the size of the training dataset.

E.7. Effect of Training Strategy

Within the OSPO framework, we train our models using the SimPO [27] algorithm, a simplified variant of DPO [33] that removes the need for a reference model. In SimPO, the hyperparameter β scales the reward margin between preferred and non-preferred images, controlling the strength of preference supervision.

Hyperparameter β. In Preference Optimization, the hyperparameter β is known to strongly influence optimization dynamics [48]. To understand its impact, we conduct an ablation study varying β and analyze how different values affect training results. Figure 9 shows that increasing β im-

Table 11. Performance comparison of SimPO and DPO training within the OSPO framework on T2I-CompBench++, DPGBench, and GenEval. ↑ indicates higher is better, with bold highlighting the best score.

Model	Size	T2I-CompBench++↑				DPGBench↑	GenEval↑
		Attribute	Layout	Non-Spatial	Complex	Overall	Overall
Janus-Pro	7B	0.418	0.292	0.311	0.387	81.87	0.796
+ OSPO w/ SimPO (default)	7B	**0.756**	0.447	**0.316**	**0.415**	**85.61**	**0.830**
+ OSPO w/ DPO	7B	0.700	**0.448**	0.315	0.400	84.80	0.808

Table 12. Performance improvement of Janus-Pro-7B on T2I-CompBench++, DPGBench, and GenEval with OSPO over three iterations. ↑ indicates higher is better, with bold highlighting the best score.

Model	Size	T2I-CompBench++↑				DPGBench↑	GenEval↑
		Attribute	Layout	Non-Spatial	Complex	Overall	Overall
Janus-Pro	7B	0.418	0.292	0.311	0.387	81.87	0.796
+ OSPO (Iter. 1)	7B	0.756	0.447	0.316	0.415	85.61	0.830
+ OSPO (Iter. 2)	7B	0.756	**0.455**	0.317	0.412	**86.00**	0.831
+ OSPO (Iter. 3)	7B	**0.760**	**0.455**	**0.318**	**0.418**	85.84	**0.835**

proves performance in the Attribute category, but also raises the risk of reduced counting accuracy. Thus, β must be carefully balanced.

Training Algorithm. Instead of SimPO, we train Janus-Pro-7B using the standard DPO algorithm [33] while keeping our object-weighted token masking strategy unchanged. For DPO, we set $\beta = 0.5$, a learning rate of 1e-5, a total batch size of 128, and 500 training steps; all other configurations follow the default OSPO training setup. As shown in Table 11, DPO also demonstrates noticeable improvements across benchmarks; however, SimPO consistently achieves larger performance gains. That's because it removes the reference-model constraint and directly optimizes the reward gap, enabling stronger object-level corrections and model updates that are essential for compositional text-image alignment.

E.8. Effect of Iterative Training

As shown in Table 12, we present the progression of model performances across training iterations. The results indicate that the OSPO framework leads to steady and meaningful improvements in text-image alignment across a wide range of compositional categories, with a particularly notable improvement observed in the first iteration and continuing through subsequent iterations.

E.9. Effect of Confirmation Bias

In the OSPO framework, single MLLM serves both as a generator and a reward estimator, following the self-improving principle of using internal feedback to guide learning. In this section, we evaluate the validity of using the MLLM itself, specifically Janus-Pro-7B, as a reward estimator within the framework, by comparing its reward outputs with those of a relatively strong vision-language model, InternVL3.5-14B [44]. In the OSPO framework, the VQA-based alignment score S serves as a reward output.

Metric	Category				Overall
	Attribute	Layout	Non-Spatial	Complex	
Hit Rate	0.91	0.80	0.85	0.90	0.87

Table 13. Hit rate across categories, measuring agreement between the self-rewarding model and the expert VQA model.

Table 13 reports the hit rate, defined as the proportion of matching binary answers ("yes" or "no") between the self-model and the expert VQA model for the same candidate images. The hit rates remain high across categories (approximately 85%), indicating that the self-model's VQA outputs are largely consistent with those of the stronger model and therefore reasonably reliable. The largest discrepancy appears in the Layout category, which is expected since many MLLMs—including the self-model—tend to struggle with spatial reasoning.

To assess whether differences in hit rate influence downstream training, we constructed a training dataset in which the VQA model used during filtering was replaced with the expert model [44]. As shown in Table 14, the models trained on datasets filtered by the self-model (OSPO) and by the expert model achieve nearly identical performance across all benchmarks. This suggests that using the model itself as the VQA-based reward estimator does not introduce meaningful confirmation bias during data filtering.

E.10. Internal Model Behavior

We analyze the model's internal behavior by examining its self-attention distribution over input text tokens. As shown

Table 14. Performance comparison of models trained with different VQA reward supervision on T2I-CompBench++, DPGBench, and GenEval. ↑ indicates higher is better, with bold highlighting the best score.

Model	Size	T2I-CompBench++↑				DPGBench↑	GenEval↑
		Attribute	Layout	Non-Spatial	Complex	Overall	Overall
Janus-Pro	7B	0.65	0.41	0.31	0.36	81.87	0.796
+ OSPO w/ self-reward (default)	7B	**0.756**	**0.447**	**0.316**	**0.415**	85.61	**0.830**
+ OSPO w/ expert-reward	7B	**0.756**	0.440	**0.316**	0.411	**85.84**	0.824

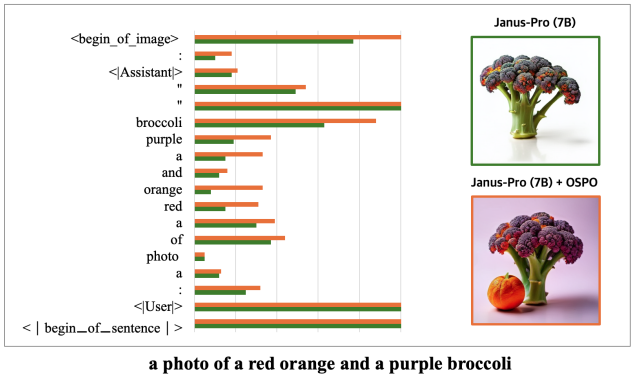


Figure 10. Comparison of globally averaged self-attention over text tokens during image-token generation for the baseline and OSPO-trained models. The baseline model fails to capture key semantic cues in the prompt, whereas the OSPO-trained model correctly reflects all intended semantics in the generated image.

