OmniGen-AR: AutoRegressive Any-to-Image Generation

Junke Wang[1,2], **Xun Wang**[3], **Qiushan Guo**[3], **Peize Sun**[4],
Weilin Huang[3], **Zuxuan Wu**[1,2†], **Yu-Gang Jiang**[1,2]
[1]Institute of Trustworthy Embodied AI, Fudan University
[2]Shanghai Collaborative Innovation Center of Intelligent Visual Computing
[3]Bytedance Seed, [4]The University of Hong Kong

Abstract

Autoregressive (AR) models have demonstrated strong potential in visual generation, offering superior performance with simple architectures and optimization objectives. However, existing methods are typically limited to single-modality conditions, *e.g.*, text, restricting their applicability in real-world scenarios that demand image synthesis from diverse controls. In this work, we present OmniGen-AR, a unified autoregressive framework for Any-to-Image generation. By discretizing various visual conditions through a shared visual tokenizer and text prompts with a text tokenizer, OmniGen-AR supports a broad spectrum of conditional inputs within a single model, including text (text-to-image generation), spatial signals (segmentation-to-image and depth-to-image), and visual context (image editing, frame prediction, and text-to-video generation). To mitigate the risk of information leakage from condition tokens to content tokens, we introduce Disentangled Causal Attention (DCA), which separates the full-sequence causal mask into condition causal attention and content causal attention. It serves as a training-time regularizer without affecting the standard next-token prediction during inference. With this design, OmniGen-AR achieves new state-of-the-art or at least competitive results across a range of benchmark, *e.g.*, 0.63 on GenEval and 80.02 on VBench, demonstrating its effectiveness in flexible and high-fidelity visual generation.

1 Introduction

In recent years, deep generative models [32, 21, 100, 61, 52] have experienced rapid development and revolutionized the way we create visual contents. Among them, autoregressive models (AR) [15, 100, 66, 82] have demonstrated the capability for high-quality image synthesis through sequential token prediction. The superior performance, flexbility, and compatibility with multimodal inputs, position them as competitive alternatives to diffusion models [25, 64, 13, 51, 70].

Despite their potential, existing autoregressive (AR) visual generation methods primarily focus on single-modality conditioning, such as category labels [15, 18, 101, 81, 79] or text prompts [60, 100, 66, 82]. While these models achieve strong performance within the respective domains, they fall short of the versatility required in real-world applications, where visual generation often respond to a diverse set of conditional inputs [106, 107, 89, 94], such as semantic masks, reference images, or history frames. In other words, building a unified AR generative model that accommodates various inputs remains under-explored.

To fill this gap, this work presents OmniGen-AR, an autoregressive framework for **Any-to-Image generation**. In addition to text, OmniGen-AR also supports a wide range of visual conditions

†: corresponding author.

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

Table 1: A system-level comparison between OmniGen-AR and other methods. Compared to OmniGen [94], OmniGen-AR additionally supports video generation.

Method	Type	Condition		
		Text	Ref	Spatial
GLIGEN [40]	Diff	✓	✗	✓
ControlNet [106]	Diff	✓	✗	✓
Uni-ControlNet [107]	Diff	✓	✗	✓
OmniGen [94]	Diff	✓	✓	✓
LLamaGen [66]	AR	✓	✗	✗
SimpleAR [82]	AR	✓	✗	✗
ControlAR [41]	AR	✓	✗	✓
Ours	AR	✓	✓	✓

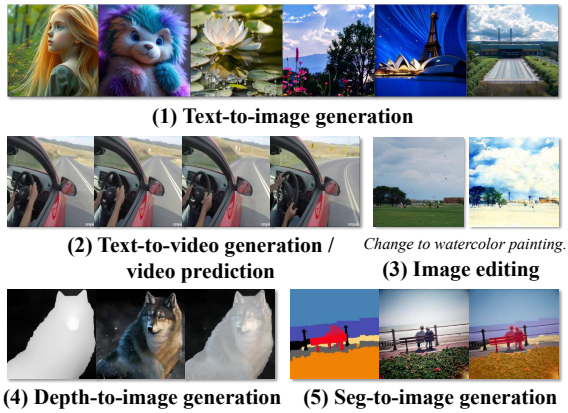


Figure 1: OmniGen-AR could handle 5 types of generation tasks within a single model.

including segmentation masks, depth maps, and reference images, by discretizing them with a shared visual tokenizer. An overview of our method is provided in Table 1 and Figure 1. While this approach allows our model to preserve the simplicity of autoregressive modeling, the serial nature of prediction introduces a potential risk of information leakage from condition tokens to content tokens. This becomes particularly problematic in tasks like image editing and frame prediction, where much of the output remains unchanged relative to the input. Consequently, the model may converge to suboptimal solutions that exploit shortcut patterns between conditioning and prediction signals, instead of producing meaningful and instruction-following results.

To address this, we introduce Disentangled Causal Attention (DCA), which separates the causal attention over the entire sequence into *condition causal attention* and *content causal attention*. This disentanglement prevents the information flow from content tokens to condition tokens while still allowing the latter to retain awareness of their relative positions. During training, we randomly replace the vanilla causal attention with DCA, which regularizes the model by discouraging over-reliance on conditional context and promoting more instruction-compliant predictions. The inference process of our model still follows the standard next-token prediction.

We validate the effectiveness of OmniGen-AR on six representative visual generation tasks, including text-to-image generation, text-to-video generation, frame prediction, image editing, depth-conditioned image generation, and segmentation-conditioned image generation. The results demonstrate that it achieves new state-of-the-art or at least competitive results on the prevalent benchmarks, *e.g.*, 0.63 on GenEval [20] and 80.02 on VBench [29]. OmniGen-AR not only maintains the inherent flexibility and scalability of autoregressive models but also enables the seamless integration of various control signals, providing a unified and effective solution for universal visual generation.

2 Related Work

2.1 Autoregressive Visual Generation

Autoregressive (AR) models have become a popular paradigm in generative modeling for both language [56, 57, 72, 3] and vision [100, 66, 82], owing to their strong capability in modeling complex distributions. Early efforts in AR-based visual generation model images as sequences of pixels [74, 10], which achieves satisfactory results but suffer from inefficiency and limited scalability. Subsequent approaches such as VQ-VAE [75] introduce discrete visual tokenization to autoregressive visual generation, enabling the use of transformer-based language models for image synthesis. These token-based methods significantly improve the generation quality and training stability, attracting a series of work that leverage learned codebooks for autoregressive image generation [60, 15, 100, 66].

Recent works have explored autoregressive generation with continuous representations [39, 108], scale-wise autoregressive modeling [69, 23], and reinforcement-learning for improved generation quality [82]. Despite these advances, existing AR models mainly focus on single-modality conditions

(*e.g.*, text or class labels), restricting their applicability in real-world scenarios requiring multi-modal controls. The most relevant literature is EditAR [49], which also employs autoregressive transformers and support multiple conditional image generation tasks. Ours work differ from them in two aspects: 1) EditAR is specifically designed for image editing and low-level control tasks (*e.g.*, depth-to-image, edge-to-image, segmentation-to-image), while our model is a unified Any-to-Image framework that handles a broader range of input modalities. 2) EditAR aims to improve text-image alignment by introducing distillation loss, while OmniGen-AR hopes to prevent information leakage through DCA, a novel training-time attention mechanism that disentangles condition and content attention paths.

