

DreamOmni2: Multimodal Instruction-based Editing and Generation

Bin Xia^{1,4} Bohao Peng¹ Yuechen Zhang¹ Junjia Huang⁴,
Jiyang Liu⁴ Jingyao Li¹ Haoru Tan³ Sitong Wu¹ Chengyao Wang¹,
Yitong Wang⁴ Xinglong Wu⁴ Bei Yu¹ and Jiaya Jia²
¹CUHK ²HKUST ³HKU ⁴ByteDance Inc



<https://github.com/dvlab-research/DreamOmni2>

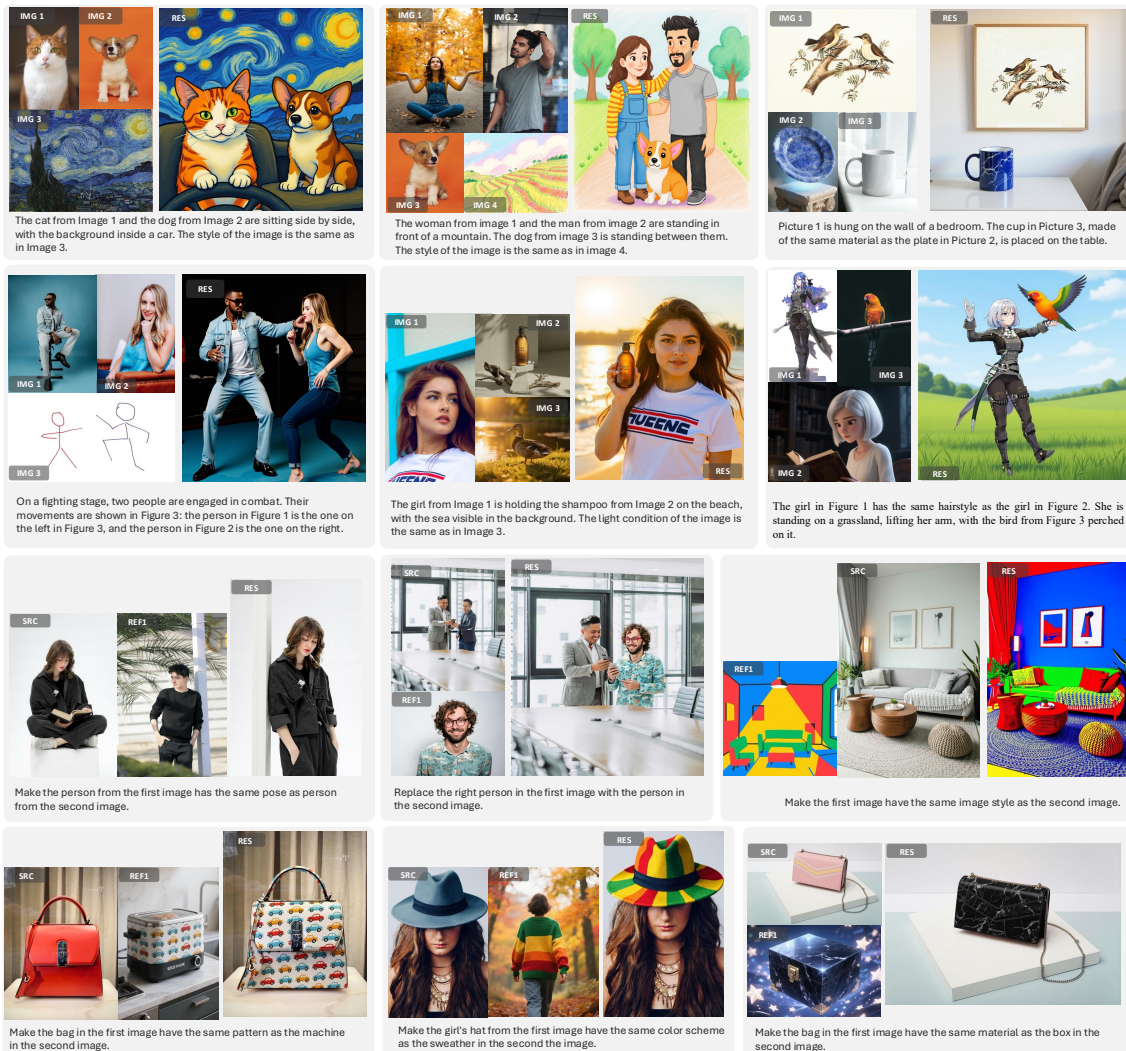


Figure 1. The overview: multimodal instruction-based editing and generation, extending beyond concrete objects to abstract attributions.

Abstract

Recent advancements in instruction-based image editing and subject-driven generation have garnered significant attention, yet both tasks still face limitations in meeting prac-

tical user needs. Instruction-based editing relies solely on language instructions, which often fail to capture specific editing details, making reference images necessary. Meanwhile, subject-driven generation is limited to combining concrete objects or people, overlooking broader, abstract

concepts. To address these challenges, we propose two novel tasks: multimodal instruction-based editing and generation. These tasks support both text and image instructions and extend the scope to include both concrete and abstract concepts, greatly enhancing their practical applications. We introduce DreamOmni2, tackling two primary challenges: data creation and model framework design. Our data synthesis pipeline consists of three steps: (1) using a feature mixing method to create extraction data for both abstract and concrete concepts, (2) generating multimodal instruction-based editing training data using the editing and extraction models, and (3) further applying the extraction model to create training data for multimodal instruction-based editing. For the framework, to handle multi-image input, we propose an index encoding and position encoding shift scheme, which helps the model distinguish images and avoid pixel confusion. Additionally, we introduce joint training with the VLM and our generation/editing model to better process complex instructions. In addition, we have proposed comprehensive benchmarks for these two new tasks to drive their development. Experiments show that DreamOmni2 achieved impressive results.

1. Introduction

Recent advancements in unified generation and editing models [11, 17] have gained significant attention and praise in the market. The success of these models can be attributed to several factors: (1) They greatly improve user experience by simplifying the process, allowing users to perform various design tasks within a single model without the need to switch between different ones. (2) Unified models reduce deployment costs for service providers. (3) Academically, they contribute to the exploration of AGI and world models, enabling the accurate understanding of user instructions and the creation or modification of real-world visual content.