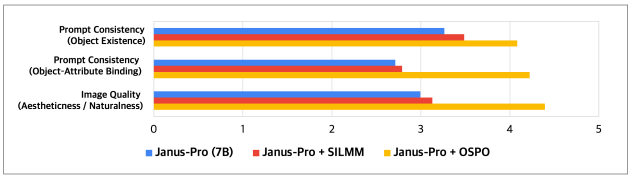


Figure 11. Human evaluation on 20 participants comparing images generated by the original Janus-Pro-7B, SILMM, and OSPO.

in Figure 10, OSPO increases attention on text tokens associated with key objects and attributes, indicating that the model becomes more grounded in the essential semantic elements of the prompt.

F. Human Evaluation

Settings. We conducted a user study to validate our model's compositional generation quality under human perception. A total of 27 questions were presented to 20 participants, each consisting of one input text prompt and three generated images produced from the same prompt by different models. Participants rated each image on a 5-point scale (1: very poor, 5: very good) across three criteria: **Prompt Consistency (Object Existence)**, which evaluates whether all objects mentioned in the prompt are present in the generated image; **Prompt Consistency (Object–Attribute Binding)**, which assesses whether object attributes such as color, shape, texture, and spatial relations are correctly reflected; and **Image Quality (Aestheticness / Naturalness)**, which measures overall visual quality including the absence of artifacts and naturalness of the output. The three compared models are the original Janus-Pro, SILMM, and OSPO.

Results. As shown in Figure 11, OSPO outperforms both the original model and SILMM across all three criteria. The most pronounced improvements are observed in the two prompt consistency criteria, demonstrating that OSPO substantially enhances the model's ability to generate images that faithfully reflect the semantics of the input prompt. OSPO also achieves the highest image quality score, indicating that improved compositional alignment does not come at the cost of visual naturalness.

G. Qualitative Evaluation

We present additional visual examples to qualitatively and intuitively compare each baseline (Janus-Pro-1B, Janus-Pro-7B, and UniTok-MLLM-8B) with the corresponding results after training by OSPO. Figures 12 and 13 present comparisons between Janus-Pro-7B and Janus-Pro-7B + OSPO on T2I-CompBench++, GenEval, and DPGBench. Figures 16 and 17 show comparisons between UniTok-MLLM-8B and UniTok-MLLM-8B + OSPO on the T2I-CompBench++, GenEval, and DPGBench as well.

As shown in the results, the baseline models Janus-Pro-1B, Janus-Pro-7B, and UniTok-MLLM-8B often fail to follow the prompt accurately, exhibiting object omissions, incorrect attribute bindings, and overall misalignment between the text prompt and the generated image. In contrast, training with OSPO framework leads to improved alignment, with models better preserving object presence and correctly binding attributes. These results highlight the effectiveness of OSPO in mitigating object-level hallucinations.

H. Prompt Data Examples

In this section, we present representative examples of data generated throughout the OSPO framework. Table 15 summarizes the category-specific base prompt formats and examples. Since the Non-spatial and Complex categories do

not follow a specific prompt format; therefore, formats for these are omitted in the table. Notably, our baseline SILMM [31] constructs its training data using the same base prompt format derived from T2I-Compbench++, ensuring a fair and directly comparable evaluation. Table 16 provides examples of perturbed prompts across the three perturbation types, and Table 17 provides the corresponding densified prompts derived from the Attribute example.

I. Prompt Templates

We leverage the In-Context Learning (ICL) [8] capabilities of MLLMs throughout the OSPO framework. Specifically, we first generate original prompts corresponding to four categories—Attribute, Layout, Non-Spatial, and Complex Composition. Using these original prompts, we construct different kinds of perturbed prompts, which are then used to produce dense prompts. During the Preference Pair Construction stage, we also generate VQA questions to perform Decompositional VQA, again leveraging the ICL capabilities of the MLLMs.

J. Future Works

Self-improving frameworks can be extended to more advanced capabilities, including self-reasoning and inference-time scaling. Recent studies [9, 30, 56] show that incorporating multimodal reasoning processes—such as self-refinement during image generation—can further improve T2I performance. As future work, we plan to integrate such reasoning mechanisms into OSPO to enhance its generation ability. Another promising direction is to extend OSPO beyond image generation by incorporating image understanding tasks, thereby moving toward a more unified and efficient self-improving framework for MLLMs.

Table 15. Generated base prompt formats and example prompts across Attribute, Layout, Non-spatial, and Complex categories.

Category	Prompt Format	Base Prompt
Attribute	a {adj} {obj} a {adj1} {obj1} and a {adj2} {obj2}	a green dress a fabric pillow and a ceramic cup
Layout	a {noun1} {spatial} a {noun2} {quantity} {obj(s)} {quantity} {obj1(s)} and {quantity} {obj2(s)}	a phone on the right side of a bag two trees four cookies and two umbrellas
Non-Spatial	-	A person is holding a book while sitting on a bench in a park.
Complex	-	The deep crimson, velvet curtains framed the large, cream-colored window and the luxurious, gold-trimmed mirrors reflected the warm, ambient lighting of the elegant room.

Table 16. Examples of perturbed prompts for the Replace, Swap, and Drop strategies. Underlined text marks the portion that differs semantically from the original prompt.