2.2 Diffusion Models for Any-to-Image Generation

Recent advances in diffusion models [52, 14] have significantly improved the quality and controllability of image synthesis from diverse conditioning signals [106, 40, 85, 104]. ControlNet [106] firstly introduces a framework that injects spatial control (*e.g.*, edge maps, segmentation masks) into a pretrained diffusion model without compromising generation quality. It demonstrates strong performance in aligning generated content with spatial priors but still requires separate adapters for each conditional modality. To address this, Uni-ControlNet [107] proposes a unified architecture that supports multiple spatial controls within a single framework by learning a modality-agnostic representation space. It improves the generality and flexibility across tasks, but still requires the separate training procedures for different types of condition. More recently, OmniGen [94] pushes toward general-purpose image generation by unifying diverse tasks, *e.g.*, text-to-image synthesis, image editing, and subject-driven generation, within a diffusion framework that eliminates external modules. Inspired by this, we explore the autoregressive framework for any-to-image generation. We adopt autoregressive modeling as it provides a more natural fit for handling sequential inputs and enabling interleaving generation.

2.3 Unified Models for Multimodal Understanding and Generation

The belief in scaling data and model size [24, 31, 102] has driven the community towards building unified and even general multimodal models [2, 68, 1]. CLIP [55] first demonstrates that large-scale contrastive pretraining on image-text pairs could yield powerful vision-language representations. Subsequent works [38, 80, 88, 12, 4] extend this paradigm to support a broader range of vision-language tasks such as captioning and VQA, across both image and video domains.

LLaVA [44, 43] opens up another chapter, *i.e.*, visual instruction tuning, by aligning pretrained vision encoders [57, 103] with large language models [72] to enabling open-ended multimodal understanding. Recently, Chameleon [67] steps beyond the scope of understanding tasks to unified multimodal understanding and generation, seamlessly integrating both modalities in a token-based framework. Following work improve the design of unified multimodal language models through better multimodal fusion [111, 95, 71] and visual encoding [92, 93]. These advancements showcase the growing potential of general multimodal artificial intelligence, pushing the boundaries of both understanding and generation tasks across multiple domains.

3 Method

Our goal is to unify conditional image generation (*i.e.*, text, spatial, image) within a single autoregressive framework. To this end, we propose OmniGen-AR, which consists of a text and visual tokenizer to discrete various inputs. With this, we model the dependency between multimodal tokens using an autoregressive transformer. The architecture of OmniGen-AR is illustrated in Figure 2.

3.1 Visual and Textual Tokenization

To enable the unified processing and generation of diverse modalities, the first question is how to represent them in a compatible format. Unlike previous work [106, 41] that rely on separate encoders to encode visual condition $V \in \mathbb{R}^{H \times W \times 3}$ (segmentation masks, depth maps, image to be edited) and the image to be generated $X \in \mathbb{R}^{H \times W \times 3}$, we adopt the same visual tokenizer [16] to convert them into discrete visual tokens: $v \in \mathbb{R}^{N_1}$ and $x \in \mathbb{R}^{N_2}$, where $N1 = N2$ denote the sequence length. Here we omit the loop-up and flatten operations for simplicity. While for the textual inputs, we tokenize them with a language model tokenizer [98] to obtain the text tokens $t \in \mathbb{R}^{M}$.

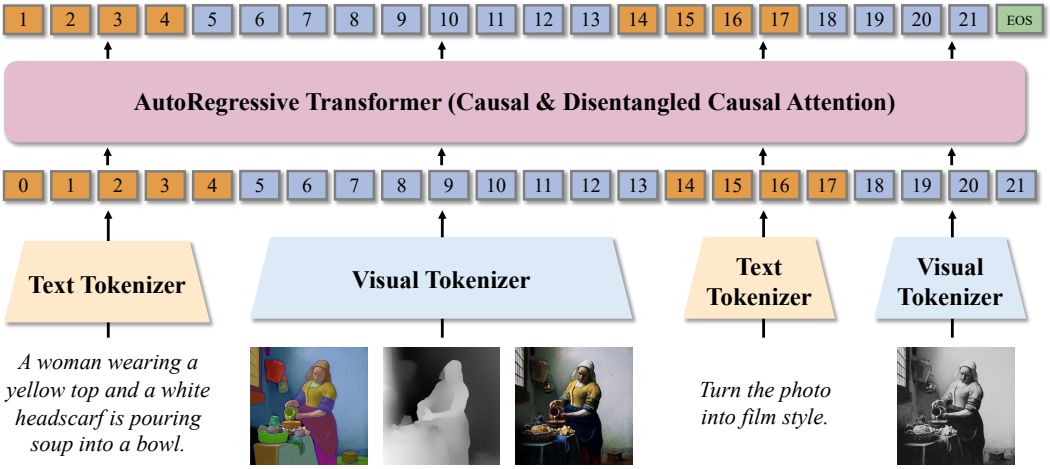


Figure 2: OmniGen-AR consists of a text tokenizer, a visual tokenizer, an autoregressive transformer.

3.2 Autoregressive Transformer for Multimodal Generation

We adopt a decoder-only transformer model for multimodal generation, which consists of stacked attention blocks [76]:

$$\text{Attention}(q, k, v) = \text{softmax}\left(\frac{qk^{\text{T}}}{\sqrt{d_k}} + m\right)v, \tag{1}$$

where q, k, $v \in \mathbb{R}^{N \times d_k}$ represent the query, key, and value embeddings, and $m \in \mathbb{R}^{N \times N}$ is the attention mask. In modern language models [57, 72] and AR-based visual generation models [100, 66, 82], m is usually implemented as a lower triangular matrix to mask out the future positions :

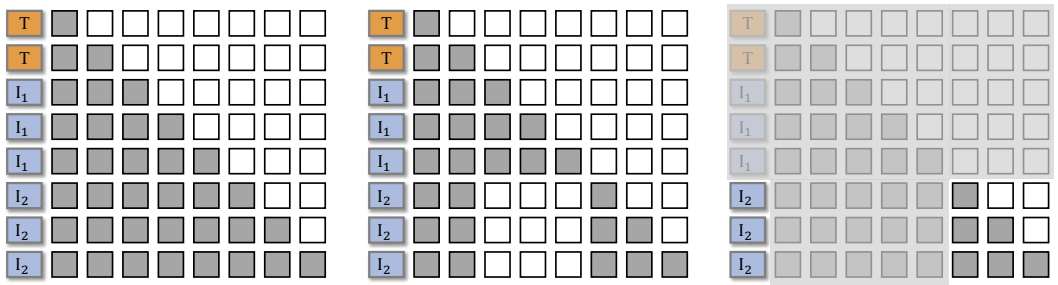
$$m_{i,j} = \begin{cases} 0, & \text{if } j \leq i \\ -\infty, & \text{otherwise} \end{cases} \tag{2}$$

This mechanism ensures that each token attends only to itself and preceding tokens in the sequence, preserving the left-to-right generation order. However, in the context of conditional image generation, the plain causal attention can lead to unintended information leakage: given access to previous condition tokens, the model may learn to exploit trivial correlations between them and the content tokens to be predicted, instead of generating meaningful tokens that follow instructions faithfully [17, 96, 37, 84].