Current released works [1, 6, 27] mainly focus on instruction-based editing and subject-driven generation with a text prompt and a single source image input, but both have limitations in application and advancing intelligence. (1) For instruction-based editing [2, 15, 32], instructions alone often fail to fully capture the user’s intent. For example, when a user says, “make the bag in the image have the same pattern as the dress in the given image,” it’s difficult to describe the complex pattern of “dress” with words. Thus, accurate editing requires multimodal instructions, including reference images and text. Notably, this challenge involves not only modifying objects but also any abstract attributes, such as texture, material, posture, hairstyle, and design style, which are difficult to describe with words. (2) Subject-driven generation models [29, 33] and commercial unified models [11] mainly generate content from specific concrete objects or people, with limited research on refer-

encing more general abstract attributes from input images.

To create a more intelligent and all-encompassing unified creation tool, we propose DreamOmni2. The biggest challenge lies in the training data, so we introduce a comprehensive data pipeline for multimodal instruction-based editing and generation, consisting of the following steps (Fig. 2): (1) We propose a feature mixing scheme to exchange attention features between two batches, allowing the model to generate pairs of images with the same abstract attribute or concrete object. Compared to the previous diptych method [29] for generating image pairs, our scheme achieves a higher success rate, produces images with greater resolution, and completely eliminates any content blending at the edges when the pair of images is split. (2) Using the pairs generated in Step 1, we train a generation-editing model as an extraction model. This model extracts concrete objects or abstract attributions from the given image and generates another based on instructions. Compared to previous methods [4, 29] relying on segmentation and detection, our extraction model offers three key advantages: it can handle abstract concepts, occluded objects, and generate more diverse reference images. We then generate multimodal instruction-based editing training data, which includes a target image, a source image, an editing instruction, and multiple reference images. We use a text-to-image (T2I) model to generate a target image based on multiple keywords or select one from a real image database. The extraction model then generates reference images for one of the keywords. Additionally, we use an instruction-based editing model [1] to transform the content defined by the selected keyword into something different, obtaining the source image. (3) We create multimodal instruction-based generation data by applying the extraction model to generate several reference images based on keywords from the source images created in Step 2. Thus, we build data for generating images from multiple reference images.

Furthermore, the current SOTA unified generation and editing models [1] still cannot handle multiple image inputs. To this end, we propose the Dreamomni2 framework. First, we propose an index encoding and position encoding shift scheme. Index encoding helps the model identify the input image’s index, improving its understanding of the referenced image in the instructions. Position encoding is shifted based on previous inputs, preventing pixel confusion and the copy-and-paste effect in the generated results. In addition, we propose a joint training scheme for the generation/editing model and VLM. While instructions in generation and editing models are typically simple, real-world user instructions are often irregular and logically complex. A VLM pre-trained on large-scale corpora can better understand these complex intentions, translating instructions into a form the model can comprehend, significantly improving performance in real-world scenarios. Furthermore, for these

two new tasks, we have built the DreamOmni2 benchmark using real image data. This allows for a more accurate assessment of the model’s generalization and performance in real-world scenarios. Our main contributions are fourfold:

- We propose two highly practical tasks: multimodal instruction-based editing and generation guided by any concrete or abstract concept. Introducing these two tasks makes current unified generation and editing models smarter and more versatile creative tools.
- We propose a three-stage data pipeline to create a high-quality, comprehensive multimodal instruction-based editing and generation dataset.
- We present the DreamOmni2 framework, which employs index encoding and position encoding shifts for multi-reference image inputs, along with joint training of the generation/editing model and VLM to enhance instruction understanding.
- For these two new tasks, we propose a DreamOmni2 benchmark built from real image data, with experiments showing its effectiveness in real-world scenarios.

2. Related Work

Instruction-based Editing refers to modifying an image based on a user’s language instruction [5, 6, 6, 7, 22, 30]. The main challenge of this task lies in the creation of high-quality and accurate editing datasets. As a pioneering work, InstructP2P [2] introduced an instruction-based image editing dataset by fine-tuning GPT-3 and Prompt-to-Prompt [12] with SD 1.5 [20]. Since then, many other approaches [9, 26, 32, 36] for creating datasets have emerged, such as employing people to create data, using inpainting methods, collage-based methods, and using different expert models. Recently, DreamVE [31] has unified instruction-based image and video editing. However, language-based editing is limited, as many details in real-world scenarios can’t be captured with words, requiring reference images for better description. To this end, we propose multimodal instruction-based editing, enabling guidance from concrete objects or abstract attributes in reference images. This makes unified image generation and editing models [1, 27] more comprehensive and practical.

Subject-driven Generation has been extensively studied. Methods like Dreambooth [21] and textual inversion [8] fine-tune models on multiple images of the same subject, enabling subject-driven generation. However, this requires users to prepare several images and perform fine-tuning for each new subject, which is not user-friendly. Later approaches like IP-adapter [34] and BLIP-diffusion [14] used visual encoders to compress the subject of a reference image into a vector and inject it into a diffusion model, enabling subject-driven generation without fine-tuning. IC LoRA [13] and Ominicontrol [23] further explored the inherent image reference capabilities of

DIT models [18]. Recently, unified generation and editing models [1, 27, 32, 33] have adopted the simple approach of encoding reference images as visual tokens, concatenating them with text and noise tokens, and feeding them into the DIT model. However, prior methods focus mainly on concrete objects, limiting their ability to capture broader abstract concepts. In this paper, we propose multimodal instruction-based generation, a task that enables referencing any concrete objects or abstract attributes in reference images to generate new ones. This extends the scope of subject-driven generation and enhances its practicality.

3. Methodology

3.1. Synthetic Data

Multimodal instruction-based editing and generation are new tasks, with the main challenge being the lack of training data. For multimodal instruction-based editing, the previous data creation pipeline [2, 26] involves generating triplets of instructions, source images, and target images. However, this approach does not allow for creating data that incorporates reference images as a condition for editing. For multimodal instruction-based generation, the previous subject-generation data pipeline [4, 29] relies on segmentation detection models to create reference images. This approach makes it difficult to synthesize data for generating reference abstract attributions or occluded concrete objects.