Category	Base Prompt	Perturbed Prompt
Attribute	a blue house and a purple cat	(Replace) a green house and a yellow cat (Swap) a purple house and a blue cat (Drop) a blue house
Layout	a book above a dog	(Replace) a book under a dog (Swap) a dog above a book (Drop) a dog and a book
Non-Spatial	A group of friends are playing soccer in a park on a sunny day.	(Replace) A group of friends are riding bicycles in a park on a sunny day. (Drop) A group of friends are playing in a park on a sunny day.
Complex	The vibrant, multicolored beads shimmered below the clear, polished glass vase.	(Replace) The dark, monochrome beads shimmered below the transparent, glossy vase. (Swap) The clear, polished glass vase shimmered below the vibrant, multicolored beads. (Drop) The plain, red, yellow beads are floating.

Table 17. Examples of densified prompts generated from the base and perturbed prompts. Underlined text highlights the differing semantic units.

Original Prompt	Densified Prompt
(Base) a blue house and a purple cat	A quaint, blue-painted house with a charming porch and a cheerful purple cat lounging on the wooden steps, watching a soft spring breeze rustle the nearby wildflowers.
(Perturbed) a purple house and a blue cat	A quaint, purple-painted house with a charming porch and a cheerful blue cat lounging on the wooden steps, watching a soft spring breeze rustle the nearby wildflowers.

Base Prompt Generation Template

System Prompt: You are an assistant dedicated to generating natural prompts that contain subjects and objects by using color words such as red, orange, yellow, green, blue, navy, purple. Do not repeat any previously used colors in subsequent prompts.

User: Generate a prompt that contains subjects and objects by using color words. Except for the color generated before, do not use the same color.
Assistant: Prompt: a blue bench and a green car.; Concepts and relations: blue bench, green car; Relation: (bench, blue), (car, green)

User: Generate a prompt that contains subjects and objects by using color words. Except for the color generated before, do not use the same color.
Assistant: Prompt: a red elephant and a white grass.; Concepts and relations: red elephant, white grass; Relation: (elephant, red), (grass, white)

User: Generate a prompt that contains subjects and objects by using color words. Except for the color generated before, do not use the same color.
Assistant: Prompt: a yellow pendant and an orange apple.; Concepts and relations: yellow pendant, orange apple; Relation: (pendant, yellow), (apple, orange)

User: Generate a prompt that contains subjects and objects by using color words. Except for the color generated before, do not use the same color.
Assistant:

Prompt Perturbation Template

System Prompt: You are a helpful assistant that generates object-level contrastive prompts by following four steps.
Step 1. Identify all objects in the prompt.
Step 2. Identify all object-attribute bindings in the prompt.
Step 3. Apply one of the following transformations to the given prompt: (1) 'replace' an attribute or object with another plausible one, (2) 'swap' two attributes or objects, or (3) 'drop' an attribute or object.
Step 4. Generate contrastive prompts based on the transformations.
Always ensure the output has opposing or conflicting semantics compared to the input. Do not just reorder or repeat the input.

User: a blue bench and a red car
Generate object-level contrastive prompts by applying the following transformation: 'replace'. Let's think step by step.
Assistant: Step 1. Objects: bench, car
Step 2. Attribute Bindings: ['blue', 'bench'], ['red', 'car']
Step 3. Transformation: 'replace'; Transformed Attributes: ['white', 'bench'], ['green', 'car']
Step 4. Contrastive Prompt: a white bench and a green car

User: a rubber tire and a wooden desk
Generate object-level contrastive prompts by applying the following transformation: 'swap'. Let's think step by step.
Assistant: Step 1. Objects: tire, desk
Step 2. Attribute Bindings: ['rubber', 'tire'], ['wooden', 'desk']
Step 3. Transformation: 'swap'; Transformed Attributes: ['wooden', 'tire'], ['rubber', 'desk']
Step 4. Contrastive Prompt: a wooden tire and a rubber desk

User: a oblong sweet potato and teardrop mango
Generate object-level contrastive prompts by applying the following transformation: 'drop'. Avoid changing the original attribute bindings of the objects that are not dropped. Let's think step by step.
Assistant: Step 1. Objects: sweet potato, mango
Step 2. Attributes Bindings: ['oblong', 'sweet potato'], ['teardrop', 'mango']
Step 3. Transformation: 'drop'; Transformed Attributes: ['teardrop', 'mango']
Step 4. Contrastive Prompt: a teardrop mango

User: `{input_prompt}`
Generate object-level contrastive prompts by applying the following transformation: `{perturbation_type}`. Let's think step by step.
Assistant:

Prompt Densification Template

System Prompt: You are an expert prompt engineer for text-to-image models. Your job is to take short and vague prompts and expand them into detailed, descriptive, and unambiguous prompts suitable for high-quality image generation.

Focus on using full sentences and include visual attributes such as objects, colors, texture, shape.
Avoid abstract or subjective words and instead use concrete and visual language.
Do not invent unrelated concepts; only expand and clarify the given prompt.
Follow these steps:
Step 1. Extract all objects and their visual attributes from Prompt 1.
Step 2. For Prompt 1, write a long, rich description that includes all identified objects and attributes from Step 1.
Step 3. Extract all objects and their visual attributes from Prompt 2.
Step 4. For Prompt 2, write a long, rich description that includes all identified objects and attributes from Step 3.
Ensure both outputs share a similar global context or scene.

User: Prompt 1: A large watermelon
Prompt 2: A small watermelon
Generate dense, detailed prompts. Ensure both outputs share a similar global context or scene but have different object-attribute bindings. Let's think step by step.
Assistant:
Step 1. Prompt 1 Object Bindings: watermelon-['large']
Step 2. Prompt 1 Dense: A large, ripe watermelon with deep green rinds and faint striping rests heavily in a handwoven wicker basket placed on the grass under dappled sunlight beside a weathered garden shed.
Step 3. Prompt 2 Object Bindings: watermelon-['small']
Step 4. Prompt 2 Dense: A small, round watermelon with bright green skin and subtle mottling sits neatly in a handwoven wicker basket placed on the grass under dappled sunlight beside a weathered garden shed.

