# *UniGen*: Enhanced Training & Test-Time Strategies for Unified Multimodal Understanding and Generation

**Rui Tian**[1 2 ∘], **Mingfei Gao**[2 ∘], **Mingze Xu**[2 ∘],
**Jiaming Hu**[2], **Jiasen Lu**[2], **Zuxuan Wu**[1 †], **Yinfei Yang**[2], **Afshin Dehghan**[2 †]
[1]Institute of Trustworthy Embodied AI, Fudan University [2]Apple
{mgao22,mingze_xu2,adehghan}@apple.com, {rtian23,zxwu}@fudan.edu.cn,
∘First authors; †Corresponding authors

## Abstract

We introduce ***UniGen***, a unified multimodal large language model (MLLM) capable of image understanding and generation. We study the full training pipeline of *UniGen* from a data-centric perspective, including multi-stage pre-training, supervised fine-tuning, and direct preference optimization. More importantly, we propose a new ***Chain-of-Thought Verification* (*CoT-V*)** strategy for test-time scaling, which significantly boosts *UniGen*'s image generation quality using a simple *Best-of-N* test-time strategy. Specifically, *CoT-V* enables *UniGen* to act as both image generator and verifier at test time, assessing the semantic alignment between a text prompt and its generated image in a step-by-step CoT manner. Trained entirely on open-source datasets across all stages, *UniGen* achieves state-of-the-art performance on a range of image understanding and generation benchmarks, with a final score of 0.78 on GenEval and 85.19 on DPG-Bench. Through extensive ablation studies, our work provides actionable insights and addresses key challenges in the full life cycle of building unified MLLMs, contributing meaningful directions to future research. Code is available at `https://github.com/apple/ml-unigen`.

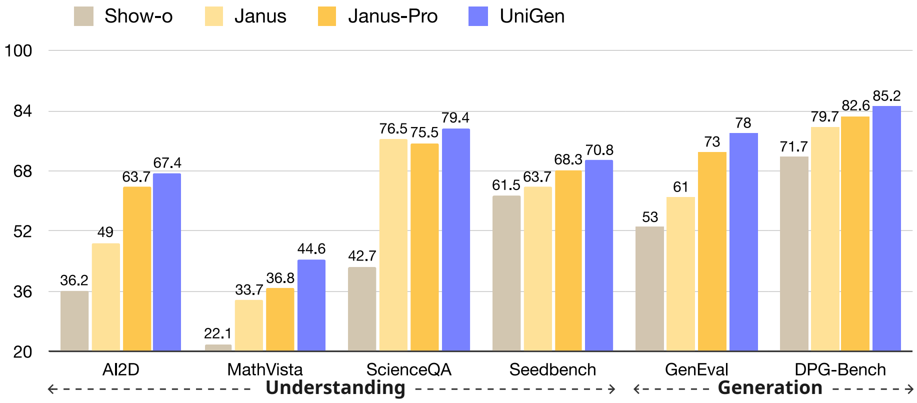


Figure 1: **Comparison against state-of-the-art unified MLLMs.** *UniGen-1.5B* outperforms Show-o-1.3B, Janus-1.3B and Janus-Pro-1.5B across understanding and generation benchmarks.

## 1 Introduction

Unifying understanding and generation within a single framework represents a key step toward general-purpose artificial intelligence models [53]. Pioneering work [9, 16, 67, 83, 84, 87, 96] has made encouraging progress but relies on distinct training recipes and internal datasets. More

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

importantly, they have yet to demonstrate good practice in wisely collaborating these two capabilities within a unified architecture to achieve substantial performance gains. We advance the development of unified multimodal large language models (MLLMs) by carefully studying the impact of their training recipes across different stages and proposing optimizations to improve both image understanding and generation. We further explore leveraging test-time interaction between understanding and generation tasks, selecting images with higher quality by using our unified MLLM as the self-verifier.

Specifically, we introduce ***UniGen***, a unified MLLM for image understanding and generation. To shed light on the impact of different training stages, we walk through the entire life cycle of the model development, including multi-stage pretraining, supervised fine-tuning [40, 54, 56], and direct preference optimization [60, 78]. We ablate the impact of each training stage and their design choices from a data-centric perspective, and draw insightful lessons for building advanced unified MLLMs. Unlike state-of-the-art models [9, 47, 87, 84] that rely on large-scale internal datasets, we curate new data mixtures across training stages by using only open-source images. We show that models trained on publicly available data can also achieve competitive results.

To further enhance image generation quality, we propose a new ***Chain-of-Thought Verification*** **(*CoT-V*)** strategy for test-time scaling. The key idea is to leverage *UniGen*'s inherent understanding ability as a self-verifier to assess the quality of its own generated images. Specifically, during inference, *UniGen* produces $N$ images for a given text prompt, while *CoT-V* progressively evaluates semantic coherence between each image-text pair and selects the best. With only lightweight fine-tuning (*e.g.*, 500 training steps), *UniGen* is able to achieve the reasoning capability, thinking step-by-step to verify each atomic fact according to the prompt and each generated image. Importantly, this CoT verification seamlessly enhances *UniGen*'s image generation quality while preserving its general understanding performance. In this way, we collaborate the understanding and generation capabilities within a unified MLLM, substantially boosting the text-to-image generation quality using a simple *Best-of-N* strategy [79, 106] and self-verification [82, 7, 24]. Our experiments show that *UniGen*'s performance is consistently improved across various image generation benchmarks.

We evaluate *UniGen* on various understanding and generation tasks, as shown in Fig. 1. For image understanding, *UniGen* outperforms comparable unified MLLMs (*e.g.* Show-o [87] and Janus-Pro [9]) across benchmarks and even ties with some strong understanding specialist models, such as LLaVA-OV [32] and MM1.5 [102], as displayed in Table 1. For text-to-image generation, *UniGen* obtains 0.78 on GenEval and 85.19 on DPG-Bench using only open-source data, surpassing state-of-the-art unified MLLMs [87, 84, 9] by a clear margin.

## 2 Related Work

**Multimodal Large Language Models (MLLMs)** have advanced significantly in image [1, 11, 40, 41, 49, 77, 106] and video understanding [42, 103, 89, 90, 101, 108]. Their architecture typically consists of a vision encoder [59, 100, 72] to extract visual features, a projector [34, 2] to align image-text embeddings, and a large language model (LLM) [1, 71, 92, 10] to generate responses. Early work focuses on pre-training using large-scale vision-language corpus [48, 69], then moves to carefully curated instructional datasets for supervised fine-tuning [32, 102] and reinforcement learning [27, 60]. Recently, enabling MLLMs to output explicit reasoning trajectories has become a promising research direction [52, 19, 68]. They explore strategies, such as chain-of-thought (CoT) prompting [13, 95], reinforcement learning [66, 88], and test-time scaling [106, 79] to enhance the visual reasoning capabilities of MLLMs.

**Unified Understanding and Generation** aims to combine visual understanding and generation within a single MLLM framework [51, 67, 58, 81, 44, 43, 8, 12, 35]. This is often achieved by jointly optimizing LLMs with multimodal objectives and generation-specific losses, such as autoregressive decoding [84], diffusion [105], flow-matching [47], and masked image prediction [94, 87]. Visual tokenizers [14, 45, 50, 73, 75, 98, 104] are critical for enabling both semantic understanding and high-fidelity generation. Recent efforts explore both decoupled encoders [70, 84] and unified tokenizers [29, 58, 85] for better task balancing. Integrating CoT into visual generation emerges as a promising strategy. PARM [20] scales test-time computation by introducing a new verification process. MINT [81], ImageGen-CoT [38], and Got [15] leverage multimodal reasoning to perform prompt planning, generation, reflection, and refinement. Despite these advances, using chain-of-thought for unified understanding and generation remains underexplored. In this work, *UniGen* incorporates a CoT-based self-verification strategy via Best-of-N selection during test-time scaling, which leads to substantial improvements in image generation performance.

# 3 Recipe for Building *UniGen*

## 3.1 Architecture

As shown in Fig. 2, we unify the image understanding and generation tasks into a pretrained LLM. Motivated by prior work [84], we separate visual encoding for understanding and generation into continuous and discrete embedding spaces, respectively.

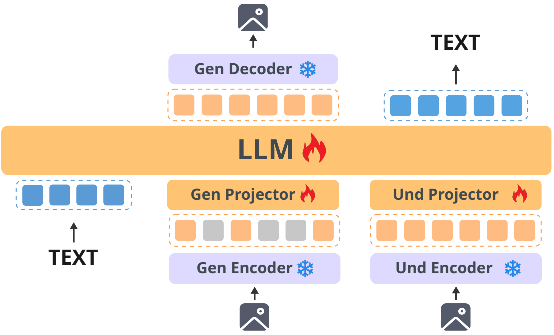


Figure 2: **The architecture of *UniGen***, which is based on an autoregressive LLM and decoupled vision encoders for image understanding and generation tasks.

**For image understanding**, we follow the LLaVA [40] workflow and adopt the next-token prediction paradigm. Given an input image $X^U$, the understanding encoder $\mathbf{Enc}^U$ (*e.g.*, SigLIP [100]) extracts its feature as a vector of continuous tokens $\mathcal{X}^U = \mathbf{Enc}^U(X^U)$. The projector $\mathbf{P}^U$ aligns the image and text embeddings into the same space, then the embeddings are fed into LLM as inputs. We compute the understanding loss using the vanilla autoregressive training objective $\mathcal{L}_{und}$. To preserve the LLM's language modeling capability, we also train *UniGen* with text-only data and backpropagate the corresponding loss $\mathcal{L}_{text}$.

**For text-to-image generation**, we employ the masked token prediction [5] as our training objectives. Unlike the autoregressive decoding for text tokens, this paradigm enables models to generate multiple image tokens in parallel, significantly accelerating the generation process. *During training*, for each image $X^G \in \mathbb{R}^{H \times W}$, the generation encoder $\mathbf{Enc}^G$ (*e.g.*, MAGVIT-v2 [97]) tokenizes it into a sequence of discrete tokens $\mathcal{X}^G$ of length $N = H/d_s \cdot W/d_s$, where $d_s$ refers to the spatial downsampling factor of $\mathbf{Enc}^G$. Then, given a masking ratio $\eta$ according to the scheduling function $\gamma(\cdot)$, we randomly sample a binary mask $\mathcal{M}(\eta) = [m_0, \cdots, m_{N-1}]$, where $\eta * N$ positions are uniformly set to $1$ and others are set to $0$. For each position $i$ where $m_i$ equals to $1$, we replace its corresponding discrete image token $\mathcal{X}_i^G$ with a special mask token [MASK] to form the final input image sequence. Finally, we prepend the textual tokens (*e.g.*, image classes or captions) with the masked sequence $\mathcal{X}^{\mathcal{M}}$. *During inference*, the image generation starts with all masked tokens $\mathcal{X}^{\mathcal{M}} = [\texttt{[MASK]}, \cdots, \texttt{[MASK]}]$, and gradually fills up the latent representation with scattered predictions in parallel.

## 3.2 Pre-Training (PT)

The goal of pre-training is to develop *UniGen*'s visual generation capability while preserving its potential for multimodal understanding. Thus, we only optimize the generation projector and the LLM with other parameters frozen. We also include image-to-text and text-only pre-training to keep *UniGen*'s language modeling capability. To encourage a better alignment between discrete image tokens and the text, we directly use the generation encoder for understanding tasks *but only* in this stage. We empirically find that this design can significantly improve the image generation performance. Specifically, we employ an "easy-to-difficult" strategy through a two-stage process.

**Pre-training Data.** We generate fine-grained captions for images from ImageNet [62] , CC-3M [63], CC-12M [6] and SAM-11M [31] dataset using Qwen2.5-VL-7B [3] to form a 40M image-text pair corpus. For text-only pre-training, we use RefinedWeb [55].

**PT-1 Stage** seeks to align the image and text embeddings and predict the distribution of basic visual concepts. Similar to prior works [84], we employ ImageNet for generation pre-training warmup and leverage the full 40M image-text pairs for the understanding task. However, we propose that *using image captions, rather than image categories, for text-to-image generation leads to better convergence*.

**PT-2 Stage** further facilitates *UniGen* to generalize to wider visual generation capabilities. We expand the text-to-image training dataset to the full 40M image-text pairs, while using the same image-to-text and text-only ones. We argue that *training data with a richer distribution enables more accurate control over generation patterns*. We name the model trained in this stage as ***UniGen-PT***.