To address the training data problem for these two tasks, we propose a comprehensive synthetic data pipeline. Specifically, as illustrated in Fig. 2, our approach consists of three stages. In the first stage, we introduce a feature mixing scheme, where a dual-branch structure is employed to simultaneously generate both the source image and the target image as follows:

$$\text{Attn}_{tar}(\vec{Q}, \vec{K}, \vec{V}) = \text{softmax}\left(\frac{\vec{Q}\vec{K}^\top}{\sqrt{d}}\right)\vec{V}, \quad (1)$$

where $Q = [Q_{tar}^n; Q_{tar}^t]$, $K = [K_{tar}^n; K_{tar}^t; K_{src}^n]$, and $V = [V_{tar}^n; V_{tar}^t; V_{src}^n]$. Q_{tar}^t , K_{tar}^t , and V_{tar}^t are the text features from the target branch, while Q_{tar}^n , K_{tar}^n , and V_{tar}^n are the noise features from the target branch. K_{src}^n and V_{src}^n are the noise features from the source branch at the same layer as the K_{tar}^n and V_{tar}^n . $[\cdot; \cdot]$ indicates token (or called length) dimension concatenation.

Our feature mixing scheme leverages the model’s inherent T2I capability to generate paired training data. The pair prompts used for generating data are created by listing all possible attribution keywords, and then synthesizing the pair prompts using DouBao [3] based on these keywords. Compared to the previous UNO [29] diptych generation method, our feature mixing scheme has several clear advantages: **(1)** The diptych method halves the image resolution by forcing two images into one, while Feature Mixing generates in two branches without reducing resolution. **(2)** The

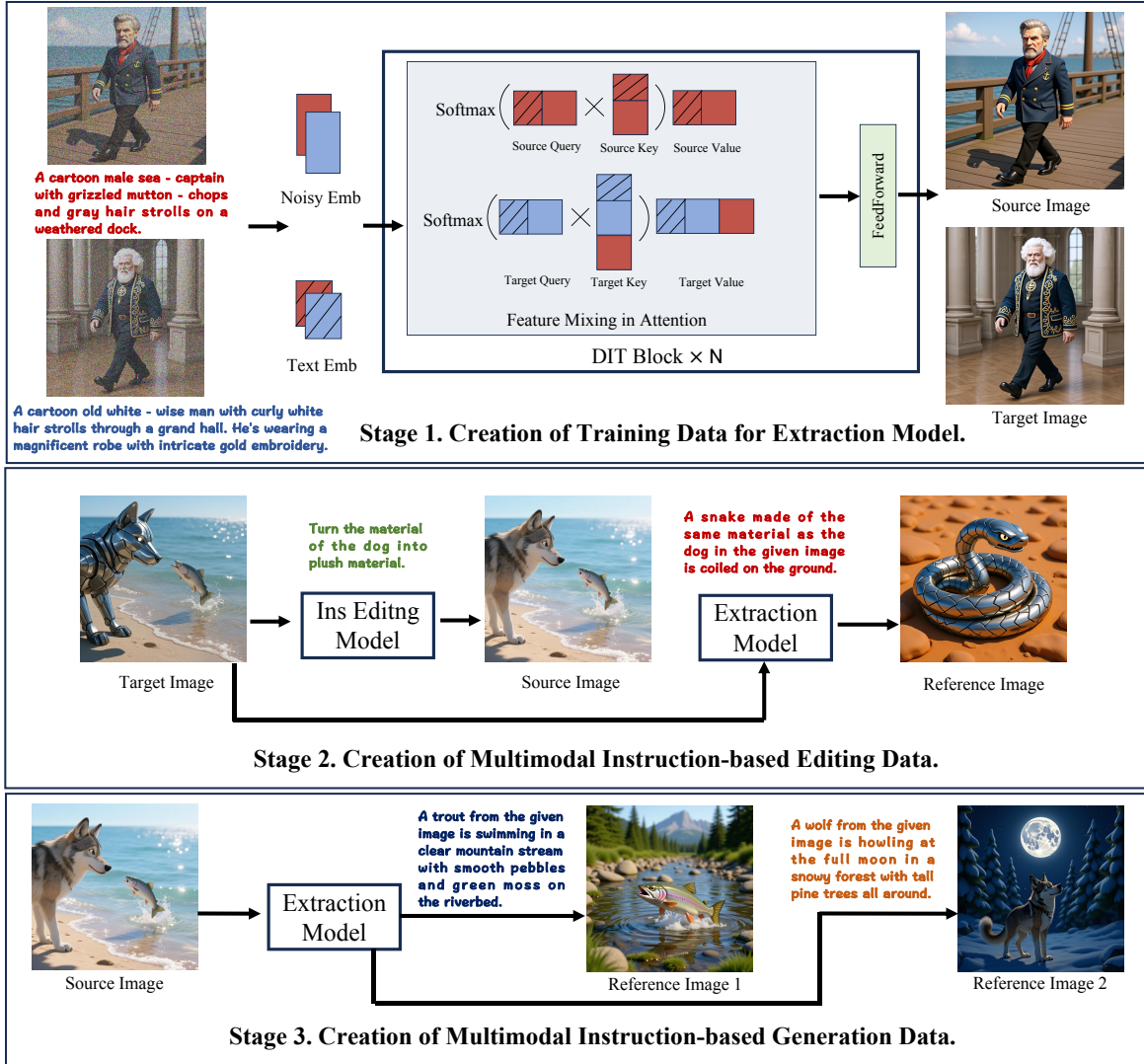


Figure 2. The overview of DreamOmni2’s training data construction. (1) In stage 1, we use a feature mixing scheme to leverage the base model’s T2I capabilities, creating high-quality data pairs with concrete objects and abstract attributes. (2) In stage 2, we generate multimodal instruction-based editing data. Using stage 1 data, we train an extraction model to simulate objects or attributes in the target image and generate a reference image based on instructions. Additionally, we use an instruction-based editing model to modify the extracted objects or attributes in the target image to be different, creating the source image. This generates training pairs from reference and source images to the target image. (3) In stage 3, we extract objects from stage 2’s source images to create new reference images, forming training data for generating target images from reference images.

diptych approach often misplaces the dividing line, leading to content blending. Our method avoids this issue. (3) Data generated by the feature mixing scheme is of higher quality and accuracy than that from the diptych approach. Then, we use the data to train extraction models. Our training data not only enhances the base model [1]’s ability to extract concrete objects but also enables it to capture abstract concepts, a capability it previously lacked. We use the data generated in stage 1 to train the extraction model. Specifically, we input the source image along with a simple description of the target image (for the attribute or person to be extracted, only indicating to refer to the source image).