To alleviate this problem, we modify the attention mask to prevent information flow from condition tokens to content tokens, while still preserving autoregressive modeling within each. Taking the image editing task as an example, we concatenate text, condition, and content tokens along the sequence dimension as the token sequence: $[t, v, x] \in \mathbb{R}^L$, where $L = M + N_1 + N_2$, and design the attention mask $m \in \mathbb{R}^{(M+N_1+N_2)\times(M+N_1+N_2)}$ in the following manner:

$$m_{i,j} = \begin{cases} 0, & \text{if } j \leq i \text{ and } (i,j) \in A \cup B \cup C \\ -\infty, & \text{if } i \in C,\ j \in B \\ -\infty, & \text{otherwise} \end{cases} \tag{3}$$
$$\textbf{where } A = [0, M), \quad B = [M, M + N_1), \quad C = [M + N_1, M + N_1 + N_2)$$

Such a masking scheme permits content tokens to attend to preceding text tokens while blocking access to other condition tokens, thereby reducing the risk of shortcut learning. During training, we randomly apply DCA in place of vanilla causal attention as a regularization. Please see Figure 3 for a better illustration. Notably, the proposed DCA differs from classifier-free guidance [26] in its treatment of condition tokens. First of all, content tokens remain aware of positional information, as the condition tokens are not dropped entirely. In addition, DCA is applied only during training and has no impact on the inference process.



(a) plain causal attention mask (b) disentangled causal attention mask (c) classifier-free guidance

Figure 3: Comparison between plain causal attention, the proposed disentangled causal attention, and classfier-free guidance. T, I_1, I_2 represent text, spatial (image) condition and content tokens.

3.3 Training and Inference

For different generation tasks, we construct the token sequence z by interleaving text, condition, and content tokens according to task-specific formats. Specifically, spatial- and image-conditioned tasks follow the manner: $z = [t, v, x]$, while text-conditioned tasks use $z = [t, x]$. OmniGen-AR is trained to autoregressively predict the next token over these sequences using language modeling loss [57, 100].

During inference, the tokens are sequentially sampled based on the learned conditional probability: $\hat{z}_i = \text{argmax}\, \text{p}_\theta(z_i|z_{<i})$. After that, we feed them to the decoder of visual tokenizer to generate images. Classfier-Free Guidance (CFG) [26] is adopted to improve the generation quality, following previous work [66, 82].

4 Experiments

4.1 Experimental Setup

Training data. The training of OmniGen-AR includes three stages: 1) single image stage (SI), where we pretrain our model on large-scale image datasets, involving CC3M [62], CC12M [7], OpenImages [36], SAM1B [33], and Megalith-huggingface [46]. We also incorporate video datasets, *i.e.*, a 9M subset of Panda70M [11] and HD-VILA-100M [97], and randomly sample 1 frame for each video. 2) image-video joint stage (IV), where we maintain the datasets used in the first stage but sampled 9 frames from the videos. 3) multi-task stage (MT), where we train our model on a wide-range of high-quality datasets, including text-to-image datasets (JourneyDB [65], Synthetic-dataset-1M [53], and 10M internal data), image editing datasets (MagicBrush [105], Instruct-Pix2Pix [5], SEED-Edit [19]), depth-to-image datasets (MultiGen-Depth [54]), segmentation-to-image datasets (MultiGen-ADE20k [54] and MultiGen-COCOStuff [54]), and text-to-video datasets (OpenSora-pexels-45k [28], OpenVid-1M [50], and 0.5M high-quality internal data). We recaption all the images and videos using Qwen2-VL [83].

Implementation details. We adopt Qwen2.5 [98] as the text tokenier and transformer model. While for visual tokenizer, we use an image-video joint tokenizer, *i.e.*, Cosmos-DV8×16×16 [16], which allows us to tokenize different controls, images, and videos with the same codebook. During the SI and IV stages, we train our model on 512 resolution, and the learning rate is set to 1e-4. While for the MT stage, we increase the resolution to 1024 and decrease the learning rate to 2e-5. We train our model on 64 A100 GPUs, the global batch size is 256 for all stages, no warm up or learning rate decay are used. AdamW [45] is employed for optimization. During the IV and MT stages, we replace the standard causal attention mask with a disentangled causal attention mask with a probability of 10%, and similarly drop the text conditions for classifier-free guidance with the same probability. We set CFG scale to 6.0 during inference.

4.2 Comparison with State-of-the-arts

Text-to-image generation. In Table 2, we compare OmniGen-AR with existing image generation models on GenEval [20], a challenging and popular text-to-image (T2I) benchmark. The results

Table 2: Text-to-image generation on GenEval, Results markdd with † result are using prompt rewriting.

Method	Par.	Two.	Pos.	Color.	Overall
SDv1.5 [61]	0.9B	0.38	0.04	0.06	0.43
PixArt-alpha [9]	0.6B	0.50	0.08	0.07	0.48
SDv2.1 [61]	0.9B	0.51	0.07	0.17	0.50
LlamaGen [66]	0.8B	0.34	0.07	0.04	0.32
SimAR-SFT [82]	0.5B	**0.75**	0.20	0.24	0.53
Ours	0.5B	0.74	**0.20**	**0.29**	**0.55**
LDM [61]	1.4B	0.29	0.02	0.05	0.37
DALL-E 2 [59]	6.5B	0.66	0.10	0.19	0.52
Show-o [95]	1.3B	0.80	0.31	0.50	0.68
Infinity [23]	2B	0.85†	**0.49**†	**0.57**†	**0.73**†
Janus [92]	1.5B	0.68	0.46	0.42	0.61
Emu3 [86]	8.5B	0.81†	**0.49**†	0.45†	0.66†
SimAR-SFT [82]	1.5B	0.87	0.27	0.33	0.61
Ours	1.5B	**0.94**	0.30	0.40	0.63

Table 3: Video generation on VBench.

Method	Par.	Qua.	Sem.	Total
CogVideo [27]	9B	72.06	46.83	67.01
LaVie [87]	3B	78.78	70.31	77.08
OpSoraP V1.3 [42]	2.7B	80.14	65.62	77.23
CogVideoX [99]	5B	83.05	77.33	81.91
Hunyuan [35]	13B	85.09	75.82	83.24
Mira [30]	1.1B	78.78	44.21	71.87
TF-T2V [8]	1.8B	80.05	56.69	75.38
OpSora V1.2 [109]	1.1B	81.35	73.39	79.76
AniDiff V2 [22]	0.9B	82.90	69.75	80.27
VidCrafter-2.0 [8]	1.4B	82.20	73.42	80.44
CogVideoX [99]	2B	82.18	75.83	80.91
Wan2.1 [78]	1.3B	**85.23**	75.65	**83.31**
Ours	0.5B	76.60	67.20	74.72
Ours	1.5B	81.51	**78.08**	80.02

Table 4: Frame prediction on Kinetics-600 (left, * denotes zero-shot evaluation), image editing on Emu-Edit test set (middle), and spatial-conditioned generation (right). CT: CLIP text similarity between edited image and edited prompt, CI: CLIP image similarity between edited image and condition image.