User: Prompt 1: A peach tree and a square computer keyboard
Prompt 2: An orange tree and a square keyboard
Generate dense, detailed prompts. Ensure both outputs share a similar global context or scene but have different object-attribute bindings. Let's think step by step.
Assistant:
Step 1. Prompt 1 Object Bindings: tree-['peach']; keyboard-['square', 'computer']
Step 2. Prompt 1 Dense: A peach tree in full bloom stands beside a sleek, square-shaped computer keyboard resting on a polished wooden desk, with soft morning sunlight streaming through the windows of a quiet home office.
Step 3. Prompt 2 Object Bindings: tree-['orange']; keyboard-['square']
Step 4. Prompt 2 Dense: An orange tree with lush green leaves and dangling fruit stands beside a sleek, square-shaped computer keyboard resting on a polished wooden desk, with soft morning sunlight streaming through the windows of a quiet home office.

User: Prompt 1: A long in length elephant and a teal owl
Prompt 2: A short elephant and a blue owl
Generate dense, detailed prompts. Let's think step by step.
Assistant:
Step 1. Prompt 1 Object Bindings: elephant-['long in length']; owl-['teal']
Step 2. Prompt 1 Dense: A long-bodied elephant with weathered gray skin trudging slowly through a misty jungle path, accompanied by a serene teal owl perched on a mossy branch overhead, its feathers glimmering in the filtered morning light.
Step 3. Prompt 2 Object Bindings: elephant-['short']; owl-['blue']
Step 4. Prompt 2 Dense: A short, stocky elephant with wrinkled gray skin navigating a misty jungle path, accompanied by a curious blue owl perched on a mossy branch overhead, its feathers catching flecks of golden morning light.

User: Prompt 1: `{base_input_prompt}`
Prompt 2: `{negative_input_prompt}`
Generate dense, detailed prompts. Ensure both outputs share a similar global context or scene but have different object-attribute bindings. Let's think step by step.
Assistant:

Decompositional Question Generation Template

System Prompt: You are an assistant dedicated to transforming a sentence into several questions. You should first divide it into simple concepts and relations, and then provide the corresponding questions. Avoid using pronouns, such as he, she, it, and they.

User: A white harp and a rust soup.
Assistant: Concepts and relations: a white harp, a rust soup; Questions: Is there a white harp? Is there a rust soup?

User: A quarter circle lily and a hexagon mirror.
Assistant: Concepts and relations: a quarter circle lily, a hexagon mirror; Questions: Is there a quarter-circle lily? Is there a hexagon mirror?

User: Shiny mop and metal key holder.
Assistant: Concepts and relations: a shiny mop, a metal key holder; Questions: Is there a shiny mop? Is there a metal key holder?

User: `{input_prompt}`
Assistant:

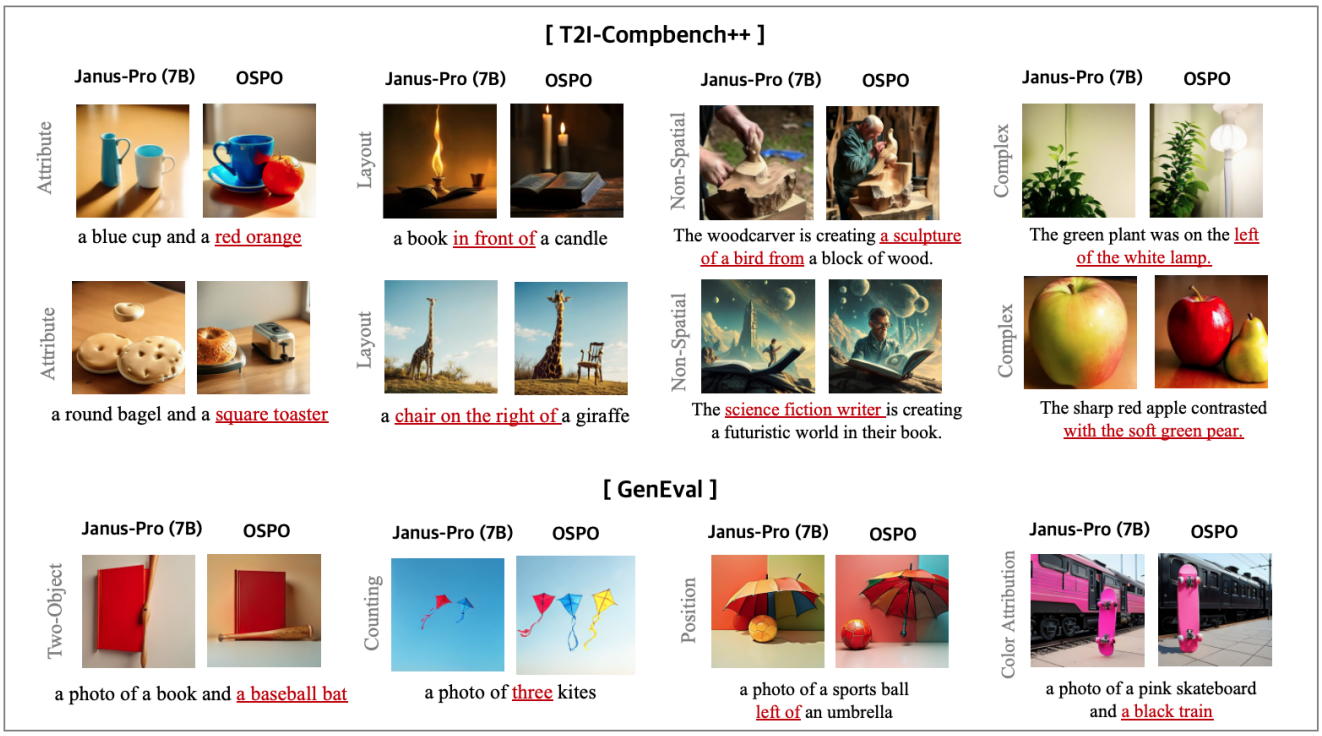


Figure 12. Additional Qualitative results of Janus-Pro-7B [3] and Janus-Pro-7B + OSPO on T2I-CompBench++ [19] and GenEval [12].