The extraction model then learns to transfer a certain attribute or appearance from the source image onto the target image for extraction.

Afterward, as shown in Fig. 2 stage 2, we create multimodal instruction-based editing data. Specifically, we first create target images, using both T2I model-generated data and real images. For T2I-generated images, we randomly select diverse element keywords (e.g., objects or attributes) and use an LLM to compose a prompt, which the T2I model then uses to generate the target image. For real images, we directly use a VLM to extract keywords. T2I data is more flexible, allowing any concept combination, while real im-

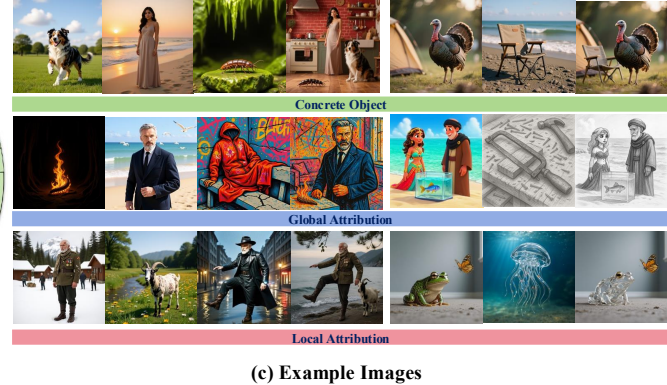
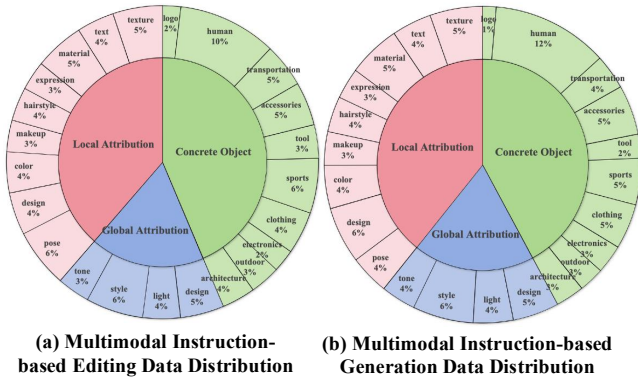


Figure 3. Data distribution and samples for multimodal instruction-based editing and generation training data. Our dataset is comprehensive and diverse, including the generation and editing of concrete objects as well as abstract attributions, such as local and global attributions.

ages reflect natural distributions. Thus, we combine both types of data. Next, using the extraction model trained in stage 1, we extract an object or attribution from the target image based on a selected keyword to create a reference image. We then apply instruction-based editing model [1] to alter the selected keyword in the target image, obtaining the source image. Finally, we use LLM to generate the editing instructions, forming a training tuple of the source image, instruction, reference image, and target image.

After that, as shown in Fig. 2 stage 3, we create multimodal instruction-based generation data. We use the extraction model to extract keywords from the source image in stage 2, generating reference images. By combining these with the reference images from stage 2, we can obtain training tuples consisting of multiple reference images, an instruction, and a target image.

Our created dataset is shown in Fig. 3. Our dataset includes both real and synthetic target data, covering a wide range of object categories for generation and editing, including various abstract attributions and concrete objects. Additionally, we provide a comprehensive set of reference images, with cases ranging from one to five references, enabling the model to handle a wide variety of tasks. We created 116K multimodal instruction-based editing data and 93K generation data.

3.2. Framework and Training

The unified generation and editing base model [1] can only process a single input image. To this end, we propose the DreamOmni2 framework (Fig. 4). In multimodal instruction-based tasks, users typically reference images as “image 1”, “image 2” for convenience. However, in DIT, positional encoding alone cannot accurately distinguish the index of reference images. Therefore, we solve this by adding an index encoding to positional channels. Although index encoding helps distinguish reference images, we found that the position encoding still requires an offset based on the size of the previously input reference images.

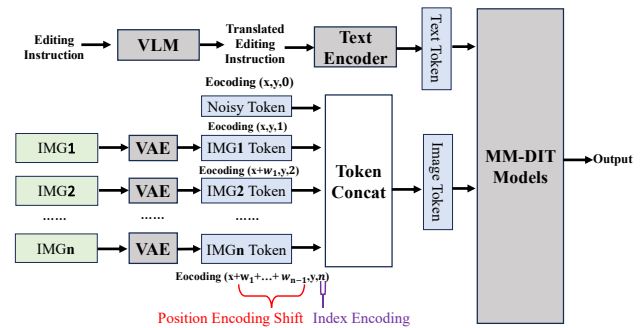


Figure 4. The framework of DreamOmni2. We introduces Position Encoding Shift and Index Encoding scheme to resolve issues of multi-reference image confusion and image identification. Additionally, we incorporate VLM to help the model better understand complex user instructions in real-world scenarios.

By adding this offset to the position encoding, we observed a reduction in copy-and-paste artifacts and pixel confusion between reference images.

Currently, training instructions for generation and editing models are usually well-structured with a fixed format. However, real-world user instructions are often irregular or logically inconsistent, creating a gap that can hinder the model’s understanding and reduce performance. To address this, we propose joint training of the VLM and generation models, enabling the VLM to interpret complex user instructions and output them in the structured format used in training, helping the editing and generation model better understand user intent. For multimodal instruction-based editing, the predefined output format combines user instructions with refined image descriptions, while for multimodal instruction-based generation, the VLM directly outputs a refined image description.

During training, we fine-tune Qwen2.5-VL [25] 7B to learn the predefined standard output format, with a learning rate of 1×10^{-5} , using approximately 10 A100 hours. We then train the editing and generation models using LoRA on Flux Kontext [1] to perform multimodal instruction-based

Table 1. DreamOmni2 vs. related benchmarks.