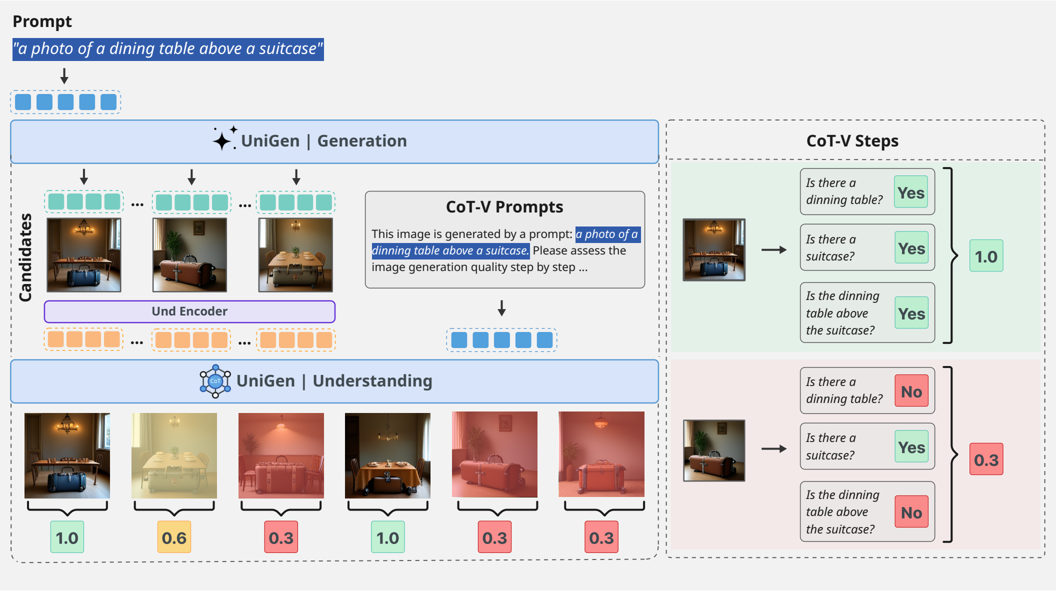

Prompt
"a photo of a dining table above a suitcase"
UniGen | Generation
Candidates
Und Encoder
CoT-V Prompts
This image is generated by a prompt: a photo of a dinning table above a suitcase. Please assess the image generation quality step by step ...
UniGen | Understanding
1.0
0.6
0.3
1.0
0.3
0.3
CoT-V Steps
Is there a dinning table?
Yes
Is there a suitcase?
Yes
Is the dinning table above the suitcase?
Yes
1.0
Is there a dinning table?
No
Is there a suitcase?
Yes
Is the dinning table above the suitcase?
No
0.3

Figure 3: **The workflow of *UniGen* using test-time scaling and *CoT-V*. Left:** Illustration of *Best-of-N* selection with *CoT-V* with $N = 6$. **Right:** Visualization of the step-by-step reasoning process in *CoT-V* for computing the final quality score.

### 3.3 Supervised Fine-Tuning (SFT)

In the SFT stage, *UniGen* is jointly trained on the image understanding and generation tasks. We fine-tune the generation projectors, understanding projectors, and the LLM, while still keeping the vision encoders frozen. **For image understanding**, we notice that the knowledge-centric understanding is limited during pre-training stages. To enhance related capabilities, we adopt the strong image mixture from SlowFast-LLaVA-1.5 [90], which was carefully curated from open-source datasets with 4.67M multimodal VQA samples. **For image generation**, prior work [9] uses high-quality synthetic data that can enable fast and robust training convergence. We share this observation by using the JourneyDB [64] and text-2-image-2M [28] to improve the aesthetic quality of our generated images. We name the model trained in this stage as ***UniGen-SFT***.

### 3.4 Direct Preference Optimization (DPO)

We further enhance *UniGen* by aligning its outputs with human preference through DPO. We first discuss how we construct our synthetic preference dataset, then describe our DPO algorithm.

**Preference Dataset.** We leverage *UniGen-SFT* to generate the images for our preference dataset. For a given prompt, 20 images are generated. A preferred and rejected sample pair is constructed by evaluating the coherence between each image and the prompt. To improve the data robustness, we collect 6k short prompts from PARM [20], 6k medium-length prompts from T2I-Comp [25] training set, and 6k long prompts from re-annotated SA1B to generate training image candidates.

For short prompts, we use the GenEval metrics to evaluate the generation quality. For prompts of medium or long lengths, we decompose each prompt into fine-grained visual questions with Qwen2.5-7B. Then, we assess image-prompt consistency by feeding Qwen2.5VL-7B with the image-question pair. An output “yes” indicates the image aligns with the description, and “no” otherwise. The final consistency score $S$ is averaged from these answers. For each prompt, we sample one highest-scored example as the preferred image and the lowest one as the rejected image. Prompts with no clear preference are filtered out. Finally, we obtain around 13k triplets for training.

**DPO Training.** We adopt the vanilla DPO training loss and freeze the understanding encoder and projector in this stage. The training ends in one epoch with a batch size of 64 and a learning rate of $1e^{-5}$. We empirically find that *this DPO training does not impair UniGen’s understanding performance.* We name the model trained in this stage as ***UniGen-DPO***.

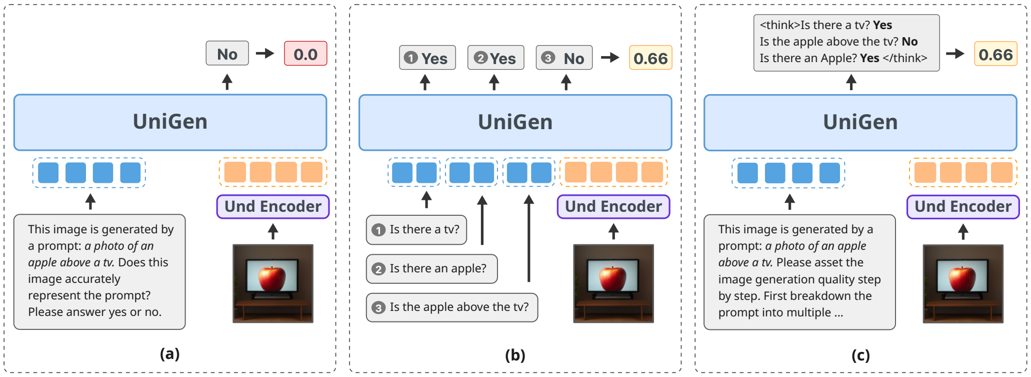

No
0.0
UniGen
Und Encoder
This image is generated by a prompt: *a photo of an apple above a tv.* Does this image accurately represent the prompt? Please answer yes or no.
(a)
1 Yes
2 Yes
3 No
0.66
UniGen
Und Encoder
1 Is there a tv?
2 Is there an apple?
3 Is the apple above the tv?
(b)
<think>Is there a tv? **Yes**
Is the apple above the tv? **No**
Is there an Apple? **Yes** </think>
0.66
UniGen
Und Encoder
This image is generated by a prompt: *a photo of an apple above a tv.* Please asset the image generation quality step by step. First breakdown the prompt into multiple ...
(c)

Figure 4: **An example of using different image verification methods**: **(a)** Outcome Verification, **(b)** Rule-based Verification and **(c)** Chain-of-Thought Verification.

### 3.5 Test-Time Scaling

Recent studies have shown the effectiveness of test-time scaling on improving both image understanding [79, 106] and generation [20, 86]. We employ the Best-of-N evaluation strategy and leverage *UniGen*'s understanding ability to conduct self-critique for image generation verification. The general workflow is illustrated in Fig. 3. *First*, *UniGen* generates $N$ candidate images for a given prompt. *Second*, we input each generated image along with its prompt into *UniGen*, which evaluates the alignment between the image and its textual description and outputs a quality score $\mathcal{S}$. *Third*, we return the image with the highest score. We propose three verification methods as shown in Fig. 4.

- **Outcome Verification (OV)** simply prompts *UniGen* to directly judge the coherence of the input prompt and each image candidate, giving a binary score (*i.e.* "yes" for a good match and "no" for a failure generation). We randomly select one if there are candidates with the same score.
- **Rule-based Verification (RV)** breaks down each prompt into several atomic questions based on pre-defined rules, then sequentially feeds them with the generated image into *UniGen* for quality verification. The results of all sub-questions are averaged as the final quality score.
- **Chain-of-Thought Verification (CoT-V)** instructs the model to think step-by-step and verifies each atomic fact according to the prompt and each generated image, following the CoT format: `<think_start>`$\texttt{Q}_1$`? `$\texttt{A}_1$`; ··· `$\texttt{Q}_\texttt{n}$`? `$\texttt{A}_\texttt{n}$`;<think_end>`. We compute the final quality score $\mathcal{S}$ by parsing the CoT outputs. Specifically, given a text prompt $T$ and a generated image $I$, *CoT-V* produces a list of visual questions $Q = \{Q_1, \cdots, Q_n\}$ and their corresponding answers $A = \{A_1, \cdots, A_n\}$. The final score $\mathcal{S}$ is computed by averaging the scores in the answer list.

OV relies on *UniGen*'s pattern-matching capabilities without intermediate reasoning. RV incorporates a rule-driven reasoning process into test-time scaling. Although effective on well-structured prompts, RV struggles with free-form or complex instructions, such as those in DPG-Bench [23]. *CoT-V* leverages the strengths of both approaches, enabling reasoning-driven image verification without the need for manual prompt decomposition. Thus, we use *CoT-V* as our default verification method.

#### 3.5.1 *CoT-V* Post-Training

*UniGen* has not been precisely trained to generate CoT responses. Here we introduce a lightweight post-training strategy upon *UniGen-DPO*, equipping it with the ability of CoT-based verification.

**Data.** To construct the *CoT-V* post-training data, we reuse the image-text pairs collected during the DPO stage (Sec. 3.4). For prompts sourced from PARM, we extract the question-answer pairs via rule-based matching, since they are built upon a clear structure [17]. For prompts from T2I-Comp that are more complicated, we first guide Qwen2.5-7B [91] to generate a series of atomic questions, then query Qwen2.5-7B-VL with each image-question pair to obtain their binary pseudo labels. We exclude the prompts from SA-1B due to the lower quality of the decomposed visual questions. We empirically find that most of the decomposed questions do not fully cover the visual concepts of the original caption. We totally sample 20K image-question-answer triplets from both prompt sources.

**Training.** We format the above 20K training pairs as instruction-following conversations, and feed them into *UniGen-DPO* for supervised fine-tuning. In this stage, we only optimize the understanding

Table 1: **Comparison with state-of-the-art models on image understanding benchmarks.** *denotes reproduced results and RW-QA denotes RealWorld-QA.

| **Model** | **#Params** | **Res.** | **AI2D** | **GQA** | **POPE** | **MMMU** | **MathVista** | **RW-QA** | **ScienceQA** | **Seedbench** |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | *Understanding MLLMs* | | | | |
| LLaVA-OV [32] | 0.5B | AnyRes | 57.1 | - | - | 31.4 | 34.8 | 55.6 | 67.2 | 65.5 |
| MM1.5 [102] | 1B | AnyRes | 59.3 | - | 88.1 | 35.8 | 37.2 | 53.3 | 82.1 | 70.2 |
| LLaVA 1.5 [39] | 7B | 336 | 55.1* | 62.0 | 86.1 | 36.3* | 26.7* | 55.8* | 66.8 | 66.1 |
| | | | | | | *Unified MLLMs* | | | | |
| Show-o [87] | 1.3B | 336 | 36.2* | 61.0* | 84.5 | 27.4 | 22.1* | 48.5* | 42.7* | 61.5* |
| Janus [84] | 1.3B | 384 | 49.0* | 59.1 | 87.0 | 30.5 | 33.7* | 48.4* | 76.5* | 63.7 |
| Janus-Pro [9] | 1.5B | 384 | 63.7* | 59.3 | 86.2 | **36.3** | 36.8* | 51.1* | 75.5* | 68.3 |
| Vila-U [85] | 7B | 384 | - | 60.8 | 85.8 | - | - | - | - | 56.3 |
| MMAR [93] | 7B | 256 | - | - | 83.0 | - | - | - | - | 64.5 |
| UniToken [29] | 7B | 384 | **68.7** | - | - | 32.8 | 38.5 | - | - | 69.9 |
| *UniGen* | 1.5B | 384 | 67.4 | **62.3** | **87.8** | 32.3 | **44.6** | **56.7** | **79.4** | **70.8** |

projector and the LLM. To ensure not impairing *UniGen*'s general understanding capabilities, we fine-tune *UniGen* on this *CoT-V* dataset for only 500 steps using a small learning rate of $1 \times 10^{-5}$. The model trained after this stage is our final model, and we name it as ***UniGen***.