Method	FVD (↓)
LVT [58]	225
ViTrans [91]	170
CogVideo [27]	109
ViVQVAE [77]	64
OmniTok [81]	**33**
VideoPoet-8B* [34]	687
Ours-1.5B*	429

Method	CT	CI
I-Pix2Pix [5]	0.22	0.83
MagBrush [105]	0.22	0.84
PnP [73]	0.09	0.52
Null-Text [47]	0.24	0.76
Emu-Edit [63]	0.23	**0.86**
OmniGen [94]	0.23	0.83
Ours-1.5B	**0.23**	0.84

Method	Mask mIoU (↑)	Depth RMSE (↓)
Uni-ControlNet [107]	19.39	40.65
GLIGEN [40]	23.78	38.83
EditAR [49]	22.62	34.93
ControlNet [106]	32.55	35.90
ControlAR [41]	39.95	**29.01**
OmniGen [94]	**40.06**	31.71
Ours-1.5B	35.28	37.42

show that OmniGen-AR significantly outperforms all other models with fewer than 1B parameters, including both diffusion models (*e.g.*, SDv2.1 [61]) and autoregressive models (*e.g.*, LLamaGen [66]). When scaled to 1.5B size, the overall performance of OmniGen-AR is improved from 0.57 to 0.63, highlighting its promising scalability when more training computes are available.

Text-to-video generation. We also evaluate OmniGen-AR on VBench [29] for text-to-video generation, and the comparison with existing video generation models is shown in Table 3. With only 0.5B parameters, our model achieves 74.72 total score on VBench, surpassing the previous SOTA AR-based models, *i.e.*, CogVideo [27], by 11% while using much fewer parameters (0.5B $v.s.$ 9B). Similar to what we have seen on T2I, the results of T2V could be signigicantly improved to 80.02 using 1.5B parameters, even beating diffusion models like OpenSora V1.2 [109]. It is also worth noting that it is the first time that a vanilla autoregressive model using discrete tokens could achieve 80+ score on VBench.

Frame prediction and image editing. To evaluate the image generation capability given visual context (image condition), we choose two types of tasks: frame prediction on Kinetics-600 [6] and image editing on Emu-Edit test set [63]. The results in Table 4 show that OmniGen-AR achieves much lower Fréchet Video Distance (FVD) than VideoPoet [34] for frame prediction. While for the more challenging image editing task, OmniGen-AR also achieves competitive results, *i.e.*, 0.23 CLIP text similarity [55].

Segmentation and depth-to-image generation. We follow previous work [41, 94] to report the segmentation-to-image and depth-to-image generation performance on ADE20K [110] and MultiGen-

Depth-Eval [54] in Table 4 (right). Compared to text or image condition, spatial conditions provide more structured and fine-grained instructions, thus posing a greater challenge to the model to geometrically accurate and contextually coherent content. We can see that OmniGen-AR achieves competitive results on both tasks, outperforming diffusion counterparts [48, 40].

4.3 Ablation Studies

Effects of disentangled causal attention. To better illustrate the potential information leakage in image-conditioned generation tasks, we compute the token match ratio (TMR) of MagicBrush [105], a popular image editing dataset, and visualize it in Figure 4. TMR is defined as the fraction of identical tokens in the same position between condition and content images: $\text{TMR} = \frac{1}{N_1}\sum_i \mathbf{1}\left[v_i = c_i\right]$, where v_i and c_i denote the i-th token from the condition and content images respectively, and N_1 is the total number of tokens. The x-axis of Figure 4 denotes binned TMR ranges (*e.g.*, 0.80–0.85), and the y-axis shows the proportion of samples falling into each bin. As can be seen, a significant portion of samples exhibit high TMR values, suggesting substantial overlap between condition and content images, which may imply unintended information leakage.

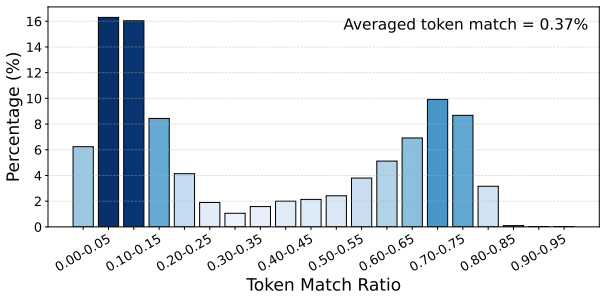


Figure 4: Token similarity on MagicBrush [105].

Table 5: OmniGen-AR w/ and w/o DCA on different generation tasks.

Method	VBen	Emu-CT	Mask
0%	70.33	0.15	24.76
5%	74.55	0.17	25.78
10%	74.72	0.20	25.33
20%	73.28	0.21	25.16
30%	71.69	0.19	21.49

As mentioned in Sec.4.1, we randomly replace standard causal attention with the proposed Disentangled Causal Attention (DCA) during training. We also conduct experiments with varying replacement probabilities using the 0.5B model across different tasks. As shown in Table5, adopting DCA with a 10% probability improves the CLIP text similarity on Emu from 0.15 to 0.20, indicating the encouragement robust conditioning without significantly limiting the access to informative context. Interestingly, DCA also yields slight gains in segmentation-to-image generation, which we hypothesize results from its ability to reduce over-reliance on exact segmentation inputs and thus improving the robustness of our model. Unless otherwise specified, we adopt a default replacement probability of 10% in all experiments.

Table 6: Joint or separate training.

Method	GEval	VBen	Emu	Mask
Joint	0.55	74.72	0.20	25.33
T2I	0.57	-	-	-
T2V	-	77.18	-	-
Edit	-	-	0.18	-
Seg	-	-	-	22.59

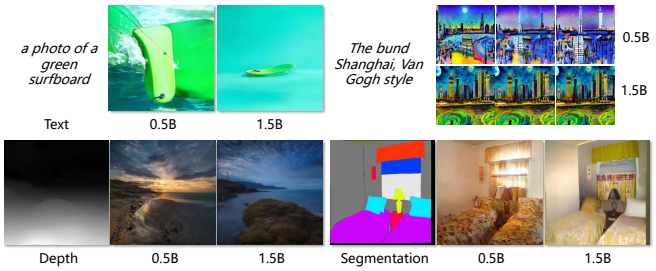


Figure 5: Effects of model scaling.

Synergy between different tasks. We study the synergy between different generation tasks by comparing joint training and separate training, both initialized with the 2nd stage checkpoint. As shown in Table 6, joint training leads to degraded performance on text-to-image and text-to-video benchmarks, possibly due to the lower visual quality of editing and spatial-conditioned datasets. In contrast, it improves the results on editing and segmentation-to-image generation tasks, suggesting that strengthening the foundation capability, *i.e.*, text-conditioned generation, can facilitate better generalization to a broader range of downstream tasks.