Benchmarks	Task Type	Num Reference	Editing Target	
			Concrete Object	Abstract Attribution
DreamBooth [21]	Generation	Single	✓	✗
OmniContext [28]	Generation	Multiple	✓	✗
DreamOmni2 (Ours)	Generation & Editing	Multiple	✓	✓

editing and generation with the predefined standard instruction format. For the specific predefined standard output format example, please refer to the supplementary. Notably, by using LoRA for training, we can retain the original instruction-editing capabilities of Kontext. As soon as a reference image is detected, our LoRA is activated, seamlessly integrating multimodal instruction-based editing and generation into the unified model. Additionally, we train LoRA for generation and editing separately, as the distinction between generation and editing lies in whether the consistency of the source image is preserved. Since instructions often do not clarify whether the user intends to edit or generate, separate training allows users to make their own choice. Both DreamOmni2 editing and generation LoRA are trained on a batch size of 16 and a learning rate of 5×10^{-6} , consuming about 384 A100 hours.

3.3. Benchmark and Evaluation

Currently, no benchmark exists for multimodal instruction-based editing and generation. As shown in Tab. 1, DreamBooth [21] only supports single-image generation. Although OmniContext [28] includes some multi-reference testing cases, it focuses solely on concrete object combinations and does not evaluate multimodal instruction-based editing or the inclusion of abstract attributes. To address this, we propose the DreamOmni2 benchmark to drive progress in these areas. Our benchmark is comprehensive, consisting of real images to accurately assess the model’s performance in real-world scenarios. The test cases cover a variety of categories, including the reference generation and editing of abstract attributions (global and local) and concrete objects. More details about our DreamOmni2 benchmark can be found in the supplementary.

Traditional metrics, such as DINO [35], are inadequate for evaluating the complex instruction editing and generation tasks in current scenarios [19]. Recent works like Step1x-Edit [15] and Kontext [1] have used VLMs or human evaluation. Our multimodal instruction tasks are even more complex, requiring evaluation based on VLMs. Thus, we propose a systematic evaluation framework focusing on three aspects: (1) whether the generated results accurately follow the instructions, (2) consistency in human appearance, objects, and abstract attributions, and (3) the absence of significant visual distortions. Only if three criteria are met do we consider the edit or generation successful, reflecting real-world use. We apply the same criteria to human evaluation of DreamOmni2 and show that VLM-based results closely match human evaluation. The system prompt for VLM evaluation is in the supplementary materials.

Table 2. Quantitative comparison of multimodal instruction-based editing. We use Gemini [10] and Doubao [3] to evaluate the success editing ratio of different models on concrete objects and abstract attributions, respectively. In addition, “Human” refers to professional engineers assessing the editing success rates of all models.

Method	Concrete Object			Abstract Attribution		
	Gemini↑	Doubao↑	Human↑	Gemini↑	Doubao↑	Human↑
GPT-4o [17]	0.6829	0.7805	0.5610	0.7195	0.7439	0.5793
Nano Banana [11]	0.6829	0.7073	0.5366	0.6463	0.5488	0.3293
UNO [29]	0.0000	0.0244	0.0000	0.0061	0.0183	0.0000
DreamO [16]	0.0244	0.0732	0.0000	0.0183	0.0183	0.0000
Omnigen2 [28]	0.2195	0.2927	0.2927	0.0427	0.0793	0.0305
Qwen-Image-Edit [27]	0.0976	0.1463	0.0244	0.0244	0.0183	0.0000
Kontext [1]	0.0488	0.1220	0.0976	0.0183	0.0122	0.0122
Qwen-Image-Edit-2509 [27]	0.2683	0.2927	0.2195	0.0488	0.1159	0.0427
DreamOmni2 (Ours)	0.5854	0.6585	0.6098	0.5854	0.6280	0.6829

4. Experiments

Evaluation on Multimodal Instruction-based Image Editing. As shown in Tab. 2, we compare several competitive models that natively support multiple image inputs, such as DreamO [16], Omnigen2 [28], and Qwen-Image-Edit-2509 [27]. Although Kontext [1] and Qwen-Image-Edit [27] do not natively support multiple image inputs, we applied the method from Diffusers [24], which combines multiple images into one input. We also compared closed-source commercial models, such as Nano Banana [11] and GPT-4o [17]. We tested editing examples with concrete objects and abstract attributions on DreamOmni2 benchmark. The models were evaluated for success rates by Gemini 2.5 [10] and Doubao 1.6 [3], and several professional engineers manually assessed the results. The evaluation criteria can be found in Sec 3.3. For manual evaluation, we assess each case with 5 evaluators, and the case is considered passed if more than 3 evaluators approve it. As shown in Tab. 2, our DreamOmni2 achieved the best performance in human evaluations. In VLM tests, DreamOmni2 significantly outperformed open-source models and achieved results close to those of commercial models. In fact, GPT-4o and Nano Banana often introduced unintended changes or inconsistencies in the edited attribution, which were not aligned with the reference images. These issues are difficult for VLMs to detect accurately. Additionally, GPT-4o caused the edited images to appear yellowed.

Qualitative results are shown in Fig. 5, where we present visualizations of editing cases involving various concrete objects and abstract attributes. It is clear that DreamOmni2 produces more accurate edits with better consistency. This further demonstrates the impressive performance of our approach in multimodal instruction-based editing.

Evaluation on Multimodal Instruction-based Image Generation. As shown in Tab. 3, our method outperforms the commercial model Nano Banana in both human evaluations and assessments by Doubao 1.6 and Gemini 2.5, achieving results comparable to GPT-4o. Compared to open-source models like DreamO, Omnigen2, and Qwen-