# 4 Experiments

## 4.1 Implementation Details

We use 32 H100-80G GPUs for pre-training stages and 8 H100-80G GPUs for the others. *UniGen* is built upon the pre-trained Qwen2.5-1.5B [91]. We adopt MAGVITv2 from Show-o [87] as our discrete visual encoder with input resolution of 256 × 256 and SigLIP [100] as our continuous visual encoder. As discussed in Sec 3.1, we use MAGVITv2 for both understanding and generation in PT-1 and PT-2, and keep using SigLIP as the understanding encoder after SFT.

**Training.** We follow Show-o [87] to use a bidirectional attention mask within image tokens, but keep the causality within text tokens and between multimodal tokens. Detailed hyperparameters for each training stage are described in Appendix Table 17 with more details in Appendix Sec. E.0.2.

**Inference and Evaluation.** We follow the common practice of image generation to use classifier-free guidance [22] and set the scale to 5.0. In addition, we follow MaskGIT [5] to adopt the cosine masking scheduler in inference and set the default number of steps to $T = 50$. We use MAGVITv2 decoder to project the visual tokens back to the pixel space. For test-time scaling with *CoT-V*, we generate $N = 20$ image candidates per text prompt and select top-K ($K = 5$) out of them, sending for evaluation on GENEVAL and DPG-BENCH.

## 4.2 Main Results

We report the performance of *UniGen* on various benchmarks (details are discussed in Appendix Sec. A) and show qualitative results in Fig 5. We mainly compare *UniGen* with state-of-the-art unified LLMs in Table 1 and 2, but also reference strong specialist models to understand our position in the whole picture of MLLMs. Here we highlight the following observation.

**First, *UniGen* achieves state-of-the-art results across understanding benchmarks compared to existing unified MLLMs.** Specifically, *UniGen* outperforms Janus-Pro on RealWorld-QA, AI2D and MathVista by +5.6%, +3.7%, and +7.8%, respectively. We believe our improvements are mainly driven by using *(i)* the decoupled generation and understanding encoders and *(ii)* the stronger SFT data mixture. Notably, *UniGen* is even comparable with some strong understanding-only MLLMs, such as LLaVA-OV-0.5B and MM1.5-1B, even though they use much higher input resolutions.

**Second, *UniGen* significantly outperforms existing unified MLLMs and strong generation-only models on text-to-image benchmarks.** Using GENEVAL in Table 2 as an example, *UniGen* achieves the overall score of 0.78, significantly outperforming Janus-Pro by 0.05. Besides, our model demonstrates an overwhelming advantage on the "Counting" task by +0.27 higher than Janus-Pro. *UniGen* even beats a range of superior generation-only models (*e.g.*, outperforming DALLE-2, and Emu3 by +0.26, and +0.24, respectively), even though they are with much larger model sizes. Similarly, *UniGen* outperforms existing models by a clear margin on DPG-BENCH as shown in Table 2, outperforming Show-o and Janus-Pro by +13.49 and +2.56, respectively.

Table 2: **Comparison with state-of-the-art models on GenEval and DPG-Bench benchmark.**

| Model | # Params | GenEval↑ | | | | | DPG-Bench↑ | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Two Obj. | Counting | Position | Color Attri. | Overall | Global | Relation | Overall |
| *Text-to-Image Generation Models* | | | | | | | | | |
| DALLE-2 [61] | 6.5B | 0.66 | 0.49 | 0.10 | 0.19 | 0.52 | - | - | - |
| DALLE-3[4] | - | 0.87 | 0.47 | 0.43 | 0.45 | 0.67 | 90.97 | 90.58 | 83.50 |
| Emu3 [80] | 8B | 0.71 | 0.34 | 0.17 | 0.21 | 0.54 | 85.21 | 90.22 | 80.60 |
| SDXL [57] | 2.6B | 0.74 | 0.39 | 0.15 | 0.23 | 0.55 | 83.27 | 86.76 | 74.65 |
| SimpleAR [76] | 1.5B | 0.90 | - | 0.28 | 0.45 | 0.63 | 87.97 | 88.33 | 81.97 |
| Infinity [21] | 2B | 0.85 | - | 0.49 | 0.57 | 0.73 | 93.11 | 90.76 | 83.46 |
| *Unified MLLMs* | | | | | | | | | |
| Show-o [87] | 1.3B | 0.52 | 0.49 | 0.11 | 0.28 | 0.53 | 80.39* | 83.36* | 71.70* |
| Janus [84] | 1.3B | 0.68 | 0.30 | 0.46 | 0.42 | 0.61 | 82.33 | 85.46 | 79.68 |
| Janus-Pro [9] | 1.5B | 0.82 | 0.51 | 0.65 | 0.56 | 0.73 | 87.58 | 88.98 | 82.63 |
| ILLUME [74] | 7B | 0.86 | 0.45 | 0.39 | 0.28 | 0.61 | - | - | - |
| UniToken [29] | 7B | 0.80 | 0.35 | 0.38 | 0.39 | 0.63 | - | - | - |
| VARGPT-v1.1 [107] | 9B | 0.53 | 0.48 | 0.13 | 0.21 | 0.53 | 84.83 | 88.13 | 78.59 |
| TokenFlow-XL [58] | 13B | 0.72 | 0.45 | 0.45 | 0.42 | 0.63 | 78.72 | 85.22 | 73.38 |
| *UniGen* | 1.5B | 0.92 | 0.68 | 0.48 | 0.52 | **0.74** | 91.53 | 91.09 | **84.89** |
| *UniGen* + CoT-V | 1.5B | 0.94 | 0.78 | 0.57 | 0.54 | **0.78** | 91.95 | 92.04 | **85.19** |

Table 3: **Ablation of different stages of our model on image understanding benchmarks.**

| Model | Stage | GenEval | DPG-Bench | AI2D | GQA | POPE | MMMU | MathVista | RW-QA | ScienceQA | Seedbench |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *UniGen* | PT-1 | 0.53 | 78.14 | - | - | - | - | - | - | - | - |
| | PT-2 | 0.55 | 80.71 | - | - | - | - | - | - | - | - |
| | SFT | 0.63 | 82.75 | 68.0 | 62.5 | 87.4 | 32.4 | 45.2 | 58.6 | 79.7 | 71.1 |
| | DPO | 0.73 | 84.89 | 67.9 | 62.4 | 88.0 | 32.9 | 45.0 | 59.0 | 79.5 | 71.0 |
| | *CoT-V* | 0.78 | 85.19 | 67.4 | 62.3 | 87.8 | 32.3 | 44.6 | 56.7 | 79.4 | 70.8 |

## 4.3 Ablation Studies

### 4.3.1 Impact of Different Training Stages

We examine our training pipeline by showing the understanding and generation performance after each stage in Table 3. Here we highlight some key observations.

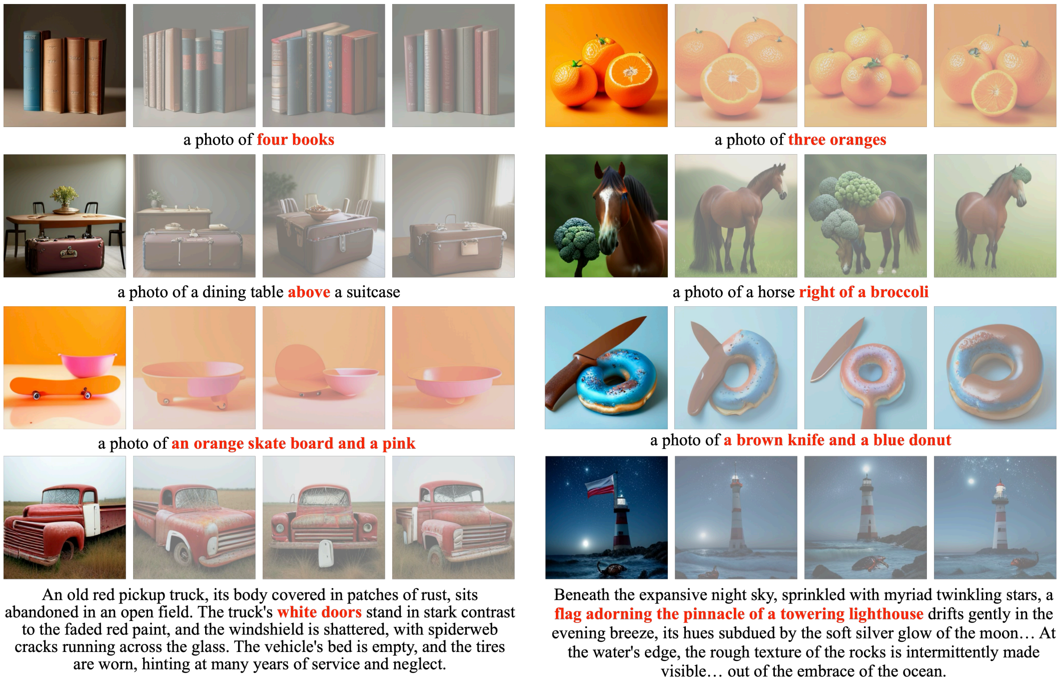
**First, *UniGen* demonstrates consistent improvements in generation performance across different training stages**, as indicated by the increasing numbers of GenEval and DPG-bench. The pre-training stages aim to warm up the generation capability of *UniGen*. The SFT boosts the GenEval and DPG-bench by using high-quality generation datasets. With the effectiveness of our preference data, the DPO stage significantly improves GenEval and DPG-bench to 0.73 (+0.10) and 84.89 (+2.14), respectively. *CoT-V* further enhances the scores to 0.78 (+0.05) and 85.19 (+0.3) via test-time scaling.

**Second, *UniGen*'s strong understanding capability is stimulated in the SFT stage and can be maintained in the following stages.** The SFT stage promotes the instruction following capability of *UniGen* that leads to strong performance on understanding benchmarks. In the DPO stage, *UniGen* successfully maintains the strong understanding capability. *CoT-V* contains an additional lightweight fine-tuning to encourage the CoT verification during test-time scaling. The results show that it does not sacrifice the general understanding capability, except for a slight regression on RealWorld-QA. We attribute this regression to the distribution gap between *CoT-V*'s synthetic training data and the real-world images in RealWorld-QA.

### 4.3.2 Ablation of *CoT-V*

Here we evaluate different verification methods discussed in Sec. 3.5 with the following highlights.

**First, CoT verification achieves the best performance and prompting *UniGen*'s thinking process is important.** As shown in Table 4, using *Outcome verification* shows no improvement, while using CoT thinking obtains a significant boost of generation performance on both GenEval and DPG-bench. We also observe that *Rule-based verification* is also effective, leading to 0.75 on GenEval. However, it is not general enough to be used on free-form prompts. Comparing the results from *CoT Verification* and *Rule-based Verification*, we can see that prompting the model itself to think is beneficial for more reliable critique.



a photo of **four books**

a photo of **three oranges**

a photo of a dining table **above** a suitcase

a photo of a horse **right of a broccoli**

a photo of **an orange skate board and a pink**

a photo of **a brown knife and a blue donut**

An old red pickup truck, its body covered in patches of rust, sits abandoned in an open field. The truck's **white doors** stand in stark contrast to the faded red paint, and the windshield is shattered, with spiderweb cracks running across the glass. The vehicle's bed is empty, and the tires are worn, hinting at many years of service and neglect.

Beneath the expansive night sky, sprinkled with myriad twinkling stars, a **flag adorning the pinnacle of a towering lighthouse** drifts gently in the evening breeze, its hues subdued by the soft silver glow of the moon… At the water's edge, the rough texture of the rocks is intermittently made visible… out of the embrace of the ocean.

Figure 5: **Visual examples of *UniGen*'s results using *CoT-V*.** The first three rows show examples for counting, position, and color attribute, respectively, and the last row shows images generated by free-form prompts. The first column contains images selected by *UniGen* as the test-time verifier.