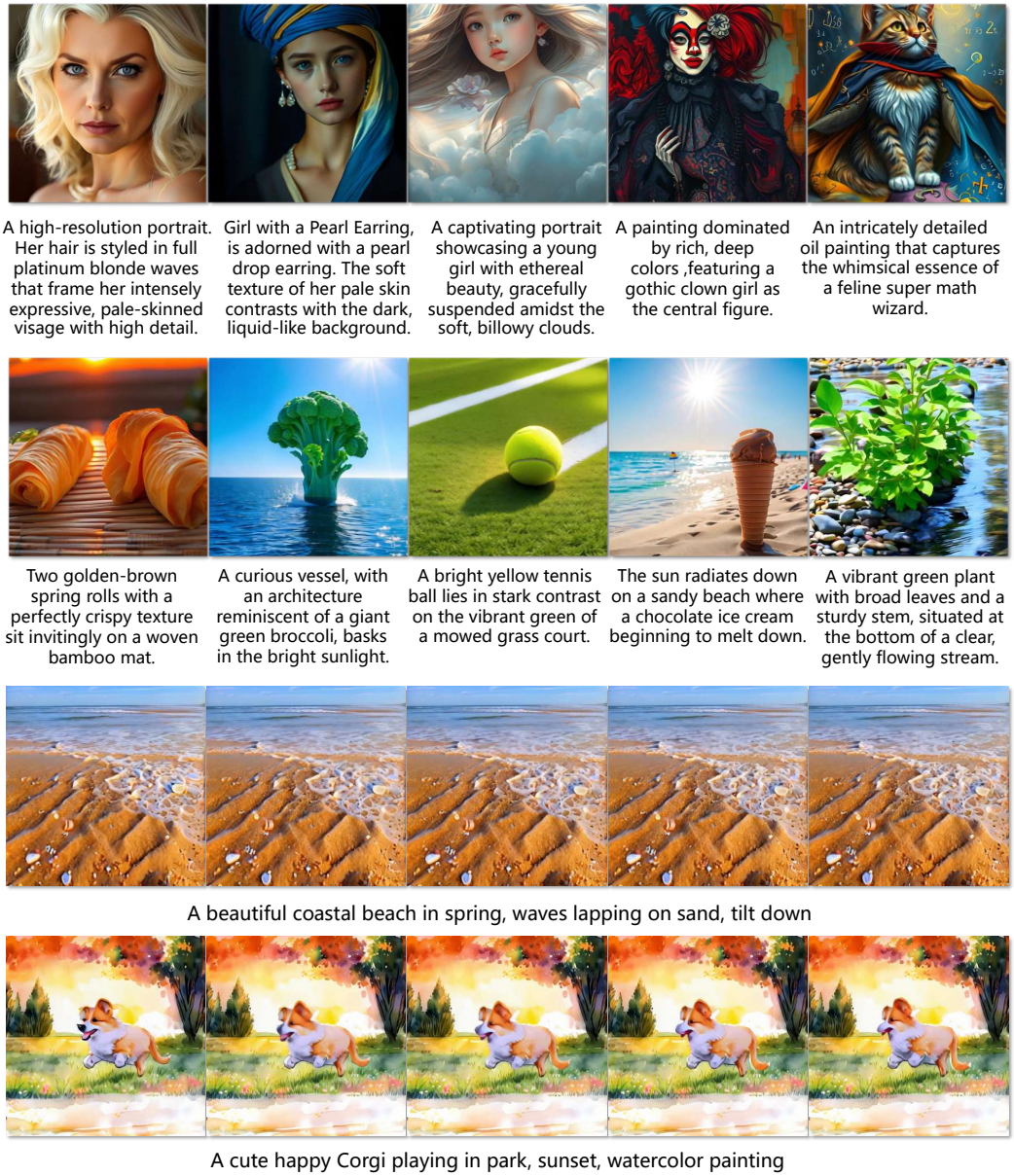


Figure 6: Visualization of text-to-image and text-to-video results generated by OmniGen-AR.

Model scaling. In Figure 5, we qualitatively compare the models with 0.5B and 1.5B parameters. It can be seen that scaling the model size could effectively improve the generation results on various tasks, leading to improved instruction-following capability and more aesthetically pleasing images.

4.4 Qualitative Results

We display some visualization results in Figure 6 and 7. OmniGen-AR could synthesize high-quality images based on various types of conditions, showcasing both versatility in handling diverse inputs and the ability to maintain semantic coherence with contexts. Several failure cases are also shown in Figure 8, which can be broadly categorized into two types: 1) Instruction-following capability. For instance, in the first row of Figure 8, the instruction is "Remove the bag on the bench next to the person sitting at the bus stop", but the model removes the person instead. This indicates a failure in grounding fine-grained spatial and referential cues from language into visual modifications. 2) Low-quality generations under sparse control signals. Examples in the second row (depth-to-image and segmentation-to-image) show blurry or structurally inconsistent results, which likely stem from noisy supervision and sparse training coverage for these conditions. These failure modes suggest two

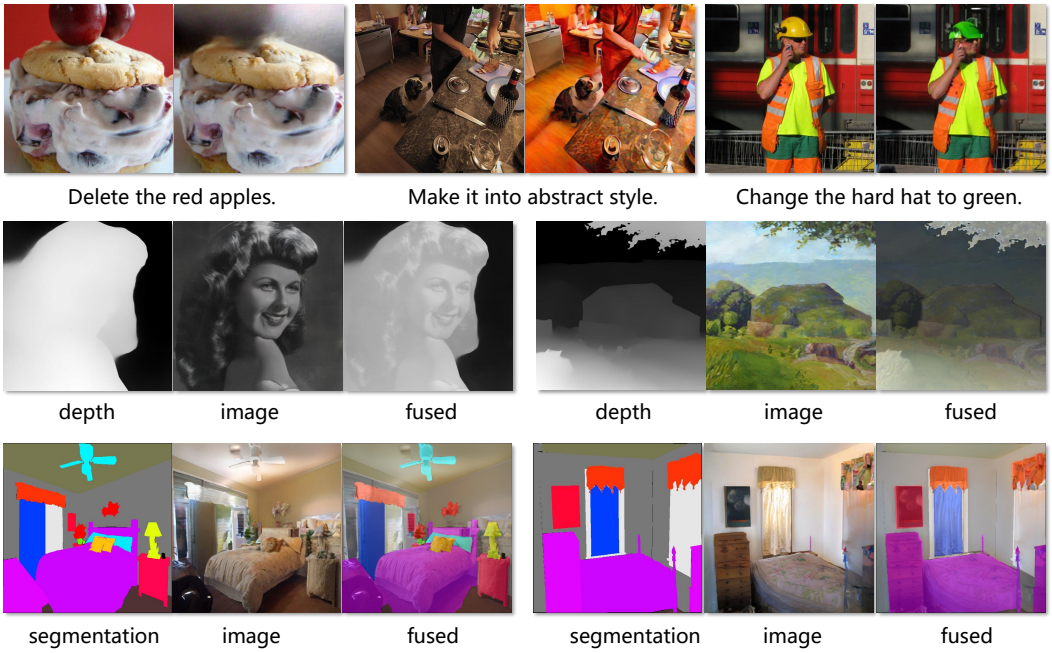


Figure 7: Image editing, depth-to-image, and segmentation-to-image generation results.