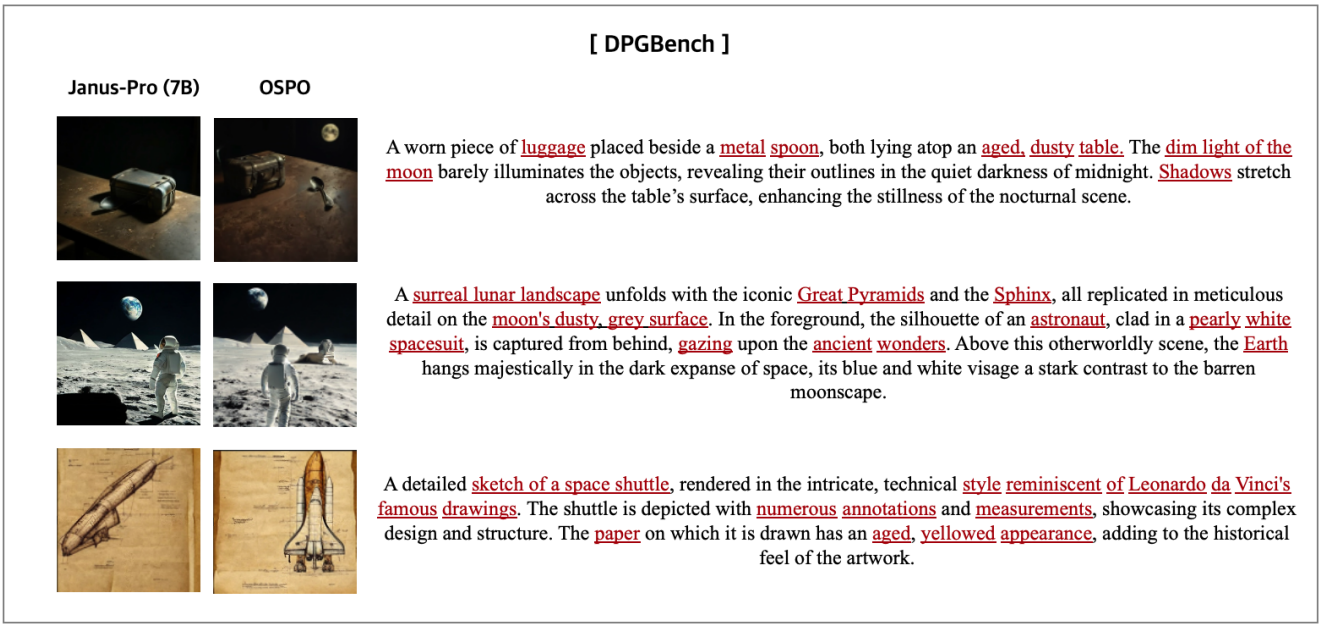


Figure 13. Additional Qualitative results of Janus-Pro-7B [3] and Janus-Pro-7B + OSPO on DPGBench [17].

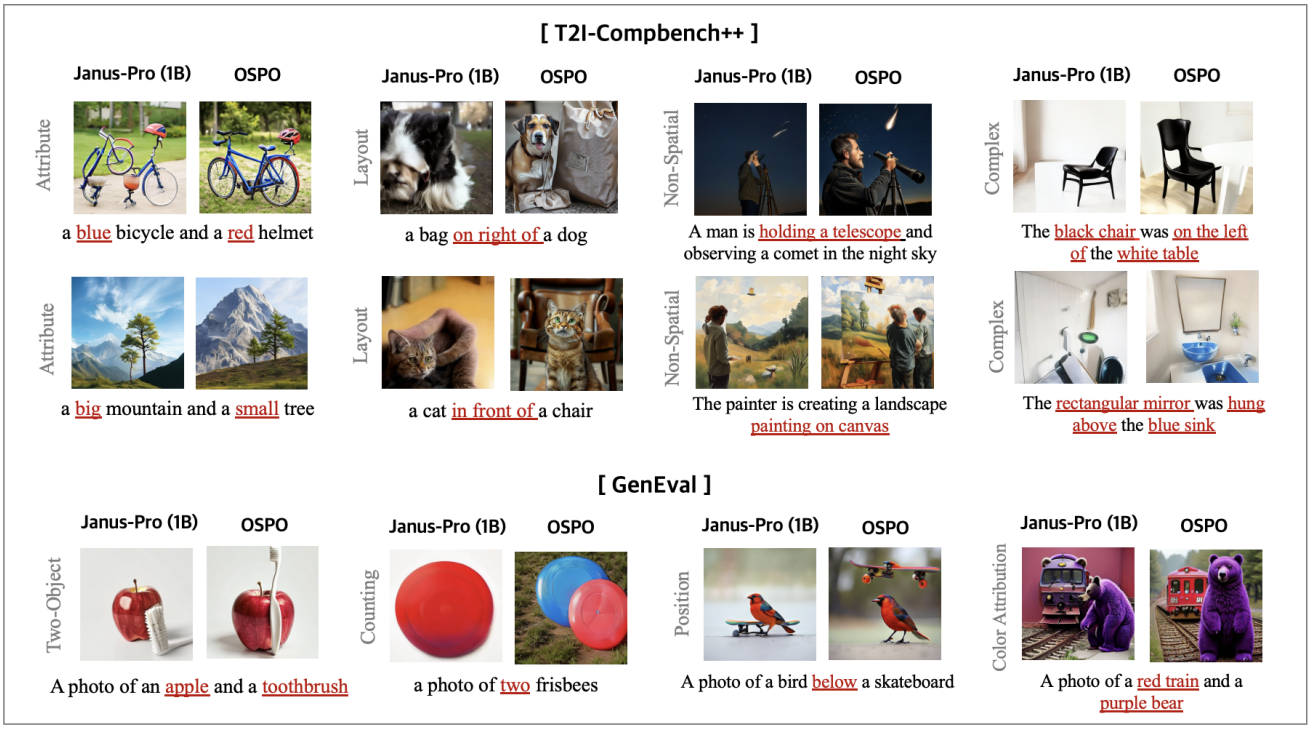


Figure 14. Additional Qualitative results of Janus-Pro-1B [3] and Janus-Pro-1B + OSPO on T2I-CompBench++ [19] and GenEval [12].