Table 4: **Ablation of verification methods.**

| Method | Outcome | Rule | CoT | GenEval | DPG-Bench |
|---|---|---|---|---|---|
| *UniGen* | ✗ | ✗ | ✗ | 0.74 | 85.02 |
| | ✓ | ✗ | ✗ | 0.74 | 85.00 |
| | ✗ | ✓ | ✗ | 0.75 | - |
| | ✗ | ✗ | ✓ | 0.78 | 85.19 |

Table 5: **Ablation of *CoT-V* post-training.**

| Method | *CoT-V* Post-train | GenEval | DPG-Bench |
|---|---|---|---|
| Show-o | ✗ | 0.64 | 76.32 |
| | ✓ | 0.66 | 77.09 |
| *UniGen* | ✗ | 0.74 | 84.89 |
| | ✓ | 0.78 | 85.19 |

**Second, *CoT-V* post-training is essential for strong test-time verification.** As shown in Table 5, directly using *UniGen* without *CoT-V* post-train leads to notable performance drop, especially for GenEval. This comparison demonstrates that *CoT-V* post-train is pivotal for CoT verification.

**Third, *CoT-V* can effectively generalize to other models.** We finetune Show-o with DPO and *CoT-V* with our generated data to boost its generation performance. Results in Table 5 show that *CoT-V* is a general technique that can also enhance Show-o's generation performance.

### 4.3.3 Ablation of DPO

We ablate the contribution of each data source and demonstrate the effectiveness of our DPO data on other unified models.

**First, every prompt source contributes positively to generation performance.** Table 6 shows that adding only PARM DPO data results in remarkable improvements (row1 vs. row2), while further adding T2I-Comp mainly benefits DPG-Bench (row2 vs. row3). *UniGen-DPO* with all of three prompts, introduces the best overall performance (row3 vs. row4).

Table 6: **Ablation study of DPO.** The results are from *UniGen-DPO* without test-time scaling.

| Method | PARM | T2I-Comp | SA1B | GenEval | DPG-bench |
|---|---|---|---|---|---|
| *UniGen* | ✗ | ✗ | ✗ | 0.63 | 82.75 |
| | ✓ | ✗ | ✗ | 0.73 | 83.48 |
| | ✓ | ✓ | ✗ | 0.72 | 84.09 |
| | ✓ | ✓ | ✓ | 0.74 | 84.89 |
| Show-o | ✗ | ✗ | ✗ | 0.56 | 71.70 |
| | ✓ | ✓ | ✓ | 0.64 | 76.32 |

**Second, our DPO data also largely improves Show-o, showing that it is generalizable to other unified models.** When fine-tuning Show-o directly with our DPO data, we also observe a notable gain, from 0.56 to 0.64 on GenEval and from 71.70 to 76.32 on DPG-Bench as shown in Table 6.

### 4.4 Ablation of SFT

Table 7: **Ablation of SFT stage.** PT-2 Data denotes the training data used in the PT-2 Stage. JD and TI denote JourneyDB and text-2-image-2M, respectively. The results are from *UniGen-SFT*.

| Und Data | Gen Data | GenEval | DPG-Bench | AI2D | GQA | POPE | MMMU | MathVista | RW-QA | ScienceQA | Seedbench | Und Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SlowFast-LLaVA-1.5 | PT-2 Data | 0.56 | 79.67 | 68.3 | 62.4 | 87.5 | 33.3 | 42.2 | 54.4 | 79.6 | 70.7 | 62.3 |
| | JD+TI | 0.63 | 82.77 | 68.0 | 62.5 | 87.4 | 32.4 | 45.2 | 58.6 | 79.7 | 71.1 | 63.1 |
| LLaVA1.5 | JD+TI | 0.64 | 81.82 | 48.7 | 62.8 | 87.4 | 27.1 | 22.1 | 53.7 | 55.5 | 64.0 | 52.7 |

By default, we use the image mixture from SlowFast-LLaVA-1.5 [90] as understanding datasets and JourneyDB and text-2-image-2M as the generation datasets. In this section, we ablate the datasets in Table 7 to evaluate their impacts and draw the following conclusion.

**First, using high-quality generation data is necessary for further lifting generation results.** JourneyDB and text-2-image-2M have much higher quality compared to the generation data used during the PT-2 stage. Table 7 (row1 vs. row2) shows that using high-quality generation data in the SFT stage results in better image generation performance.

**Second, using a stronger data mixture is crucial to improve the understanding performance, which is also helpful for fine-grained text-to-image generation.** As shown in Table 7 (row2 vs. row3), replacing SlowFast-LLaVA-1.5 mixture with LLaVA1.5's induces much worse understanding performance. Also, training with SlowFast-LLaVA-1.5 data produces higher results on DPG-BENCH. We believe a better understanding capability is important for comprehending the complex text prompts of DPG-BENCH that can eventually be beneficial for better text-to-image generation.

### 4.5 Ablation of PT-1 and PT-2

Table 8: **Impact of using understanding task in PT stages.** The results are from *UniGen-SFT*.

| Und Data PT-1 | Und Data PT-2 | GenEval | DPG-bench | AI2D | GQA | POPE | MMMU | MathVista | RW-QA | ScienceQA | Seedbench | Und Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ✗ | ✗ | 0.61 | 82.51 | 60.5 | 59.6 | 87.4 | 30.9 | 38.1 | 49.0 | 72.0 | 66.1 | 58.0 |
| ✓ | ✓ | 0.63 | 82.75 | 68.0 | 62.5 | 87.4 | 32.4 | 45.2 | 58.6 | 79.7 | 71.1 | 63.1 |

Table 9: **Ablation of PT-1 stage.** Cls and Recap indicate class names and high-quality captions are used for generating images, respectively. The results are from *UniGen-SFT*.

| Stage I | Gen Data | GenEval | DPG-bench | AI2D | GQA | POPE | MMMU | MathVista | RW-QA | ScienceQA | Seedbench | Und Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ✗ | – | 0.64 | 82.26 | 70.3 | 62.5 | 87.9 | 33.7 | 45.7 | 54.2 | 80.5 | 71.7 | 63.3 |
| ✓ | ImageNet(Cls) | 0.63 | 82.75 | 67.0 | 62.5 | 88.0 | 31.4 | 41.4 | 53.6 | 79.8 | 71.1 | 61.8 |
| ✓ | ImageNet(Recap) | 0.63 | 82.75 | 68.0 | 62.5 | 87.4 | 32.4 | 45.2 | 58.6 | 79.7 | 71.1 | 63.1 |

We explore the necessity and key factors of *UniGen*'s pre-training stages. We first discuss whether we should include the understanding dataset in the pre-training stages as shown in Table 8. Second, we ablate the impact of the generation datasets to both generation and understanding performance in Table 9 (for PT-1) and Table 10 (for PT-2). Since PT-1 and PT-2 are early stages in *UniGen*'s training pipeline, we continue the training to the SFT stage to verify their impact on the final performance more reliably. Unless noted otherwise, all ablations in this section use *UniGen*'s default SFT settings. Here we highlight the following observations.

**First, including understanding data in pre-training stages is crucial for both generation and understanding performance.** In Table 8's row 1, we keep the default setting and only remove the understanding loss from the training objectives. We observe a significant performance decrease across generation and understanding benchmarks at the SFT stage. We attribute this to the fact that understanding data is important for a better vision-language alignment in early training stages, which is helpful for both image-to-text and text-to-image tasks.

**Second, the high-quality text-to-image task is more effective than the de facto class-to-image task in PT-1.** One common practice of unified MLLMs for pre-training is using the class-to-image task with ImageNet [84, 87]. However, we find that using ImageNet with fine-grained captions leads to better performance for understanding tasks in the *UniGen-SFT* stage as shown in Table 9 (row2 vs. row3). This is a result of better vision-language alignment introduced by the detailed caption-to-image mapping.

Table 10: **Ablation of PT-2 stage.** The reported results are from *UniGen-SFT*.

| Stage II | Gen & Und Data | GenEval | DPG-bench | AI2D | GQA | POPE | MMMU | MathVista | RW-QA | ScienceQA | Seedbench | Und Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ✗ | – | 0.58 | 79.25 | 70.2 | 61.9 | 87.3 | 31.6 | 47.0 | 54.9 | 82.5 | 71.2 | 63.3 |
| ✓ | CC+SA+IMN | 0.59 | 80.64 | 64.5 | 61.6 | 87.9 | 30.8 | 40.9 | 54.2 | 77.0 | 69.6 | 60.8 |
| ✓ | (SA+IMN)(Recap) | 0.64 | 82.63 | 67.2 | 62.4 | 87.5 | 31.0 | 40.8 | 56.5 | 79.7 | 71.1 | 62.0 |
| ✓ | (CC+IMN)(Recap) | 0.63 | 82.75 | 67.9 | 62.1 | 87.8 | 31.6 | 42.0 | 58.2 | 80.3 | 70.8 | 62.6 |
| ✓ | (CC+SA)(Recap) | 0.62 | 82.34 | 68.6 | 62.2 | 87.4 | 30.6 | 45.4 | 59.3 | 80.5 | 70.8 | 63.1 |
| ✓ | (CC+SA+IMN)(Recap) | 0.63 | 82.75 | 68.0 | 62.5 | 87.4 | 32.4 | 45.2 | 58.6 | 79.7 | 71.1 | 63.1 |

**Third, to maintain high performance on generation, we need both PT-1 and PT-2.** According to Table 9 (row1 vs. row3) and Table 10 (row1 vs. row6), we notice that completely removing PT-1 or PT-2 stage will largely decrease the generation metrics. Especially, eliminating PT-2 has a much bigger negative impact, leading to a dramatic drop of numbers on both GenEval and DPG-bench. Excluding PT-1 will not apparently affect GenEval, but hurts DPG-bench. This is because the prompts of DPG-bench are more complicated, thus more pre-training helps our model to better comprehend their semantics.

**Fourth, to keep a strong understanding performance, we need at least one of the PT-1 and PT-2.** According to Table 8, we infer that the understanding performance will be destroyed if we remove both PT-1 and PT-2. However, discarding PT-1 or PT-2 in Table 9 and Table 10 does not impact understanding numbers. As a result, we recommend keeping at least one of them for good understanding capability and leveraging both of them for the best generation and understanding performance if the compute budget allows.

**Fifth, using high-quality captions in PT-2 is important for understanding and generation performance.** Table 10 (row2 vs. row6) demonstrates that using high-quality image captions results in stronger performance in both understanding and generation tasks. This is due to the better text-to-image and image-to-text alignment learned from the fine-grained captions.

**Sixth, each data source of PT-2 has meaningful contributions.** We remove each data component from the training set of PT-2 and observe that retaining all of them leads to the best performance as shown in Table 10 (row3 to row6). This finding supports the usefulness of each dataset we curated.

## 5 Conclusion

We present *UniGen*, an MLLM for unified multimodal understanding and generation. We discuss the key factors along the entire training pipeline and propose optimization methods to improve the performance. We also make the first attempt to collaborate *UniGen*'s understanding and generation capabilities, by enabling *UniGen* to perform as both image generator and verifier during test-time scaling. As a result, we successfully further boost the image generation quality by a clear margin. Trained with only open-source datasets, *UniGen* achieves the state-of-the-art performance across extensive understanding and generation benchmarks. We hope our exploration and ablation studies provide insights into the future development of strong unified MLLMs.

**Limitation.** *First*, we instantiate *UniGen* with only a 1.5B model, since larger scales will impose much higher demands on the computational cost. However, larger models have been shown effective for improving both understanding and generation performance [9]. *Second*, our generation capability targets at promoting semantic alignment between the input text prompt and the generated image, therefore we only focus on a resolution of 256 × 256. We plan to support higher resolution image generation, such as 480p or even 1080p, which is valuable for improving the visual fidelity. *Third*, although achieving convincing results on DPG-Bench, *CoT-V* is still limited for complicated text prompts, due to the noisy CoT data generated by Qwen2.5VL as a pseudo labeler. This could be largely relieved by using a stronger pseudo labeler or leveraging human filtering in the future. Equipping *UniGen* with stronger reasoning and CoT capabilities in an earlier stage is also a promising direction.

**Broader Impact.** Unified MLLMs offer scientific benefits by enabling human-AI interaction and advancing general-purpose multimodal understanding. There are many real-world applications, such as design assistants, education, and collaborative robots. However, there could be unintended usages and we advocate responsible usage complying with applicable laws and regulations.