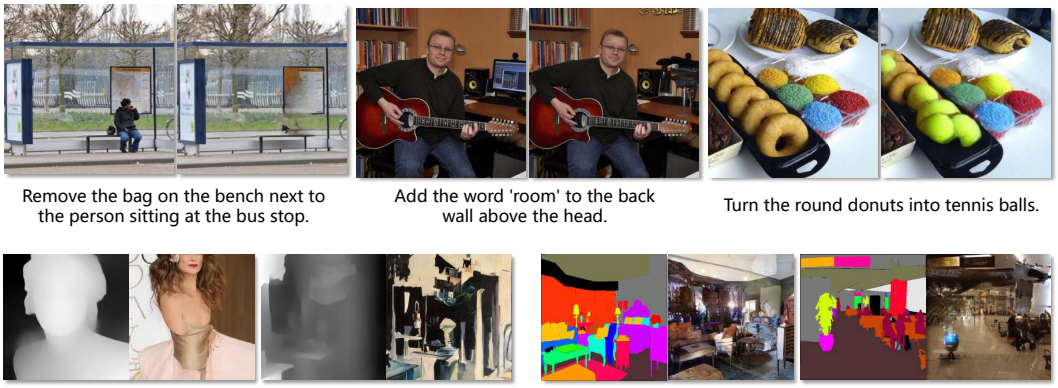


Figure 8: Failure cases generated by OmniGen-AR.

potential directions for future work: 1) Scaling up the model and training data to build a stronger base model with improved generalization and instruction-following ability across diverse visual tasks. 2) Leveraging chain-of-thought (CoT) [90] to improve the reasoning ability on complex prompts.

5 Conclusion and Broader Impacts

This paper presented OmniGen-AR, a unified autoregressive framework for any-to-image generation. OmniGen-AR represents a wide spectrum of conditional inputs, *i.e.*, text prompts, spatial controls, and visual contexts, as discrete tokens, and trains a unified autoregressive transformer to model the dependencies between these conditions and the target image tokens. To mitigate the potential information leakage in conditional generation tasks, we proposed Disentangled Causal Attention (DCA), which separates the attention pathways between condition and content tokens to facilitate the learning of instruction-following generation. Comprehensive experiments demonstrate that OmniGen-AR achieves state-of-the-art or at least competitive performance across a wide range of visual generation tasks.

Acknowledgement This project was supported by NSFC under Grant No. 62032006 and No. 624B2043.

References

[1] M. Abdin, J. Aneja, H. Behl, S. Bubeck, R. Eldan, S. Gunasekar, M. Harrison, R. J. Hewett, M. Javaheripi, P. Kauffmann, et al. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*, 2024.

[2] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[3] J. Bai, S. Bai, Y. Chu, Z. Cui, K. Dang, X. Deng, Y. Fan, W. Ge, Y. Han, F. Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

[4] D. Bolya, P.-Y. Huang, P. Sun, J. H. Cho, A. Madotto, C. Wei, T. Ma, J. Zhi, J. Rajasegaran, H. Rasheed, et al. Perception encoder: The best visual embeddings are not at the output of the network. *arXiv preprint arXiv:2504.13181*, 2025.

[5] T. Brooks, A. Holynski, and A. A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, 2023.

[6] J. Carreira, E. Noland, A. Banki-Horvath, C. Hillier, and A. Zisserman. A short note about kinetics-600. *arXiv preprint arXiv:1808.01340*, 2018.

[7] S. Changpinyo, P. Sharma, N. Ding, and R. Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *CVPR*, 2021.

[8] H. Chen, Y. Zhang, X. Cun, M. Xia, X. Wang, C. Weng, and Y. Shan. Videocrafter2: Overcoming data limitations for high-quality video diffusion models. In *CVPR*, 2024.

[9] J. Chen, J. Yu, C. Ge, L. Yao, E. Xie, Y. Wu, Z. Wang, J. Kwok, P. Luo, H. Lu, et al. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*, 2023.

[10] M. Chen, A. Radford, R. Child, J. Wu, H. Jun, D. Luan, and I. Sutskever. Generative pretraining from pixels. In *ICML*, 2020.

[11] T.-S. Chen, A. Siarohin, W. Menapace, E. Deyneka, H.-w. Chao, B. E. Jeon, Y. Fang, H.-Y. Lee, J. Ren, M.-H. Yang, et al. Panda-70m: Captioning 70m videos with multiple cross-modality teachers. In *CVPR*, 2024.

[12] Z. Chen, J. Wu, W. Wang, W. Su, G. Chen, S. Xing, M. Zhong, Q. Zhang, X. Zhu, L. Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *CVPR*, 2024.

[13] P. Dhariwal and A. Nichol. Diffusion models beat gans on image synthesis. In *NeurIPS*, 2021.

[14] P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *ICLR*, 2024.

[15] P. Esser, R. Rombach, and B. Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, 2021.

[16] N. et. al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

[17] Z. Fu, W. Lam, A. M.-C. So, and B. Shi. A theoretical analysis of the repetition problem in text generation. In *AAAI*, 2021.

[18] S. Ge, T. Hayes, H. Yang, X. Yin, G. Pang, D. Jacobs, J.-B. Huang, and D. Parikh. Long video generation with time-agnostic vqgan and time-sensitive transformer. In *ECCV*, 2022.

[19] Y. Ge, S. Zhao, C. Li, Y. Ge, and Y. Shan. Seed-data-edit technical report: A hybrid dataset for instructional image editing. *arXiv preprint arXiv:2405.04007*, 2024.

[20] D. Ghosh, H. Hajishirzi, and L. Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. In *NeurIPS*, 2024.

[21] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NeurIPS*, 2014.

[22] Y. Guo, C. Yang, A. Rao, Z. Liang, Y. Wang, Y. Qiao, M. Agrawala, D. Lin, and B. Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725*, 2023.

[23] J. Han, J. Liu, Y. Jiang, B. Yan, Y. Zhang, Z. Yuan, B. Peng, and X. Liu. Infinity: Scaling bitwise autoregressive modeling for high-resolution image synthesis. In *CVPR*, 2025.

[24] T. Henighan, J. Kaplan, M. Katz, M. Chen, C. Hesse, J. Jackson, H. Jun, T. B. Brown, P. Dhariwal, S. Gray, et al. Scaling laws for autoregressive generative modeling. *arXiv preprint arXiv:2010.14701*, 2020.

[25] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

[26] J. Ho and T. Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[27] W. Hong, M. Ding, W. Zheng, X. Liu, and J. Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv preprint arXiv:2205.15868*, 2022.

[28] hpcai tech. open-sora-pexels-45k. https://huggingface.co/datasets/hpcai-tech/open-sora-pexels-45k, 2025.

[29] Z. Huang, Y. He, J. Yu, F. Zhang, C. Si, Y. Jiang, Y. Zhang, T. Wu, Q. Jin, N. Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *CVPR*, 2024.