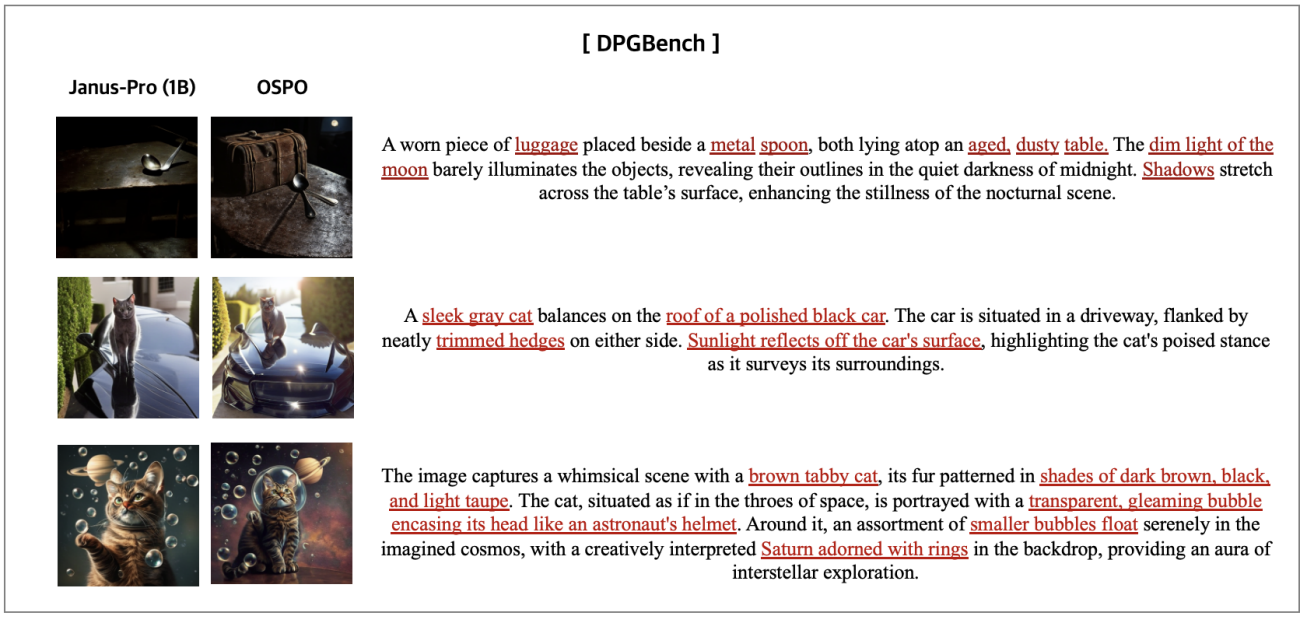


Figure 15. Additional Qualitative results of Janus-Pro-1B [3] and Janus-Pro-1B + OSPO on DPGBench [17].

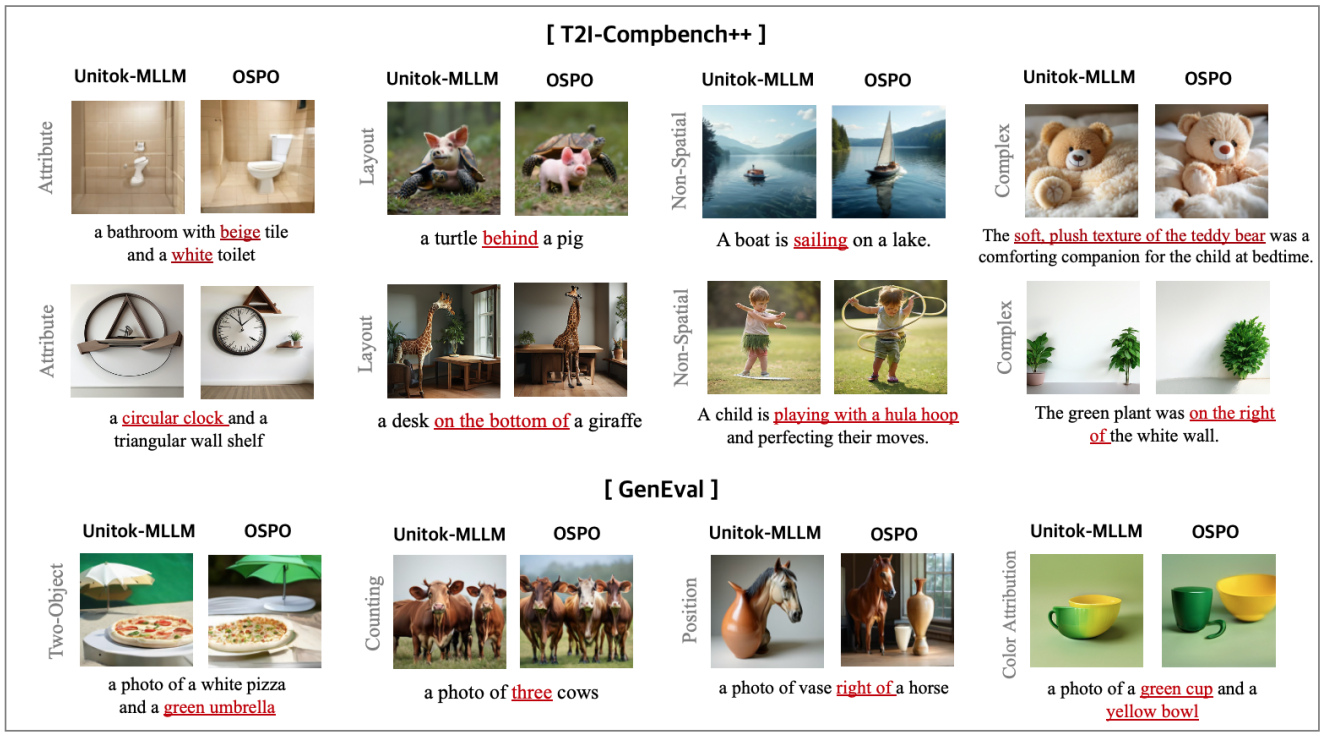


Figure 16. Additional Qualitative results of UniTok-MLLM-8B [25] and UniTok-MLLM-8B + OSPO on T2I-CompBench++ [19] and GenEval [12].