**Acknowledgment.** We thank Haiming Gang, Jesse Allardice, Shiyu Li, and Yifan Jiang for their kind help.

## References

[1] Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv:2404.14219*, 2024. 2

[2] 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. In *NeurIPS*, 2022. 2

[3] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv:2502.13923*, 2025. 3

[4] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science.*, 2023. 7, 26

[5] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *CVPR*, 2022. 3, 6

[6] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *CVPR*, 2021. 3

[7] Jiefeng Chen, Jie Ren, Xinyun Chen, Chengrun Yang, Ruoxi Sun, and Sercan Ö Arık. Sets: Leveraging self-verification and self-correction for improved test-time scaling. *arXiv:2501.19306*, 2025. 2

[8] Jiuhai Chen, Zhiyang Xu, Xichen Pan, Yushi Hu, Can Qin, Tom Goldstein, Lifu Huang, Tianyi Zhou, Saining Xie, Silvio Savarese, et al. Blip3-o: A family of fully open unified multimodal models-architecture, training and dataset. *arXiv preprint arXiv:2505.09568*, 2025. 2

[9] 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:2501.17811*, 2025. 1, 2, 4, 6, 7, 10, 25, 26

[10] DeepSeek-AI. Deepseek-v3 technical report. *arXiv:2412.19437*, 2024. 2

[11] Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, et al. Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models. *arXiv:2409.17146*, 2024. 2

[12] Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, et al. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025. 2

[13] Yuhao Dong, Zuyan Liu, Hai-Long Sun, Jingkang Yang, Winston Hu, Yongming Rao, and Ziwei Liu. Insight-v: Exploring long-chain visual reasoning with multimodal large language models. *arXiv:2411.14432*, 2024. 2

[14] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, 2021. 2

[15] Rongyao Fang, Chengqi Duan, Kun Wang, Linjiang Huang, Hao Li, Shilin Yan, Hao Tian, Xingyu Zeng, Rui Zhao, Jifeng Dai, et al. Got: Unleashing reasoning capability of multimodal large language model for visual generation and editing. *arXiv:2503.10639*, 2025. 2

[16] Yuying Ge, Sijie Zhao, Jinguo Zhu, Yixiao Ge, Kun Yi, Lin Song, Chen Li, Xiaohan Ding, and Ying Shan. Seed-x: Multimodal models with unified multi-granularity comprehension and generation. *arXiv:2404.14396*, 2024. 1

[17] Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. In *Advances in Neural Information Processing Systems*, 2023. 5, 25

[18] Grok-1.5. https://x.ai/news/grok-1.5v, 2024. 25

[19] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv:2501.12948*, 2025. 2

[20] Ziyu Guo, Renrui Zhang, Chengzhuo Tong, Zhizheng Zhao, Peng Gao, Hongsheng Li, and Pheng-Ann Heng. Can we generate images with cot? let's verify and reinforce image generation step by step. *arXiv:2501.13926*, 2025. 2, 4, 5

[21] Jian Han, Jinlai Liu, Yi Jiang, Bin Yan, Yuqi Zhang, Zehuan Yuan, Bingyue Peng, and Xiaobing Liu. Infinity: Scaling bitwise autoregressive modeling for high-resolution image synthesis. *arXiv:2412.04431*, 2024. 7, 26

[22] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv:2207.12598*, 2022. 6

[23] Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. Ella: Equip diffusion models with llm for enhanced semantic alignment. *arXiv:2403.05135*, 2024. 5, 25

[24] Chengsong Huang, Langlin Huang, Jixuan Leng, Jiacheng Liu, and Jiaxin Huang. Efficient test-time scaling via self-calibration. *arXiv:2503.00031*, 2025. 2

[25] Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench: A comprehensive benchmark for open-world compositional text-to-image generation. In *NeurIPS*, 2023. 4

[26] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*, 2019. 25

[27] Hamish Ivison, Yizhong Wang, Jiacheng Liu, Zeqiu Wu, Valentina Pyatkin, Nathan Lambert, Noah A Smith, Yejin Choi, and Hanna Hajishirzi. Unpacking dpo and ppo: Disentangling best practices for learning from preference feedback. In *NeurIPS*, 2024. 2

[28] jackyhate. text-to-image-2m: A high-quality, diverse text-to-image training dataset. `https://huggingface.co/datasets/jackyhate/text-to-image-2M`, 2024. 4

[29] Yang Jiao, Haibo Qiu, Zequn Jie, Shaoxiang Chen, Jingjing Chen, Lin Ma, and Yu-Gang Jiang. Unitoken: Harmonizing multimodal understanding and generation through unified visual encoding. *arXiv:2504.04423*, 2025. 2, 6, 7, 26

[30] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *ECCV*, 2016. 25

[31] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, 2023. 3

[32] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv:2408.03326*, 2024. 2, 6

[33] Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv:2307.16125*, 2023. 25

[34] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023. 2

[35] Yanghao Li, Rui Qian, Bowen Pan, Haotian Zhang, Haoshuo Huang, Bowen Zhang, Jialing Tong, Haoxuan You, Xianzhi Du, Zhe Gan, et al. Manzano: A simple and scalable unified multimodal model with a hybrid vision tokenizer. *arXiv preprint arXiv:2509.16197*, 2025. 2

[36] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *EMNLP*, 2023. 25

[37] Zhimin Li, Jianwei Zhang, Qin Lin, Jiangfeng Xiong, Yanxin Long, Xinchi Deng, Yingfang Zhang, Xingchao Liu, Minbin Huang, Zedong Xiao, et al. Hunyuan-dit: A powerful multi-resolution diffusion transformer with fine-grained chinese understanding. *arXiv:2405.08748*, 2024. 26

[38] Jiaqi Liao, Zhengyuan Yang, Linjie Li, Dianqi Li, Kevin Lin, Yu Cheng, and Lijuan Wang. Imagegen-cot: Enhancing text-to-image in-context learning with chain-of-thought reasoning. *arXiv:2503.19312*, 2025. 2

[39] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *CVPR*, 2024. 6

[40] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. 2023. 2, 3

[41] Zhijian Liu, Ligeng Zhu, Baifeng Shi, Zhuoyang Zhang, Yuming Lou, Shang Yang, Haocheng Xi, Shiyi Cao, Yuxian Gu, Dacheng Li, et al. Nvila: Efficient frontier visual language models. *arXiv:2412.04468*, 2024. 2

[42] Zuyan Liu, Yuhao Dong, Ziwei Liu, Winston Hu, Jiwen Lu, and Yongming Rao. Oryx MLLM: On-demand spatial-temporal understanding at arbitrary resolution. *ICLR*, 2025. 2

[43] Jiasen Lu, Christopher Clark, Sangho Lee, Zichen Zhang, Savya Khosla, Ryan Marten, Derek Hoiem, and Aniruddha Kembhavi. Unified-io 2: Scaling autoregressive multimodal models with vision, language, audio, and action. *arXiv:2312.17172*, 2023. 2

[44] Jiasen Lu, Christopher Clark, Rowan Zellers, Roozbeh Mottaghi, and Aniruddha Kembhavi. Unified-io: A unified model for vision, language, and multi-modal tasks. *arXiv:2206.08916*, 2022. 2

[45] Jiasen Lu, Liangchen Song, Mingze Xu, Byeongjoo Ahn, Yanjun Wang, Chen Chen, Afshin Dehghan, and Yinfei Yang. Atoken: A unified tokenizer for vision. *arXiv preprint arXiv:2509.14476*, 2025. 2

[46] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *ICLR*, 2024. 25

[47] Yiyang Ma, Xingchao Liu, Xiaokang Chen, Wen Liu, Chengyue Wu, Zhiyu Wu, Zizheng Pan, Zhenda Xie, Haowei Zhang, Liang Zhao, et al. Janusflow: Harmonizing autoregression and rectified flow for unified multimodal understanding and generation. *arXiv:2411.07975*, 2024. 2

[48] Brandon McKinzie, Zhe Gan, Jean-Philippe Fauconnier, Sam Dodge, Bowen Zhang, Philipp Dufter, Dhruti Shah, Xianzhi Du, Futang Peng, Anton Belyi, et al. Mm1: methods, analysis and insights from multimodal llm pre-training. In *ECCV*, 2024. 2

[49] Lingchen Meng, Jianwei Yang, Rui Tian, Xiyang Dai, Zuxuan Wu, Jianfeng Gao, and Yu-Gang Jiang. Deepstack: Deeply stacking visual tokens is surprisingly simple and effective for lmms. In *NeurIPS*, 2024. 2

[50] Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: Vq-vae made simple. *arXiv:2309.15505*, 2023. 2

[51] David Mizrahi, Roman Bachmann, Oğuzhan Fatih Kar, Teresa Yeo, Mingfei Gao, Afshin Dehghan, and Amir Zamir. 4M: Massively multimodal masked modeling. In *NeurIPS*, 2023. 2

[52] OpenAI. Openai-o1. 2

[53] OpenAI. Gpt-4o, 2024. 1

[54] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *NeurIPS*, 2022. 2

[55] Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. The refinedweb dataset for falcon llm: outperforming curated corpora with web data, and web data only. *arXiv:2306.01116*, 2023. 3

[56] Wujian Peng, Lingchen Meng, Yitong Chen, Yiweng Xie, Yang Liu, Tao Gui, Hang Xu, Xipeng Qiu, Zuxuan Wu, and Yu-Gang Jiang. Inst-it: Boosting multimodal instance understanding via explicit visual prompt instruction tuning. *arXiv preprint arXiv:2412.03565*, 2024. 2

[57] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *ICLR*, 2024. 7, 26

[58] Liao Qu, Huichao Zhang, Yiheng Liu, Xu Wang, Yi Jiang, Yiming Gao, Hu Ye, Daniel K Du, Zehuan Yuan, and Xinglong Wu. Tokenflow: Unified image tokenizer for multimodal understanding and generation. *arXiv:2412.03069*, 2024. 2, 7, 26

[59] 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 *ICML*, 2021. 2

[60] 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. In *NeurIPS*, 2023. 2

[61] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv:2204.06125*, 2022. 7, 26

[62] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. Imagenet-21k pretraining for the masses. *arXiv:2104.10972*, 2021. 3

[63] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018. 3

[64] Keqiang Sun, Junting Pan, Yuying Ge, Hao Li, Haodong Duan, Xiaoshi Wu, Renrui Zhang, Aojun Zhou, Zipeng Qin, Yi Wang, et al. Journeydb: A benchmark for generative image understanding. In *NeurIPS*, 2023. 4

[65] Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv:2406.06525*, 2024. 25

[66] Zhiqing Sun, Sheng Shen, Shengcao Cao, Haotian Liu, Chunyuan Li, Yikang Shen, Chuang Gan, Liang-Yan Gui, Yu-Xiong Wang, Yiming Yang, Kurt Keutzer, and Trevor Darrell. Aligning large multimodal models with factually augmented rlhf. *arXiv:2309.14525*, 2023. 2

[67] Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv:2405.09818*, 2024. 1, 2

[68] Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1.5: Scaling reinforcement learning with llms. *arXiv:2501.12599*, 2025. 2

[69] Peter Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Adithya Jairam Vedagiri IYER, Sai Charitha Akula, Shusheng Yang, Jihan Yang, Manoj Middepogu, Ziteng Wang, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. In *NeurIPS*, 2024. 2

[70] Shengbang Tong, David Fan, Jiachen Zhu, Yunyang Xiong, Xinlei Chen, Koustuv Sinha, Michael Rabbat, Yann LeCun, Saining Xie, and Zhuang Liu. Metamorph: Multimodal understanding and generation via instruction tuning. *arXiv:2412.14164*, 2024. 2

[71] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv:2302.13971*, 2023. 2

[72] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv:2502.14786*, 2025. 2

[73] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. In *NeurIPS*, 2017. 2

[74] Chunwei Wang, Guansong Lu, Junwei Yang, Runhui Huang, Jianhua Han, Lu Hou, Wei Zhang, and Hang Xu. Illume: Illuminating your llms to see, draw, and self-enhance. *arXiv:2412.06673*, 2024. 7, 26

[75] Junke Wang, Yi Jiang, Zehuan Yuan, Bingyue Peng, Zuxuan Wu, and Yu-Gang Jiang. Omnitokenizer: A joint image-video tokenizer for visual generation. In *NeurIPS*, 2024. 2