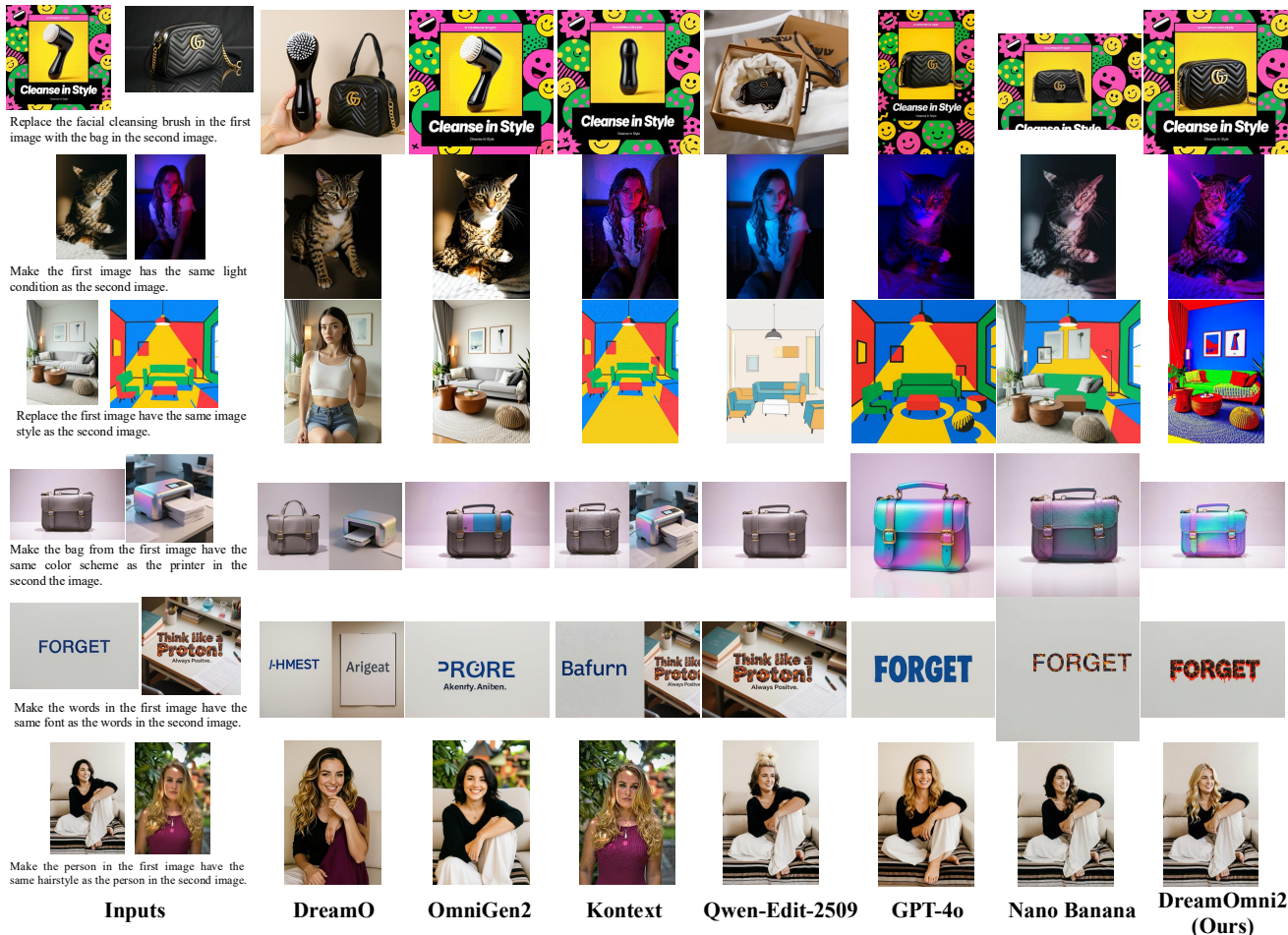


Figure 5. Visual comparison of multimodal instruction-based editing. Compared to other competitive methods and even closed-source commercial models (GPT-4o and Nano Banana), DreamOmni2 shows more accurate editing results and better consistency.

Edit-2509, which focus primarily on generating images with multiple concrete objects, DreamOmni2 still significantly outperforms them in both generation accuracy and object consistency, even within their specialized domains. This further underscores the effectiveness of DreamOmni2 in multimodal instruction-based generation.

Quantitative results, as shown in Fig. 6, indicate that open-source models struggle with generating abstract attributes. Even in generating concrete objects, which these models are specifically optimized for, DreamOmni2 outperforms them in both instruction adherence and object consistency. Furthermore, DreamOmni2 even outperforms the commercial model Nano Banana.

Joint Training. As shown in Tab. 4, we validate the impact of joint training of generation or editing and VLM. Scheme 1 represents the base model, Kontext. In Scheme 2, we train the generation and editing models with basic instructions without introducing VLM. In Scheme 3, we train the VLM with standard descriptive instructions and input the VLM-generated descriptions into Kontext. In Scheme 4, we perform joint training of the VLM and our genera-

Table 3. Quantitative comparison of multimodal instruction-based generation. We use Gemini [10] and Doubao [3] to evaluate the success editing ratio on concrete objects and abstract attributions, respectively. In addition, ‘‘Human’’ refers to professional engineers assessing the editing success rates of all models.

Method	Concrete Object			Abstract Attribution		
	Gemini [†]	Doubao [†]	Human [†]	Gemini [†]	Doubao [†]	Human [†]
GPT-4o [17]	0.6250	0.6250	0.5610	0.6889	0.6333	0.5793
Nano Banana [11]	0.5000	0.5417	0.5366	0.5556	0.5111	0.3293
UNO [29]	0.0000	0.0000	0.0000	0.0333	0.0556	0.0000
DreamO [16]	0.0417	0.0833	0.0000	0.0667	0.0222	0.0000
OmniGen2 [28]	0.2083	0.2500	0.2927	0.1000	0.0778	0.0305
Qwen-Image-Edit [27]	0.0417	0.1250	0.0244	0.0889	0.1000	0.0000
Kontext [1]	0.2500	0.3750	0.0976	0.0556	0.1222	0.0122
Qwen-Image-Edit-2509 [27]	0.1250	0.2917	0.2195	0.1111	0.1556	0.0427
DreamOmni2 (Ours)	0.5833	0.6667	0.6098	0.5778	0.6333	0.6829

Table 4. Joint training for generation or editing models and VLM.