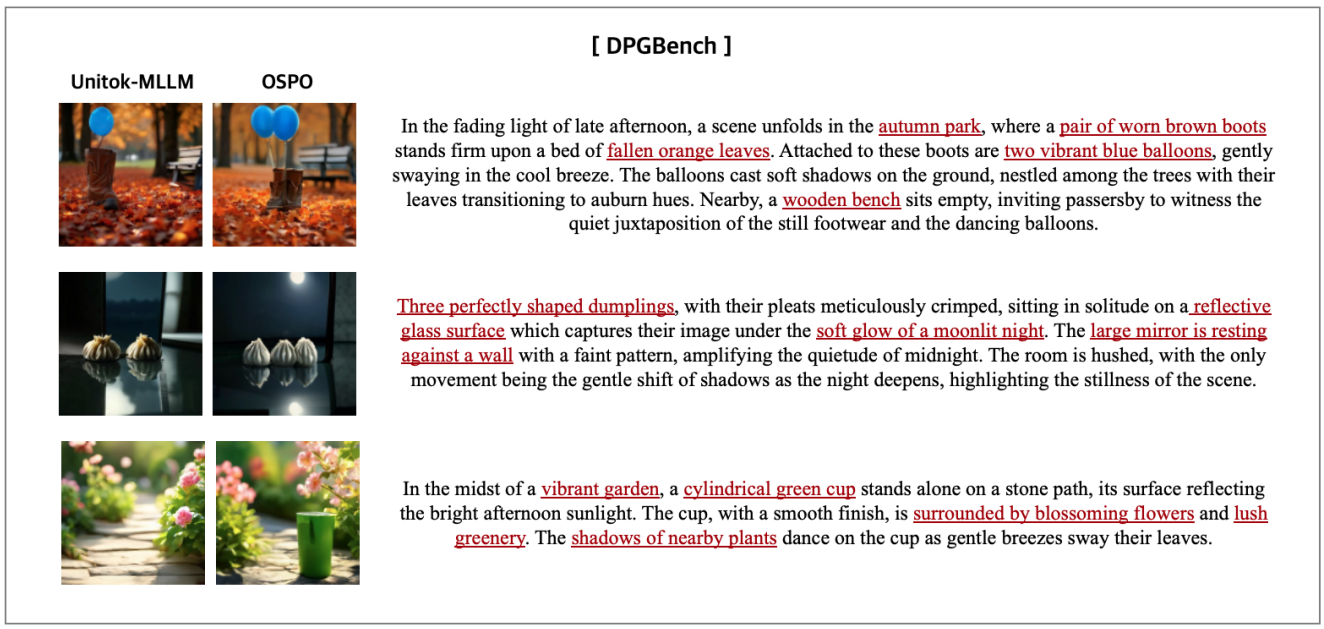


Figure 17. Additional Qualitative results of UniTok-MLLM-8B [25] and UniTok-MLLM-8B + OSPO on DPGBench [17].

References

[1] Lital Binyamin, Yoad Tewel, Hilit Segev, Eran Hirsch, Royi Rassin, and Gal Chechik. Make it count: Text-to-image generation with an accurate number of objects. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 13242–13251, 2025. 4

[2] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. 6, 4

[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. 2, 5, 13, 14

[4] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24185–24198, 2024. 2

[5] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299, 2022. 3

[6] Jaemin Cho, Yushi Hu, Roopal Garg, Peter Anderson, Ranjay Krishna, Jason Baldridge, Mohit Bansal, Jordi Pont-Tuset, and Su Wang. Davidsonian scene graph: Improving reliability in fine-grained evaluation for text-to-image generation. *arXiv preprint arXiv:2310.18235*, 2023. 4, 3

[7] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in neural information processing systems*, 36:49250–49267, 2023. 2

[8] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, et al. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*, 2022. 8

[9] Chengqi Duan, Rongyao Fang, Yuqing Wang, Kun Wang, Linjiang Huang, Xingyu Zeng, Hongsheng Li, and Xihui Liu. Got-r1: Unleashing reasoning capability of mllm for visual generation with reinforcement learning. *arXiv preprint arXiv:2505.17022*, 2025. 8

[10] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021. 2

[11] Yuying Ge, Sijie Zhao, Ziyun Zeng, Yixiao Ge, Chen Li, Xintao Wang, and Ying Shan. Making llama see and draw with seed tokenizer. *arXiv preprint arXiv:2310.01218*, 2023. 1, 2

[12] Dhruba 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. 4, 5, 6, 13, 14, 15

[13] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*, 2021. 6

[14] Jixiang Hong, Yiran Zhang, Guanzhong Wang, Yi Liu, Ji-Rong Wen, and Rui Yan. Suder: Self-improving unified large multimodal models for understanding and generation with dual self-rewards. 2025. 1, 2, 5, 8

[15] Cheng-Yu Hsieh, Jieyu Zhang, Zixian Ma, Aniruddha Kembhavi, and Ranjay Krishna. Sugarcrepe: Fixing hackable benchmarks for vision-language compositionality. *Advances in neural information processing systems*, 36:31096–31116, 2023. 4

[16] 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. 2

[17] Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. Ella: Equip diffusion models with llm for enhanced semantic alignment. *arXiv preprint arXiv:2403.05135*, 2024. 4, 5, 6, 13, 14, 15

[18] Yushi Hu, Benlin Liu, Jungo Kasai, Yizhong Wang, Mari Ostendorf, Ranjay Krishna, and Noah A Smith. Tifa: Accurate and interpretable text-to-image faithfulness evaluation with question answering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20406–20417, 2023.