[76] Junke Wang, Zhi Tian, Xun Wang, Xinyu Zhang, Weilin Huang, Zuxuan Wu, and Yu-Gang Jiang. Simplear: Pushing the frontier of autoregressive visual generation through pretraining, sft, and rl. *arXiv:2504.11455*, 2025. 7, 26

[77] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv:2409.12191*, 2024. 2

[78] Weiyun Wang, Zhe Chen, Wenhai Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Jinguo Zhu, Xizhou Zhu, Lewei Lu, Yu Qiao, et al. Enhancing the reasoning ability of multimodal large language models via mixed preference optimization. *arXiv:2411.10442*, 2024. 2

[79] Weiyun Wang, Zhangwei Gao, Lianjie Chen, Zhe Chen, Jinguo Zhu, Xiangyu Zhao, Yangzhou Liu, Yue Cao, Shenglong Ye, Xizhou Zhu, et al. Visualprm: An effective process reward model for multimodal reasoning. *arXiv:2503.10291*, 2025. 2, 5

[80] 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:2409.18869*, 2024. 7, 26

[81] Yi Wang, Mushui Liu, Wanggui He, Longxiang Zhang, Ziwei Huang, Guanghao Zhang, Fangxun Shu, Zhong Tao, Dong She, Zhelun Yu, et al. Mint: Multi-modal chain of thought in unified generative models for enhanced image generation. *arXiv:2503.01298*, 2025. 2

[82] Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. Large language models are better reasoners with self-verification. In *EMNLP*, 2023. 2

[83] Zejia Weng, Xitong Yang, Zhen Xing, Zuxuan Wu, and Yu-Gang Jiang. Genrec: Unifying video generation and recognition with diffusion models. In *NeurIPS*, 2024. 1

[84] 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. *arXiv:2410.13848*, 2024. 1, 2, 3, 6, 7, 9, 25, 26

[85] 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. In *ICLR*, 2025. 2, 6

[86] Enze Xie, Junsong Chen, Yuyang Zhao, Jincheng Yu, Ligeng Zhu, Chengyue Wu, Yujun Lin, Zhekai Zhang, Muyang Li, Junyu Chen, et al. Sana 1.5: Efficient scaling of training-time and inference-time compute in linear diffusion transformer. *arXiv preprint arXiv:2501.18427*, 2025. 5

[87] 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:2408.12528*, 2024. 1, 2, 6, 7, 9, 25, 26

[88] Tianyi Xiong, Xiyao Wang, Dong Guo, Qinghao Ye, Haoqi Fan, Quanquan Gu, Heng Huang, and Chunyuan Li. Llava-critic: Learning to evaluate multimodal models. *arXiv:2410.02712*, 2024. 2

[89] Mingze Xu, Mingfei Gao, Zhe Gan, Hong-You Chen, Zhengfeng Lai, Haiming Gang, Kai Kang, and Afshin Dehghan. Slowfast-llava: A strong training-free baseline for video large language models. *arXiv:2407.15841*, 2024. 2

[90] Mingze Xu, Mingfei Gao, Shiyu Li, Jiasen Lu, Zhe Gan, Zhengfeng Lai, Meng Cao, Kai Kang, Yinfei Yang, and Afshin Dehghan. Slowfast-llava-1.5: A family of token-efficient video large language models for long-form video understanding. *arXiv:2503.18943*, 2025. 2, 4, 9, 28

[91] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv:2412.15115*, 2024. 5, 6

[92] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv:2412.15115*, 2024. 2

[93] Jian Yang, Dacheng Yin, Yizhou Zhou, Fengyun Rao, Wei Zhai, Yang Cao, and Zheng-Jun Zha. Mmar: Towards lossless multi-modal auto-regressive probabilistic modeling. *arXiv:2410.10798*, 2024. 6

[94] Ling Yang, Xinchen Zhang, Ye Tian, Chenming Shang, Minghao Xu, Wentao Zhang, and Bin Cui. Hermesflow: Seamlessly closing the gap in multimodal understanding and generation. *arXiv:2502.12148*, 2025. 2

[95] Huanjin Yao, Jiaxing Huang, Wenhao Wu, Jingyi Zhang, Yibo Wang, Shunyu Liu, Yingjie Wang, Yuxin Song, Haocheng Feng, Li Shen, et al. Mulberry: Empowering mllm with o1-like reasoning and reflection via collective monte carlo tree search. *arXiv:2412.18319*, 2024. 2

[96] Hanrong Ye, De-An Huang, Yao Lu, Zhiding Yu, Wei Ping, Andrew Tao, Jan Kautz, Song Han, Dan Xu, Pavlo Molchanov, et al. X-vila: Cross-modality alignment for large language model. *arXiv:2405.19335*, 2024. 1

[97] Lijun Yu, Jose Lezama, Nitesh Bharadwaj Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Agrim Gupta, Xiuye Gu, Alexander G Hauptmann, et al. Language model beats diffusion-tokenizer is key to visual generation. In *ICLR*, 2024. 3

[98] Lijun Yu, José Lezama, Nitesh B. Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Agrim Gupta, Xiuye Gu, Alexander G. Hauptmann, Boqing Gong, Ming-Hsuan Yang, Irfan Essa, David A. Ross, and Lu Jiang. Language model beats diffusion – tokenizer is key to visual generation. *arXiv:2310.05737*, 2023. 2

[99] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *CVPR*, 2024. 25

[100] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023. 2, 3, 6

[101] Boqiang Zhang, Kehan Li, Zesen Cheng, Zhiqiang Hu, Yuqian Yuan, Guanzheng Chen, Sicong Leng, Yuming Jiang, Hang Zhang, Xin Li, et al. VideoLLaMA 3: Frontier multimodal foundation models for image and video understanding. *arXiv:2501.13106*, 2025. 2

[102] Haotian Zhang, Mingfei Gao, Zhe Gan, Philipp Dufter, Nina Wenzel, Forrest Huang, Dhruti Shah, Xianzhi Du, Bowen Zhang, Yanghao Li, et al. Mm1. 5: Methods, analysis & insights from multimodal llm fine-tuning. *arXiv:2409.20566*, 2024. 2, 6

[103] Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data. *arXiv:2410.02713*, 2024. 2

[104] Yue Zhao, Yuanjun Xiong, and Philipp Krähenbühl. Image and video tokenization with binary spherical quantization. *arXiv:2406.07548*, 2024. 2

[105] Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model. *arXiv:2408.11039*, 2024. 2

[106] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Yuchen Duan, Hao Tian, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv:2504.10479*, 2025. 2, 5

[107] Xianwei Zhuang, Yuxin Xie, Yufan Deng, Dongchao Yang, Liming Liang, Jinghan Ru, Yuguo Yin, and Yuexian Zou. Vargpt-v1. 1: Improve visual autoregressive large unified model via iterative instruction tuning and reinforcement learning. *arXiv:2504.02949*, 2025. 7, 26

[108] Orr Zohar, Xiaohan Wang, Yann Dubois, Nikhil Mehta, Tong Xiao, Philippe Hansen-Estruch, Licheng Yu, Xiaofang Wang, Felix Juefei-Xu, Ning Zhang, et al. Apollo: An exploration of video understanding in large multimodal models. *arXiv:2412.10360*, 2024. 2

## 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 abstract and introduction have accurately claimed the contributions of this work, including the *UniGen* model and the *CoT-V* test-time scaling strategy.

   Guidelines:

   - The answer NA 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 NA 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: [Yes]

   Justification: Limitations have been discussed in Sec.5.

   Guidelines:

   - The answer NA 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: [NA]

Justification: This paper does not include theoretical results.

Guidelines:

- The answer NA 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 fully discloses all the information needed to reproduce the main experimental results in main paper's Sec. 3 and Sec. 4.1.

   Guidelines:

   - The answer NA 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: [Yes]

Justification: The paper plans to open-source the code and data after the internal review.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.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://nips.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, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper has specified the training and test details in Sec. 4.1.

   Guidelines:

   - The answer NA 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: Following common practice in the multimodal learning literature, we do not report error bars in this paper because of the heavy computation overheads.

   Guidelines:

   - The answer NA 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: The paper has provided the computation information in Sec. 4.1.

   Guidelines:

   - The answer NA 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: The paper is under the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA 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: [Yes]

    Justification: Unified MLLMs offer significant scientific benefits by enabling more intuitive human-AI interaction and advancing general-purpure multimodal understanding. There are many real-world application scenarios, such as design assistant, education, and collaborative robots. However, there could be unintended usages and we advocate responsible usage complying with applicable laws and regulations.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA 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 properly cite the original assets in the paper.

    Guidelines:

    - The answer NA 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: [Yes]

    Justification: The details of the newly contributed dataset/code/model have been discussed in Sec. 3 of the main paper.

    Guidelines:

    - The answer NA 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: [NA]

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

    Guidelines:

    - The answer NA 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: [NA]

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

    Guidelines:

    - The answer NA 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 rigorousness, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: The LLM is used to labeling the training data. We clarified their usage in the main paper.

    Guidelines:

    - The answer NA 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 (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

# A Benchmarks and Evaluation Protocol

**For image understanding**, we include widely-used *(i)* general VQA benchmarks, such as GQA [26], RealWorld-QA [18], and Seedbench [33], *(ii)* knowledge-based benchmarks, such as AI2D [30], MMMU [99], and MathVista [46], and *(iii)* hallucination benchmarks, such as POPE [36]. We leverage the `lmms-eval`[1] toolkit to compute the results for the above benchmarks.

**For text-to-image generation benchmarks**, we report results on GenEval [17] and DPG-Bench [23] to comprehensively evaluate the semantic alignment between a text prompt and the generated images. To fairly compare with recent unified MLLMs [84, 9, 87], our results are obtained using the official evaluation repository of GenEval[2] and DPG-Bench[3].

# B More Ablation Studies

Table 11: **Ablation of Best-of-N strategy.** The results are from *UniGen* with *CoT-V*.

| $N$ | GenEval | DPG-Bench | Speed (s/img) |
|---|---|---|---|
| 1 | 0.73 | 84.89 | 3.55 |
| 3 | 0.77 | 85.13 | 16.59 |
| 5 | 0.78 | 85.19 | 27.22 |

Table 12: **Ablation of the order of *CoT-V* post-training and DPO training.** The results are from *UniGen* with *CoT-V*.

| Sequence | Und Avg. | GenEval | DPG-bench |
|---|---|---|---|
| DPO → *CoT-V* Post-train | 62.7 | 0.78 | 85.19 |
| *CoT-V* Post-train → DPO | 62.7 | 0.76 | 85.20 |

**Choosing $N$ for Best-of-N strategy.** We ablate $N$ for Best-of-$N$ selection with *CoT-V*. As shown in Table 11, using larger $N$ consistently yields higher performance on GenEval benchmark. To assess the efficiency trade-off, we further measure the inference speed of *CoT-V* on the same H100 GPU, considering both MLLM execution time and tokenizer decoding overhead. The average speed is computed across all GenEval samples using one prompt per batch. Under the default setting of (N=5), the total inference time is approximately $8\times$ slower than standard inference, highlighting the trade-off between increased computational cost and improved accuracy.

**Switching the order of CoT-V post-training and DPO training.** We reverse the order of DPO training and CoT-V post-training and present the results in Table 12. The findings indicate that the training order has minimal impact, with the default setting showing a slight advantage on GenEval.

Table 13: **Ablation of visual tower for understanding.** The results are from *UniGen*-SFT.

| Visual Encoder | Und Avg. | GenEval | DPG-Bench |
|---|---|---|---|
| Freeze | 63.11 | 0.63 | 82.75 |
| Unfreeze | 63.16 | 0.65 | 82.71 |

Table 14: **Ablation of visual tokenizer for generation.** The results are from *UniGen*-SFT.

| Tokenizer | GenEval | DPG-bench |
|---|---|---|
| MAGViTv2 | 0.63 | 82.75 |
| VQ-16 | 0.62 | 82.93 |

**Freezing the visual tower for understanding** during supervised fine-tuning achieves performance on par with the unfrozen setting on understanding benchmarks. As reported in Table 13, unfreezing the visual tower leads to a 2% improvement on GenEval, while freezing the encoder reduces computational cost. These results indicate that the frozen design leads to a more cost-efficient option with only minor performance differences.