Method	Generation or Editing Model Training	VLM Training	Editing		Generation	
			Concrete Object	Abstract Attribution	Concrete Object	Abstract Attribution
Scheme 1	×	×	0.1220	0.0122	0.3750	0.1222
Scheme 2	✓	×	0.3659	0.3171	0.4583	0.3444
Scheme 3	×	✓	0.2439	0.3415	0.5417	0.4778
Scheme 4 (Ours)	✓	✓	0.6585	0.6280	0.6667	0.6333

tion or editing model on our data. Comparing Scheme 2 with Scheme 1, we see that our data significantly enhances the model’s ability to handle multimodal instruction-based

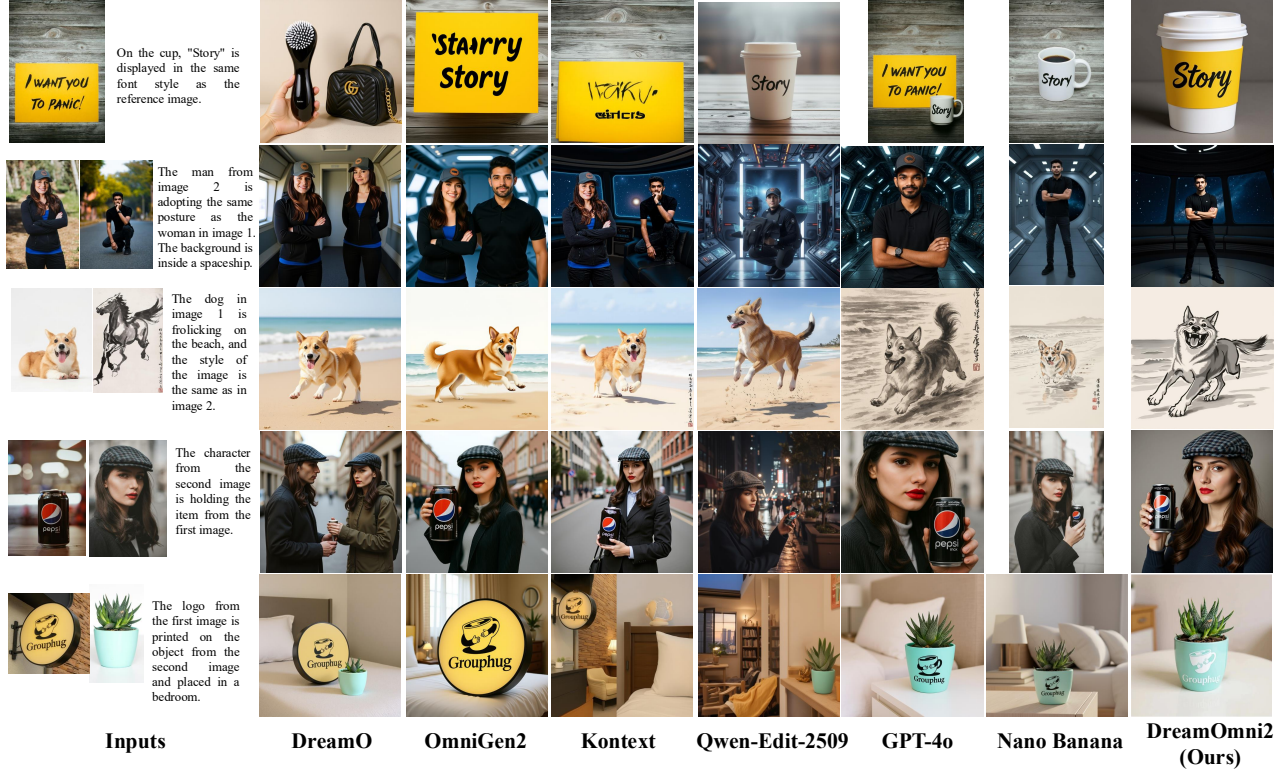


Figure 6. Visual comparison of multimodal instruction-based generation. Our DreamOmni2 significantly outperforms current open-source models and achieves generation results comparable to closed-source commercial models (GPT-4 and Nano Banana).

editing and generation. Comparing Scheme 3 with Scheme 4, we observe that introducing VLM helps the generation and editing models better understand real-world user complex instructions, improving performance. Moreover, our joint training scheme in Scheme 4 outperforms Scheme 2 and Scheme 3, demonstrating its effectiveness.

Index and Position Encoding. As shown in Tab. 5, we compare different encoding schemes to help the model adapt to multiple image inputs. Comparing Scheme 3 and Scheme 1, we find that adding index encoding enables the model to understand which image corresponds to references like "Image 1," "Image 2," and "Image 3" in user instructions, resulting in more accurate generation and editing. Additionally, when comparing Scheme 3 and Scheme 4, we observe that with the inclusion of index encoding, multiple images require position encoding shifts instead of using the same position encoding. This adjustment prevents the copy-and-paste effect and improves the model's editing and generation performance. Therefore, in DreamOmni2, we incorporate index encoding along with position encoding shifts for multiple reference images.

5. Conclusion

Current instruction-based editing relies on language, but it often struggles to clearly describe desired edits. Therefore, reference images are needed to guide the process. Additionally, subject-driven generation models typically focus

Table 5. Different encoding schemes for multiple image inputs.

Method	Index Encoding	Position Encoding Shift	Editing		Generation	
			Concrete Object	Abstract Attribution	Concrete Object	Abstract Attribution
Scheme 1	✗	✗	0.2439	0.2805	0.2917	0.2222
Scheme 2	✗	✓	0.4634	0.5427	0.5417	0.5111
Scheme 3	✓	✗	0.3415	0.3902	0.4167	0.4556
Scheme 4 (Ours)	✓	✓	0.6585	0.6280	0.6667	0.6333

on concrete objects and cannot generate images based on abstract concepts. To this end, we propose two new tasks: multimodal instruction-based editing and generation, where references include both concrete objects and abstract attributions. These tasks face two main challenges: training data and the framework supporting multi-image input. For training data, we introduce a three-stage data synthesis pipeline. In stage 1, we use a feature mixing approach to create data for an extraction model, which can generate images with the same elements (objects or attributes) as the given image. In stage 2, we use the extraction and instruction-based editing models to create multimodal instruction-based editing data. In stage 3, we apply the extraction model to stage 2 data to generate multimodal instruction-based generation data. For the framework, we design an index encoding and position encoding shift scheme to help the model distinguish multiple images and avoid the copy-and-paste effect. We also propose a joint training scheme for the generation/editing model and the VLM, improving the model's ability to understand complex real-world instructions. Extensive experiments show the impressive performance of DreamOmni2.

6. Acknowledgments

This work was supported in part by the Research Grants Council under the Areas of Excellence scheme grant AoE/E-601/22-R.