[30] X. Ju, Y. Gao, Z. Zhang, Z. Yuan, X. Wang, A. Zeng, Y. Xiong, Q. Xu, and Y. Shan. Miradata: A large-scale video dataset with long durations and structured captions. In *NeurIPS*, 2024.

[31] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[32] D. P. Kingma, M. Welling, et al. Auto-encoding variational bayes, 2013.

[33] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al. Segment anything. In *ICCV*, 2023.

[34] D. Kondratyuk, L. Yu, X. Gu, J. Lezama, J. Huang, G. Schindler, R. Hornung, V. Birodkar, J. Yan, M.-C. Chiu, et al. Videopoet: A large language model for zero-shot video generation. In *ICML*, 2024.

[35] W. Kong, Q. Tian, Z. Zhang, R. Min, Z. Dai, J. Zhou, J. Xiong, X. Li, B. Wu, J. Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

[36] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 2020.

[37] H. Li, T. Lan, Z. Fu, D. Cai, L. Liu, N. Collier, T. Watanabe, and Y. Su. Repetition in repetition out: Towards understanding neural text degeneration from the data perspective. In *NeurIPS*, 2023.

[38] J. Li, D. Li, C. Xiong, and S. Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022.

[39] T. Li, Y. Tian, H. Li, M. Deng, and K. He. Autoregressive image generation without vector quantization. In *NeurIPS*, 2024.

[40] Y. Li, H. Liu, Q. Wu, F. Mu, J. Yang, J. Gao, C. Li, and Y. J. Lee. Gligen: Open-set grounded text-to-image generation. In *CVPR*, 2023.

[41] Z. Li, T. Cheng, S. Chen, P. Sun, H. Shen, L. Ran, X. Chen, W. Liu, and X. Wang. Controlar: Controllable image generation with autoregressive models. In *ICLR*, 2025.

[42] B. Lin, Y. Ge, X. Cheng, Z. Li, B. Zhu, S. Wang, X. He, Y. Ye, S. Yuan, L. Chen, et al. Open-sora plan: Open-source large video generation model. *arXiv preprint arXiv:2412.00131*, 2024.

[43] H. Liu, C. Li, Y. Li, and Y. J. Lee. Improved baselines with visual instruction tuning. In *CVPR*, 2024.

[44] H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning. In *NeuIPS*, 2023.

[45] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[46] madebyollin. Megalith-huggingface. https://huggingface.co/datasets/madebyollin/megalith-10m, 2024.

[47] R. Mokady, A. Hertz, K. Aberman, Y. Pritch, and D. Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *CVPR*, 2023.

[48] C. Mou, X. Wang, L. Xie, Y. Wu, J. Zhang, Z. Qi, and Y. Shan. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In *AAAI*, 2024.

[49] J. Mu, N. Vasconcelos, and X. Wang. Editar: Unified conditional generation with autoregressive models. In *CVPR*, 2025.

[50] K. Nan, R. Xie, P. Zhou, T. Fan, Z. Yang, Z. Chen, X. Li, J. Yang, and Y. Tai. Openvid-1m: A large-scale high-quality dataset for text-to-video generation. *arXiv preprint arXiv:2407.02371*, 2024.

[51] W. Peebles and S. Xie. Scalable diffusion models with transformers. In *CVPR*, 2023.

[52] D. Podell, Z. English, K. Lacey, A. Blattmann, T. Dockhorn, J. Müller, J. Penna, and R. Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

[53] ProGamerGov. synthetic-dataset-1m-dalle3-high-quality-captions. https://huggingface.co/datasets/ProGamerGov/synthetic-dataset-1m-dalle3-high-quality-captions, 2022.

[54] C. Qin, S. Zhang, N. Yu, Y. Feng, X. Yang, Y. Zhou, H. Wang, J. C. Niebles, C. Xiong, S. Savarese, et al. Unicontrol: A unified diffusion model for controllable visual generation in the wild. *arXiv preprint arXiv:2305.11147*, 2023.

[55] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[56] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al. Improving language understanding by generative pre-training. *OpenAI Blog*, 2018.

[57] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI Blog*, 2019.

[58] R. Rakhimov, D. Volkhonskiy, A. Artemov, D. Zorin, and E. Burnaev. Latent video transformer. *arXiv preprint arXiv:2006.10704*, 2020.

[59] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[60] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever. Zero-shot text-to-image generation. In *ICML*, 2021.

[61] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

[62] P. Sharma, N. Ding, S. Goodman, and R. Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018.

[63] S. Sheynin, A. Polyak, U. Singer, Y. Kirstain, A. Zohar, O. Ashual, D. Parikh, and Y. Taigman. Emu edit: Precise image editing via recognition and generation tasks. In *CVPR*, 2024.

[64] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. In *ICLR*, 2021.

[65] K. Sun, J. Pan, Y. Ge, H. Li, H. Duan, X. Wu, R. Zhang, A. Zhou, Z. Qin, Y. Wang, et al. Journeydb: A benchmark for generative image understanding. In *NeurIPS*, 2023.

[66] P. Sun, Y. Jiang, S. Chen, S. Zhang, B. Peng, P. Luo, and Z. Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024.

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

[68] G. Team, R. Anil, S. Borgeaud, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, K. Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[69] K. Tian, Y. Jiang, Z. Yuan, B. Peng, and L. Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *arXiv preprint arXiv:2404.02905*, 2024.

[70] R. Tian, Q. Dai, J. Bao, K. Qiu, Y. Yang, C. Luo, Z. Wu, and Y.-G. Jiang. Reducio! generating 1k video within 16 seconds using extremely compressed motion latents. In *ICCV*, 2025.

[71] R. Tian, M. Gao, M. Xu, J. Hu, J. Lu, Z. Wu, Y. Yang, and A. Dehghan. Unigen: Enhanced training & test-time strategies for unified multimodal understanding and generation. In *NeurIPS*, 2025.

[72] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[73] N. Tumanyan, M. Geyer, S. Bagon, and T. Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *CVPR*, 2023.

[74] A. van den Oord and N. Kalchbrenner. Pixel rnn. In *ICML*, 2016.

[75] A. Van Den Oord, O. Vinyals, et al. Neural discrete representation learning. In *NeurIPS*, 2017.

[76] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[77] J. Walker, A. Razavi, and A. v. d. Oord. Predicting video with vqvae. *arXiv preprint arXiv:2103.01950*, 2021.

[78] A. Wang, B. Ai, B. Wen, C. Mao, C.-W. Xie, D. Chen, F. Yu, H. Zhao, J. Yang, J. Zeng, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

[79] H. Wang, S. Suri, Y. Ren, H. Chen, and A. Shrivastava. Larp: Tokenizing videos with a learned autoregressive generative prior. In *ICLR*, 2025.

[80] J. Wang, D. Chen, Z. Wu, C. Luo, L. Zhou, Y. Zhao, Y. Xie, C. Liu, Y.-G. Jiang, and L. Yuan. Omnivl: One foundation model for image-language and video-language tasks. In *NeurIPS*, 2022.