**Changing the discrete visual tokenizers.** By default, we adopt the MAGViTv2 implementation from Show-o[4]. To ablate the impact of discrete visual tokenizer on generation, we further experiment with the VQ-16 tokenizer from LLamaGen [65]. As shown in Table 14, both tokenizers achieve comparable performance on GenEval and DPG-Bench after supervised fine-tuning, demonstrating the robustness and generalizability of our training framework.

[1] https://github.com/EvolvingLMMs-Lab/lmms-eval

[2] https://github.com/djghosh13/geneval/tree/main

[3] https://github.com/TencentQQGYLab/ELLA/tree/main

[4] https://huggingface.co/showlab/magvitv2

# C More Results

We present the breakdown comparison of *UniGen* against state-of-the-art models on GENEVAL and DPG-BENCH in Table 15 and Table 16.

Table 15: **Comparison with state-of-the-art models on the GenEval benchmark.**

| Model | #Params | Single Obj. | Two Obj. | Counting | Colors | Position | Color Attri. | Overall↑ |
|---|---|---|---|---|---|---|---|---|
| *Text-to-Image Generation Models* | | | | | | | | |
| DALLE-2 [61] | 6.5B | 0.94 | 0.66 | 0.49 | 0.77 | 0.10 | 0.19 | 0.52 |
| DALLE-3[4] | - | 0.96 | 0.87 | 0.47 | 0.83 | 0.43 | 0.45 | 0.67 |
| Emu3 [80] | 8B | 0.98 | 0.71 | 0.34 | 0.81 | 0.17 | 0.21 | 0.54 |
| SDXL [57] | 2.6B | 0.98 | 0.74 | 0.39 | 0.85 | 0.15 | 0.23 | 0.55 |
| SimpleAR [76] | 1.5B | - | 0.90 | - | - | 0.28 | 0.45 | 0.63 |
| Infinity [21] | 2B | - | 0.85 | - | - | 0.49 | 0.57 | 0.73 |
| *Unified MLLMs* | | | | | | | | |
| Show-o [87] | 1.3B | 0.95 | 0.52 | 0.49 | 0.82 | 0.11 | 0.28 | 0.53 |
| Janus [84] | 1.3B | 0.97 | 0.68 | 0.30 | 0.84 | 0.46 | 0.42 | 0.61 |
| Janus-Pro [9] | 1.5B | 0.98 | 0.82 | 0.51 | 0.89 | 0.65 | 0.56 | 0.73 |
| ILLUME [74] | 7B | 0.99 | 0.86 | 0.45 | 0.71 | 0.39 | 0.28 | 0.61 |
| UniToken [29] | 7B | 0.99 | 0.80 | 0.35 | 0.84 | 0.38 | 0.39 | 0.63 |
| VARGPT-v1.1 [107] | 9B | 0.96 | 0.53 | 0.48 | 0.83 | 0.13 | 0.21 | 0.53 |
| TokenFlow-XL [58] | 13B | 0.93 | 0.72 | 0.45 | 0.82 | 0.45 | 0.42 | 0.63 |
| *UniGen* | 1.5B | 1.00 | 0.92 | 0.68 | 0.87 | 0.48 | 0.52 | **0.74** |
| *UniGen + CoT-V* | 1.5B | 1.00 | 0.94 | 0.78 | 0.87 | 0.57 | 0.54 | **0.78** |

Table 16: **Comparison with state-of-the-art models on the DPG-bench benchmark**.

| Model | #Params | Global | Entity | Attribute | Relation | Other | Overall↑ |
|---|---|---|---|---|---|---|---|
| *Text-to-Image Generation Models* | | | | | | | |
| Hunyuan-DiT [37] | - | 84.59 | 80.59 | 88.01 | 74.36 | 86.41 | 78.87 |
| DALLE-3[4] | - | 90.97 | 89.61 | 88.39 | 90.58 | 89.83 | 83.50 |
| Emu3 [80] | 8B | 85.21 | 86.68 | 86.84 | 90.22 | 83.15 | 80.60 |
| SDXL [57] | 2.6B | 83.27 | 82.43 | 80.91 | 86.76 | 80.41 | 74.65 |
| SimpleAR [76] | 1.5B | 87.97 | - | - | 88.33 | - | 81.97 |
| Infinity [21] | 2B | 93.11 | - | - | 90.76 | - | 83.46 |
| *Unified MLLMs* | | | | | | | |
| Show-o* [87] | 1.3B | 80.39 | 80.94 | 82.17 | 83.36 | 82.88 | 71.70 |
| Janus [84] | 1.3B | 82.33 | 87.38 | 87.70 | 85.46 | 86.41 | 79.68 |
| Janus-Pro [9] | 1.5B | 87.58 | 88.63 | 88.17 | 88.98 | 88.30 | 82.63 |
| VARGPT-v1.1 [107] | 9B | 84.83 | 82.80 | 84.95 | 88.13 | 87.70 | 78.59 |
| TokenFlow-XL [58] | 13B | 78.72 | 79.22 | 81.29 | 85.22 | 71.20 | 73.38 |
| *UniGen* | 1.5B | 91.53 | 90.39 | 90.30 | 91.09 | 90.86 | **84.89** |
| *UniGen + CoT-V* | 1.5B | 91.95 | 89.68 | 90.90 | 92.04 | 90.91 | **85.19** |

# D Details of Test-Time Strategies

### D.0.1 Prompts of different verifications for test-time inference

Prompt. 1: Chain-of-Thought Verification

{image} This image is generated by a prompt: {prompt}. Please assess the image generation quality step by step. First, breakdown the prompt into multiple visual questions and iteratively answer each question with Yes or No between <think_start> <think_end>. Questions should cover all-round details about whether the image accurately represents entity categories, counting of entities, color, spatial relationship in the prompt. Next, output the final result between <answer_start> <answer_end>. Output Yes if all multi-choice answers equal yes to show the image has accurate alignment with the prompt. Otherwise answer with No.

Prompt. 2: Outcome Verification

{image} This image is generated by a prompt: {prompt}. Does this image accurately represent the prompt? Please answer yes or no.

Prompt. 3: Rule-based Verification

{image} {question} Please answer yes or no with detail explanation.

## E Details of Training

Table 17: **Hyperparameter setup for different training stages of *UniGen*.** Data ratio refers to the ratio of image understanding data, pure text data, and image generation data.

| **Hyperparameters** | **PT-1** | **PT-2** | **SFT** | **DPO** | ***CoT-V* Post-Training** |
|---|---|---|---|---|---|
| Learning rate | $1.0 \times 10^{-4}$ | $1.0 \times 10^{-4}$ | $1.0 \times 10^{-3}$ | $1.0 \times 10^{-5}$ | $1.0 \times 10^{-5}$ |
| LR scheduler | Cosine | Cosine | Cosine | Cosine | Cosine |
| Weight decay | 0.01 | 0.01 | 0.05 | 0.05 | 0.05 |
| Gradient clip | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| Optimizer | AdamW | AdamW | AdamW | AdamW | AdamW |
| Warm-up steps | 6000 | 5000 | 1000 | 500 | 0 |
| Training steps | 150k | 400k | 146k | 1.6k | 0.5k |
| H100 hours | 1.0k | 2.8k | 240 | 5 | 0.7 |
| Batch size | 896 | 512 | 64 | 80 | 64 |
| Data ratio | 2:1:4 | 2:1:4 | 4:1:3 | -:-:1 | 1:-:- |

### E.0.1 Training Parameters

Details of hyperparameters during each training stage are presented in Table 17.

### E.0.2 Training Data Overview

We list the datasets used in our training stages in Table 18. Refer to Appendix E.0.4 and Appendix E.0.5 for more details about the preference data and *CoT-V* data.

### E.0.3 Prompts for Generating Pre-Train Data

We use Prompt. 4 to prompt Qwen2.5VL-7B for generating fine-grained captions for CC-3M, CC-12M, SA-1B and ImageNet that are used in pre-training stages as shown in Table 18.

Prompt. 4: Re-caption

{image} What is the content of this image?

### E.0.4 Preference Data Generation for DPO

**PARM.** GenEval metric is used to rate each generated image candidate per prompt. The highest and lowest rated ones are used as the preferred and rejected samples for this prompt.

**T2I-Comp and SA-1B.** These prompts are more complex than the prompts of PARM, therefore, it is difficult to rate each generated image using rule-based metrics. We adopt a two-step approach to evaluate the coherence between an image and the prompt. First, we use Qwen2.5-7B to decompose each text prompt into atomic facts represented as questions using Prompt. 5. Then, the image with each decomposed question is fed into Qwen2.5VL-7B with Prompt. 6. The model responds with *yes* if visual generation passes the fact-check and with *no* otherwise. The final score of an image is calculated by averaging the results of all fact-checks. We take the most and least aligned images per prompt as the preferred and rejected sample pairs.

**Prompt. 5: Visual Questions Generation**

Now you need to convert an image description into fine-grained, related visual questions. The questions should comprehensively cover detailed visual facts of entities, attributes (e.g., color, count, texture, shape, and size), and relationships (e.g., spatial and non-spatial) between the entities mentioned in the description. Please complete the task by analyzing each clause in the sentence step by step. For each clause, first raise questions about whether each mentioned entity exists in the image. Then, raise questions about whether the attributes or relationships of the entities are accurately represented in the image. For an image accurately aligned with the description, all questions should be answered with "yes"; otherwise, they should be answered with "no".
Make sure all questions are able to be responded with yes or no and are connected with semicolon. Here are examples:
Example 1:
*description*: three black keys, four chickens and a fabric blanket
*output*: Are there keys?; Are there three keys?; Are the keys black?; Are there chickens?; Are there four chickens?; Is there a blanket?; Is the blanket fabric?
Example 2:
*description*: A person in a blue shirt and red and black apron is using a power tool, likely a drill, to assemble a white cabinet or shelving unit indoors. The floor is covered with light-colored wood or laminate material.
*output*: Is there a person?; Is the person wearing a shirt; Is the shirt blue?; Is the person wearing a apron?; Is the apron red and black?; Is the person using a drill?; Is there a white cabinet or shelving unit?; Is the person using the drill indoors?; Is there light-colored wood on the floor?; Is there laminate material on the floor?
Example 3:
*description*: a large Ferris wheel with a digital clock showing the time as 11:00. The Ferris wheel is located in an urban area, as indicated by the modern buildings in the background. There is also a tree on the left side of the image, partially obscuring the view of the Ferris wheel. The sky appears clear, suggesting a sunny day.
*output*: Is there a Ferris wheel?; Is there a digital clock?; Is the digital clock on the Ferris wheel?; Is the digital clock showing the time as 11:00?; Is the Ferris wheel located in an urban area?; Are there modern buildings in the background?; Is there a tree on the left side?; Is the sky clear and sunny?
Please convert this image description:{description}into fine-grained related visual questions.

**Prompt. 6: Visual Fact-Check**

{image} {question} Please answer yes or no without explanation.

We display some visual results of DPO preference data in Fig. 6.