References

- [1] Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv e-prints*, pages arXiv–2506, 2025. 2, 3, 4, 5, 6, 7
- [2] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, 2023. 2, 3
- [3] ByteDance. Doubao. <https://www.doubao.com/>, 2025. 3, 6, 7
- [4] Bowen Chen, Mengyi Zhao, Haomiao Sun, Li Chen, Xu Wang, Kang Du, and Xinglong Wu. Xverse: Consistent multi-subject control of identity and semantic attributes via dit modulation. *arXiv preprint arXiv:2506.21416*, 2025. 2, 3
- [5] Xi Chen, Zhifei Zhang, He Zhang, Yuqian Zhou, Soo Ye Kim, Qing Liu, Yijun Li, Jianming Zhang, Nanxuan Zhao, Yilin Wang, et al. Unreal: Universal image generation and editing via learning real-world dynamics. In *CVPR*, 2025. 3
- [6] 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, 3
- [7] Rongyao Fang, Chengqi Duan, Kun Wang, Hao Li, Linjiang Huang, Hao Tian, Xingyu Zeng, Rui Zhao, Jifeng Dai, Hongsheng Li, et al. Puma: Empowering unified mllm with multi-granular visual generation. In *CVPR*, 2025. 3
- [8] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022. 3
- [9] Yuying Ge, Sijie Zhao, Chen Li, Yixiao Ge, and Ying Shan. Seed-data-edit technical report: A hybrid dataset for instructional image editing. *arXiv preprint arXiv:2405.04007*, 2024. 3
- [10] Google. Gemini. <https://deepmind.google/models/gemini/>, 2025. 6, 7
- [11] Google. Nano banana. <https://aistudio.google.com/models/gemini-2-5-flash-image>, 2025. 2, 6, 7
- [12] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022. 3
- [13] Lianghua Huang, Wei Wang, Zhi-Fan Wu, Yupeng Shi, Huanzhang Dou, Chen Liang, Yutong Feng, Yu Liu, and Jingren Zhou. In-context lora for diffusion transformers. *arXiv preprint arXiv:2410.23775*, 2024. 3
- [14] Dongxu Li, Junnan Li, and Steven Hoi. Blip-diffusion: Pre-trained subject representation for controllable text-to-image generation and editing. *NeurIPS*, 2023. 3
- [15] Shiyu Liu, Yucheng Han, Peng Xing, Fukun Yin, Rui Wang, Wei Cheng, Jiaqi Liao, Yingming Wang, Honghao Fu, Chunrui Han, et al. Step1x-edit: A practical framework for general image editing. *arXiv preprint arXiv:2504.17761*, 2025. 2, 6
- [16] Chong Mou, Yanze Wu, Wenxu Wu, Zinan Guo, Pengze Zhang, Yufeng Cheng, Yiming Luo, Fei Ding, Shiwen Zhang, Xinghui Li, et al. Dreamo: A unified framework for image customization. *arXiv preprint arXiv:2504.16915*, 2025. 6, 7
- [17] OpenAI. Gpt-4o image generation. <https://openai.com/index/introducing-4o-image-generation/>, 2025. 2, 6, 7
- [18] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023. 3
- [19] Yuang Peng, Yuxin Cui, Haomiao Tang, Zekun Qi, Runpei Dong, Jing Bai, Chunrui Han, Zheng Ge, Xiangyu Zhang, and Shu-Tao Xia. Dreambench++: A human-aligned benchmark for personalized image generation. *arXiv preprint arXiv:2406.16855*, 2024. 6
- [20] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 3
- [21] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *CVPR*, 2023. 3, 6
- [22] Shelly Sheynin, Adam Polyak, Uriel Singer, Yuval Kirstain, Amit Zohar, Oron Ashual, Devi Parikh, and Yaniv Taigman. Emu edit: Precise image editing via recognition and generation tasks. *arXiv preprint arXiv:2311.10089*, 2023. 3
- [23] Zhenxiong Tan, Songhua Liu, Xingyi Yang, Qiaochu Xue, and Xinchao Wang. Ominicontrol: Minimal and universal control for diffusion transformer. *arXiv preprint arXiv:2411.15098*, 2024. 3
- [24] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, William Berman, Yiyi Xu, Steven Liu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. <https://github.com/huggingface/diffusers>, 2022. 6
- [25] 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 preprint arXiv:2409.12191*, 2024. 5
- [26] Cong Wei, Zheyang Xiong, Weiming Ren, Xeron Du, Ge Zhang, and Wenhui Chen. Omniedit: Building image editing generalist models through specialist supervision. In *ICLR*, 2024. 3
- [27] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025. 2, 3, 6, 7
- [28] Chenyuan Wu, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyang Jiang, Yexin Liu, Junjie

- Zhou, et al. Omnigen2: Exploration to advanced multimodal generation. *arXiv preprint arXiv:2506.18871*, 2025. 6, 7
- [29] Shaojin Wu, Mengqi Huang, Wenxu Wu, Yufeng Cheng, Fei Ding, and Qian He. Less-to-more generalization: Unlocking more controllability by in-context generation. *ICCV*, 2025. 2, 3, 6, 7
- [30] Bin Xia, Shiyin Wang, Yingfan Tao, Yitong Wang, and Jiaya Jia. Llmga: Multimodal large language model based generation assistant. In *ECCV*, 2024. 3
- [31] Bin Xia, Jiyang Liu, Yuechen Zhang, Bohao Peng, Ruihang Chu, Yitong Wang, Xinglong Wu, Bei Yu, and Jiaya Jia. Dreamve: Unified instruction-based image and video editing. *arXiv preprint arXiv:2508.06080*, 2025. 3
- [32] Bin Xia, Yuechen Zhang, Jingyao Li, Chengyao Wang, Yitong Wang, Xinglong Wu, Bei Yu, and Jiaya Jia. Dreamomni: Unified image generation and editing. In *CVPR*, 2025. 2, 3
- [33] Shitao Xiao, Yueze Wang, Junjie Zhou, Huaying Yuan, Xingrun Xing, Ruiran Yan, Chaofan Li, Shuting Wang, Tiejun Huang, and Zheng Liu. Omnigen: Unified image generation. In *CVPR*, 2025. 2, 3
- [34] Hu Ye, Jun Zhang, Sibao Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023. 3
- [35] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022. 6
- [36] Kai Zhang, Lingbo Mo, Wenhui Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. *NeurIPS*, 2024. 3