[19] Kaiyi Huang, Chengqi Duan, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench++: An enhanced and comprehensive benchmark for compositional text-to-image generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025. 3, 4, 5, 6, 13, 14, 15

[20] 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. 8

[21] Seil Kang, Jinyeong Kim, Junhyeok Kim, and Seong Jae Hwang. Your large vision-language model only needs a few attention heads for visual grounding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 9339–9350, 2025. 4

[22] 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. 3

[23] WeiJie Li, Jin Wang, and Xuejie Zhang. Promptist: Automated prompt optimization for text-to-image synthesis. In *CCF international conference on natural language processing and Chinese computing*, pages 295–306. Springer, 2024. 4

[24] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023. 2

[25] Chuofan Ma, Yi Jiang, Junfeng Wu, Jihan Yang, Xin Yu, Zehuan Yuan, Bingyue Peng, and Xiaojuan Qi. Unitok: A unified tokenizer for visual generation and understanding. *arXiv preprint arXiv:2502.20321*, 2025. 2, 4, 15

[26] Weijia Mao, Zhenheng Yang, and Mike Zheng Shou. Unirl: Self-improving unified multimodal models via su-

pervised and reinforcement learning. *arXiv preprint arXiv:2505.23380*, 2025. 1, 2

[27] Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37: 124198–124235, 2024. 5

[28] Nobuyuki Otsu et al. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27, 1975. 4, 1

[29] Kaihang Pan, Wendong Bu, Yuruo Wu, Yang Wu, Kai Shen, Yunfei Li, Hang Zhao, Juncheng Li, Siliang Tang, and Yueting Zhuang. Focusdiff: Advancing fine-grained text-image alignment for autoregressive visual generation through rl. *arXiv preprint arXiv:2506.05501*, 2025. 8, 4

[30] Kaihang Pan, Yang Wu, Wendong Bu, Kai Shen, Juncheng Li, Yingting Wang, Yunfei Li, Siliang Tang, Jun Xiao, Fei Wu, Hang Zhao, and Yueting Zhuang. Janus-pro-r1: Advancing collaborative visual comprehension and generation via reinforcement learning. In *Proceedings of the 39th Annual Conference on Neural Information Processing Systems (NeurIPS 2025)*, 2025. Poster. 8

[31] Leigang Qu, Haochuan Li, Wenjie Wang, Xiang Liu, Juncheng Li, Liqiang Nie, and Tat-Seng Chua. Silmm: Self-improving large multimodal models for compositional text-to-image generation. *arXiv preprint arXiv:2412.05818*, 2024. 1, 2, 5, 8

[32] 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. 6, 3, 4

[33] 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. 1, 5, 6

[34] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021. 3

[35] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024. 3

[36] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 1

[37] 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. 1, 8

[38] Quan Sun, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, Yueze Wang, Hongcheng Gao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Emu: Generative pretraining in multimodality. *arXiv preprint arXiv:2307.05222*, 2023. 1, 2

[39] Quan Sun, Yufeng Cui, Xiaosong Zhang, Fan Zhang, Qiying Yu, Yueze Wang, Yongming Rao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Generative multimodal models are in-context learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14398–14409, 2024. 2

[40] Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024. 1

[41] Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing vision and language models for visio-linguistic compositionality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5238–5248, 2022. 4

[42] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 2

[43] Fei Wang, Wenxuan Zhou, James Y Huang, Nan Xu, Sheng Zhang, Hoifung Poon, and Muhao Chen. mdpo: Conditional preference optimization for multimodal large language models. *arXiv preprint arXiv:2406.11839*, 2024. 1

[44] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025. 5, 6

[45] Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024. 1, 2

[46] Zhijie Wang, Yuheng Huang, Da Song, Lei Ma, and Tianyi Zhang. Promptcharm: Text-to-image generation through multi-modal prompting and refinement. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pages 1–21, 2024. 4

[47] Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, et al. Janus: Decoupling visual encoding for unified multimodal understanding and generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12966–12977, 2025. 2

[48] Junkang Wu, Yuexiang Xie, Zhengyi Yang, Jiancan Wu, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. β-dpo: Direct preference optimization with dynamic β. *Advances in Neural Information Processing Systems*, 37: 129944–129966, 2024. 5

[49] Yecheng Wu, Zhuoyang Zhang, Junyu Chen, Haotian Tang, Dacheng Li, Yunhao Fang, Ligeng Zhu, Enze Xie, Hongxu Yin, Li Yi, et al. Vila-u: a unified foundation model integrating visual understanding and generation. *arXiv preprint arXiv:2409.04429*, 2024. 1, 2

[50] 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.

[51] Jinheng Xie, Zhenheng Yang, and Mike Zheng Shou. Show-o2: Improved native unified multimodal models. *arXiv preprint arXiv:2506.15564*, 2025. 1

[52] Hongji Yang, Yucheng Zhou, Wencheng Han, and Jianbing Shen. Self-rewarding large vision-language models for opti-

mizing prompts in text-to-image generation. *arXiv preprint arXiv:2505.16763*, 2025. 4

[53] Zhengyuan Yang, Jianfeng Wang, Linjie Li, Kevin Lin, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. Idea2img: Iterative self-refinement with gpt-4v (ision) for automatic image design and generation. *arXiv preprint arXiv:2310.08541*, 2023. 4

[54] Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*, 2023. 2, 3

[55] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 4

[56] Yu Zhang, Yunqi Li, Yifan Yang, Rui Wang, Yuqing Yang, Dai Qi, Jianmin Bao, Dongdong Chen, Chong Luo, and Lili Qiu. Reasongen-r1: Cot for autoregressive image generation models through sft and rl. *arXiv preprint arXiv:2505.24875*, 2025. 8

[57] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Simple multi-dataset detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7571–7580, 2022. 3

[58] Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. Texygen: A benchmarking platform for text generation models. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 1097–1100, 2018. 3