Table 18: **Training data overview.** CC, SA, IMN, JD, T2I indicate CC-3M&CC-12M, SA-1B, ImageNet, JourneyDB, text-2-image-2M, respectively. Recap denotes that the images are re-captioned using Qwen2.5VL-7B.

| Stage | Gen Data | Und Data | Text-only |
|---|---|---|---|
| PT-1 | IMN (Recap) | (CC+SA+IMN) (Recap) | RefinedWeb |
| PT-2 | (CC+SA+IMN) (Recap) | (CC+SA+IMN) (Recap) | RefinedWeb |
| SFT | JD+T2I | SF-LLaVA1.5 (Image Mixture) [90] | RefinedWeb |
| DPO | Preference Data | – | – |
| *CoT-V* | – | *CoT-V* data | – |

#### E.0.5 *CoT-V* Post-Training Data

We sample 20K preference data from PARM and T2I-Comp in Appendix E.0.4 to construct our *CoT-V* post-train data and use Prompt. 1 to encourage *UniGen* to generate CoT reasoning during training. To supervise the training process, we construct the CoT reasoning labels based on decomposed

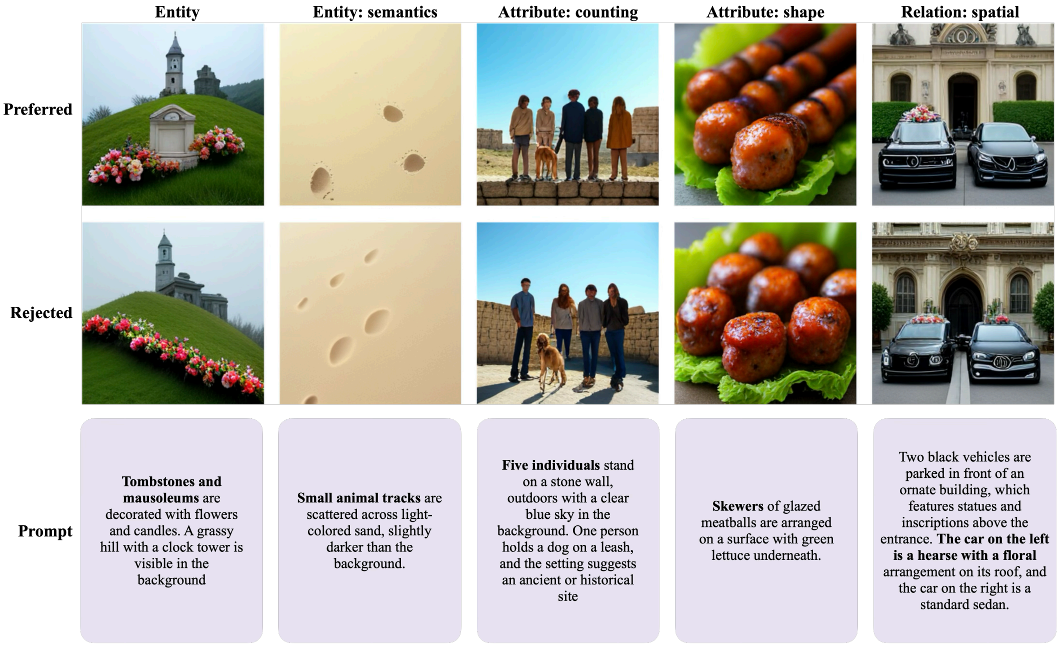

Entity
Entity: semantics
Attribute: counting
Attribute: shape
Relation: spatial
Preferred
Rejected
Prompt
Tombstones and mausoleums are decorated with flowers and candles. A grassy hill with a clock tower is visible in the background
Small animal tracks are scattered across light-colored sand, slightly darker than the background.
Five individuals stand on a stone wall, outdoors with a clear blue sky in the background. One person holds a dog on a leash, and the setting suggests an ancient or historical site
Skewers of glazed meatballs are arranged on a surface with green lettuce underneath.
Two black vehicles are parked in front of an ornate building, which features statues and inscriptions above the entrance. The car on the left is a hearse with a floral arrangement on its roof, and the car on the right is a standard sedan.

Figure 6: **Visual examples of our generated preference data for DPO training.**

atomic question-answer pairs corresponding to visual facts presented in the image. For PARM, we separate each prompt into fine-grained sub-questions according to the templates originally used for generating the prompt. Rules of GenEval are used to label each sub-question corresponding to the image with *yes* or *no*. For T2I-Comp, we directly use the decomposed question-answers from the preference data. The final answer is *yes* if all the sub-questions are answered with *yes* and it is *no* otherwise. To form the CoT label, the separated question-answers are treated as a thinking process enclosed within special tokens `<think_start><think_end>`, and the final answer resides within `<answer_start><answer_end>`.

# F More Qualitative Results

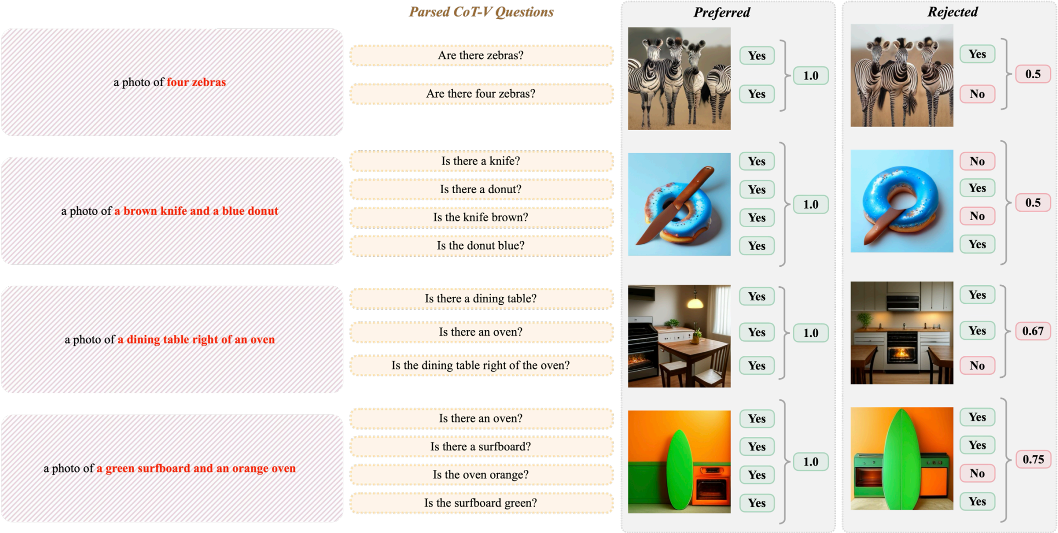
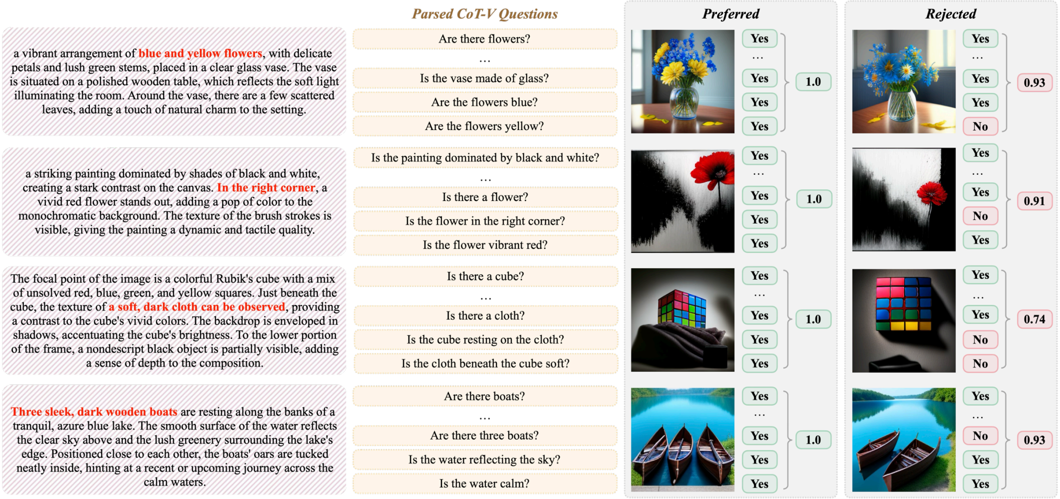


Figure 7: **Successful examples and *CoT-V* verification on GenEval.**



a vibrant arrangement of **blue and yellow flowers**, with delicate petals and lush green stems, placed in a clear glass vase. The vase is situated on a polished wooden table, which reflects the soft light illuminating the room. Around the vase, there are a few scattered leaves, adding a touch of natural charm to the setting.

Are there flowers?
…
Is the vase made of glass?
Are the flowers blue?
Are the flowers yellow?

a striking painting dominated by shades of black and white, creating a stark contrast on the canvas. **In the right corner**, a vivid red flower stands out, adding a pop of color to the monochromatic background. The texture of the brush strokes is visible, giving the painting a dynamic and tactile quality.

Is the painting dominated by black and white?
…
Is there a flower?
Is the flower in the right corner?
Is the flower vibrant red?

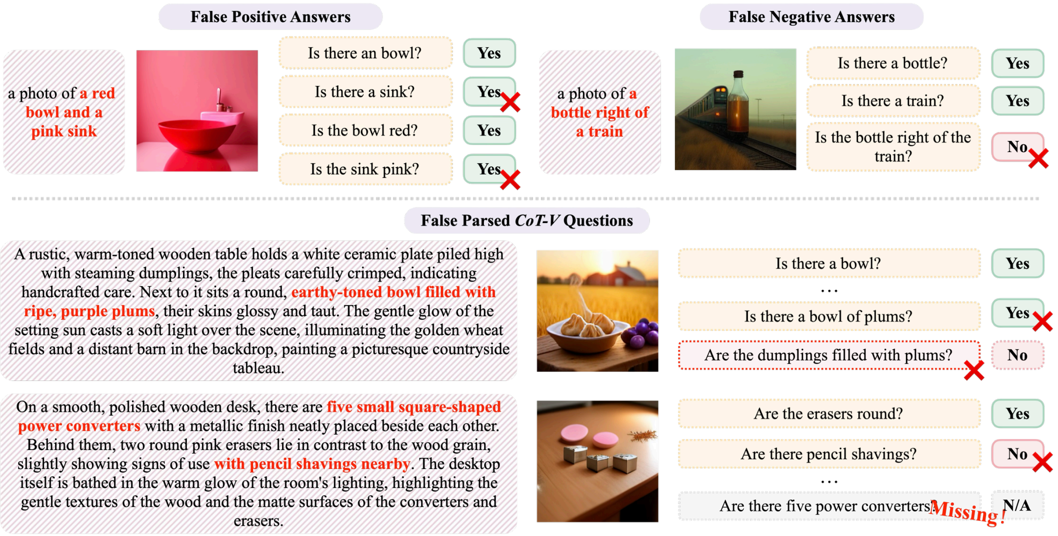
The focal point of the image is a colorful Rubik's cube with a mix of unsolved red, blue, green, and yellow squares. Just beneath the cube, the texture of **a soft, dark cloth can be observed**, providing a contrast to the cube's vivid colors. The backdrop is enveloped in shadows, accentuating the cube's brightness. To the lower portion of the frame, a nondescript black object is partially visible, adding a sense of depth to the composition.

Is there a cube?
…
Is there a cloth?
Is the cube resting on the cloth?
Is the cloth beneath the cube soft?

**Three sleek, dark wooden boats** are resting along the banks of a tranquil, azure blue lake. The smooth surface of the water reflects the clear sky above and the lush greenery surrounding the lake's edge. Positioned close to each other, the boats' oars are tucked neatly inside, hinting at a recent or upcoming journey across the calm waters.

Are there boats?
…
Are there three boats?
Is the water reflecting the sky?
Is the water calm?

Figure 8: **Successful examples and *CoT-V* verification on DPG-Bench.**



a photo of **a red bowl and a pink sink**

Is there an bowl?
Is there a sink?
Is the bowl red?
Is the sink pink?

a photo of **a bottle right of a train**

Is there a bottle?
Is there a train?
Is the bottle right of the train?

A rustic, warm-toned wooden table holds a white ceramic plate piled high with steaming dumplings, the pleats carefully crimped, indicating handcrafted care. Next to it sits a round, **earthy-toned bowl filled with ripe, purple plums**, their skins glossy and taut. The gentle glow of the setting sun casts a soft light over the scene, illuminating the golden wheat fields and a distant barn in the backdrop, painting a picturesque countryside tableau.

Is there a bowl?
…
Is there a bowl of plums?
Are the dumplings filled with plums?

On a smooth, polished wooden desk, there are **five small square-shaped power converters** with a metallic finish neatly placed beside each other. Behind them, two round pink erasers lie in contrast to the wood grain, slightly showing signs of use **with pencil shavings nearby**. The desktop itself is bathed in the warm glow of the room's lighting, highlighting the gentle textures of the wood and the matte surfaces of the converters and erasers.

Are the erasers round?
Are there pencil shavings?
…
Are there five power converters?

Figure 9: **Failure cases on Geneval and DPG-Bench.** The **top half** of the image shows failed examples of *CoT-V* on short prompts. The **bottom half** of the image shows additional cases with bad or missing questions when *CoT-V* parses the complicated and long prompts.

We present qualitative results in Fig. 7 and Fig. 8. They indicate *CoT-V*'s effectiveness on selecting images that accurately convey the entities, color, counting and spatial relation. However, as failure cases shown in Fig. 9, *CoT-V* may struggle with hallucination in more difficult cases. Particularly, we acknowledge that *UniGen* still falls short of generating an accurate reasoning process given free-form complex prompts. We posit that scaling up the model size or improving our CoT training via reinforcement learning algorithms could improve the capability of reasoning and image generation, and consequently enhance the overall performance of *CoT-V*.