[81] J. Wang, Y. Jiang, Z. Yuan, B. Peng, Z. Wu, and Y.-G. Jiang. Omnitokenizer: A joint image-video tokenizer for visual generation. In *NeurIPS*, 2024.

[82] J. Wang, Z. Tian, X. Wang, X. Zhang, W. Huang, Z. Wu, and Y.-G. Jiang. Simplear: Pushing the frontier of autoregressive visual generation through pretraining, sft, and rl. *arXiv preprint arXiv:2504.11455*, 2025.

[83] P. Wang, S. Bai, S. Tan, S. Wang, Z. Fan, J. Bai, K. Chen, X. Liu, J. Wang, W. Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[84] W. Wang, Z. Li, D. Lian, C. Ma, L. Song, and Y. Wei. Mitigating the language mismatch and repetition issues in llm-based machine translation via model editing. *arXiv preprint arXiv:2410.07054*, 2024.

[85] X. Wang, T. Darrell, S. S. Rambhatla, R. Girdhar, and I. Misra. Instancediffusion: Instance-level control for image generation. In *CVPR*, 2024.

[86] X. Wang, X. Zhang, Z. Luo, Q. Sun, Y. Cui, J. Wang, F. Zhang, Y. Wang, Z. Li, Q. Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024.

[87] Y. Wang, X. Chen, X. Ma, S. Zhou, Z. Huang, Y. Wang, C. Yang, Y. He, J. Yu, P. Yang, et al. Lavie: High-quality video generation with cascaded latent diffusion models. *IJCV*, 2025.

[88] Y. Wang, K. Li, Y. Li, Y. He, B. Huang, Z. Zhao, H. Zhang, J. Xu, Y. Liu, Z. Wang, et al. Internvideo: General video foundation models via generative and discriminative learning. *arXiv preprint arXiv:2212.03191*, 2022.

[89] Y. Wang, H. Xu, X. Zhang, Z. Chen, Z. Sha, Z. Wang, and Z. Tu. Omnicontrolnet: Dual-stage integration for conditional image generation. In *CVPR*, 2024.

[90] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, 2022.

[91] D. Weissenborn, O. Täckström, and J. Uszkoreit. Scaling autoregressive video models. *arXiv preprint arXiv:1906.02634*, 2019.

[92] C. Wu, X. Chen, Z. Wu, Y. Ma, X. Liu, Z. Pan, W. Liu, Z. Xie, X. Yu, C. Ruan, et al. Janus: Decoupling visual encoding for unified multimodal understanding and generation. In *CVPR*, 2024.

[93] Y. Wu, Z. Zhang, J. Chen, H. Tang, D. Li, Y. Fang, L. Zhu, E. Xie, H. Yin, L. Yi, et al. Vila-u: a unified foundation model integrating visual understanding and generation. In *ICLR*, 2025.

[94] S. Xiao, Y. Wang, J. Zhou, H. Yuan, X. Xing, R. Yan, C. Li, S. Wang, T. Huang, and Z. Liu. Omnigen: Unified image generation. *arXiv preprint arXiv:2409.11340*, 2024.

[95] J. Xie, W. Mao, Z. Bai, D. J. Zhang, W. Wang, K. Q. Lin, Y. Gu, Z. Chen, Z. Yang, and M. Z. Shou. Show-o: One single transformer to unify multimodal understanding and generation. In *ICLR*, 2025.

[96] J. Xu, X. Liu, J. Yan, D. Cai, H. Li, and J. Li. Learning to break the loop: Analyzing and mitigating repetitions for neural text generation. In *NeurIPS*, 2022.

[97] H. Xue, T. Hang, Y. Zeng, Y. Sun, B. Liu, H. Yang, J. Fu, and B. Guo. Advancing high-resolution video-language representation with large-scale video transcriptions. In *CVPR*, 2022.

[98] A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[99] Z. Yang, J. Teng, W. Zheng, M. Ding, S. Huang, J. Xu, Y. Yang, W. Hong, X. Zhang, G. Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024.

[100] J. Yu, Y. Xu, J. Y. Koh, T. Luong, G. Baid, Z. Wang, V. Vasudevan, A. Ku, Y. Yang, B. K. Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. In *ICLR*, 2024.

[101] Q. Yu, M. Weber, X. Deng, X. Shen, D. Cremers, and L.-C. Chen. An image is worth 32 tokens for reconstruction and generation. In *NeurIPS*, 2024.

[102] X. Zhai, A. Kolesnikov, N. Houlsby, and L. Beyer. Scaling vision transformers. In *CVPR*, 2022.

[103] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023.

[104] H. Zhang, D. Hong, Y. Wang, J. Shao, X. Wu, Z. Wu, and Y.-G. Jiang. Creatilayout: Siamese multimodal diffusion transformer for creative layout-to-image generation. In *ICCV*, 2025.

[105] K. Zhang, L. Mo, W. Chen, H. Sun, and Y. Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. In *NeurIPS*, 2023.

[106] L. Zhang, A. Rao, and M. Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023.

[107] S. Zhao, D. Chen, Y.-C. Chen, J. Bao, S. Hao, L. Yuan, and K.-Y. K. Wong. Uni-controlnet: All-in-one control to text-to-image diffusion models. In *NeurIPS*, 2023.

[108] P. Zheng, J. Wang, Y. Chang, Y. Yu, R. Ma, and Z. Wu. Rethinking discrete tokens: Treating them as conditions for continuous autoregressive image synthesis. In *ICCV*, 2025.

[109] Z. Zheng, X. Peng, T. Yang, C. Shen, S. Li, H. Liu, Y. Zhou, T. Li, and Y. You. Open-sora: Democratizing efficient video production for all. *arXiv preprint arXiv:2412.20404*, 2024.

[110] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017.

[111] C. Zhou, L. Yu, A. Babu, K. Tirumala, M. Yasunaga, L. Shamis, J. Kahn, X. Ma, L. Zettlemoyer, and O. Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model. In *ICLR*, 2025.

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: We clearly and accurately state the contribution and scope in the abstract and introduction (Sec. 1).

   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: We discuss the limitations by showing failure cases in the experimental section (Sec. 4.4) and conclusion section (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: The 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: We specify the model configurations, hyperparameters, and data in Sec. 4, supporting the reproduction of main experimental results.

   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: We will release the full codebase upon publication, including implementation details and training scripts necessary to reproduce the main experimental results. The training data includes a combination of public datasets and internal data that cannot be released due to licensing constraints. We will provide clear documentation specifying how the datasets were used and how public data can be substituted to approximate our results.

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: We specify the model configurations, hyperparameters, and data in Sec. 4.

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: Error bars would be too computationally expensive to report. We claim that gains in our experimental results are consistent and significant.

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 specifies details about computer resources in Sec. 4.

   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: We make sure to conduct the research strictly following 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: We discuss the broader impacts in Sec. 5.

    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: [No]

    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 cite all third-party assets (e.g., code, datasets, pretrained models) used in the paper. While license terms are not explicitly listed in the current draft, we will ensure that all licenses and terms of use are properly documented and respected in the final version upon publication.

    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: [No]

    Justification: The paper does not release new assets.

    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: [NA]

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

    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.