

Generative Refinement Networks for Visual Synthesis

Jian Han, Jinlai Liu, Jiahuan Wang, Bingyue Peng, and Zehuan Yuan*

ByteDance, Beijing, China

{hanjian.thu123, liujinlai.licio, wangjiahuan.123}@bytedance.com

{bingyue.peng, yuanzehuan}@bytedance.com

Code and models: <https://github.com/bytedance/GRN>



Fig. 1: Qualitative results for the class-to-image generation task.

Abstract. While diffusion models dominate the field of visual generation, they remain computationally inefficient, as they allocate uniform computational effort to samples with varying levels of complexity. In contrast, autoregressive (AR) models are inherently complexity-aware, as evidenced by their variable likelihoods, but are often hindered by lossy discrete tokenization and error accumulation. In this work, we introduce Generative Refinement Networks (GRN), a next-generation visual synthesis paradigm to address these issues. At its core, GRN addresses the discrete tokenization bottleneck through a theoretically near-lossless Hierarchical Binary Quantization (HBQ), achieving a reconstruction quality comparable to continuous counterparts. Built upon HBQ’s latent space, GRN fundamentally upgrades AR generation with a global refinement mechanism that progressively perfects and corrects artworks — like a human artist painting. Besides, GRN integrates an entropy-guided sampling strategy, enabling complexity-aware, adaptive-step generation without compromising visual quality. On the ImageNet benchmark, GRN establishes new records in image reconstruction (0.56 rFID) and class-conditional image generation (1.81 gFID). We also scale GRN to more challenging text-to-image and text-to-video generation, delivering superior performance on an equivalent scale. We release all models and code to foster further research on GRN.

Keywords: Visual Generation · Tokenizer · Autoregressive Model

* Corresponding author

1 Introduction

The field of visual generation has advanced rapidly, driven primarily by scaling diffusion transformers [4, 22, 29, 43, 66]. By progressively integrating trajectories along a learned velocity field that transports simple noise prior to the empirical data distribution, these models demonstrate strong capabilities in synthesizing high-quality visual content. However, this continuous flow paradigm inherently lacks adaptive-step capacity. Optimized via mean squared error (MSE) without explicit likelihoods, these models are restricted to a fixed number of steps, rigidly allocating identical computational resources to all samples regardless of varying levels of complexity.

Meanwhile, inspired by the success of token-level likelihood estimation in large language models [1, 41], autoregressive (AR) models have also garnered extensive research interest in visual synthesis [26, 52, 56, 58]. Nevertheless, current AR approaches are bottlenecked by two critical shortcomings. First, they intrinsically suffer from inferior reconstruction quality when utilizing discrete tokens as opposed to continuous representations. Second, their strictly causal prediction mechanism, whether they operate token-by-token or scale-by-scale, inevitably causes severe error accumulation over multi-step generation. This exposes a critical lack of error-correction capability, as the model cannot retroactively refine previous mistakes. Furthermore, even with parallel prediction in masked AR models [7, 17], high-confidence tokens become immutable and cannot be revised later. They still inherently lack a holistic refinement mechanism.

These observations motivate us to propose a simple yet intuitive refinement-based AR generation framework equipped with adaptive computation capabilities. Specifically, we introduce Generative Refinement Networks (GRN), a next-generation visual synthesis paradigm designed to overcome the rigidly fixed computational costs of diffusion models and the inherent shortcomings of standard autoregressive models. Specifically, to address the inferior reconstruction of discrete tokens, we first propose Hierarchical Binary Quantization (HBQ). By ensuring an exponential decay of reconstruction error without increasing latent channels, HBQ empowers discrete image and video tokenizers to achieve near-lossless reconstruction, matching the performance of continuous tokenizers at a higher compression rate. Building upon these robust representations, GRN executes a complexity-aware adaptive-step generation process by employing an entropy-guided sampling mechanism. It dynamically distributes computational loads based on the varying difficulty of visual content, while employing a global refinement mechanism to retroactively mitigate accumulated errors.

Extensive experiments on diverse visual tasks validate the superiority of our framework. On the ImageNet 256×256 benchmark [16] for class-conditional image synthesis, GRN sets a new record for both image reconstruction and generation quality. Furthermore, demonstrating exceptional task generalization and scalability, we also successfully scale GRN to high-resolution text-to-image (T2I) and text-to-video (T2V) scenarios. When scaled up, GRN demonstrates the capability to generate photorealistic 1024×1024 images alongside dynamic, high-

fidelity 480p videos ranging from 2 to 10 seconds. In summary, our main contributions are as follows:

1. We propose GRN, the next-generation visual synthesis framework. It is characterized by a global refinement mechanism and complexity-aware generation, achieving robust and efficient visual generation.
2. We introduce Hierarchical Binary Quantization and contribute a series of discrete image/video tokenizers. For the first time, discrete visual tokenizers are on par with continuous ones with the same latent dimensions.
3. Extensive experiments show that GRN achieves state-of-the-art results on standard C2I benchmarks, with an rFID of 0.56 and a gFID of 1.81. When scaled to more challenging T2I and T2V tasks, it demonstrates superior performance compared to methods at an equivalent scale.

2 Related Work

2.1 Visual Tokenizer

Visual tokenizers [19, 47, 53] compress visual content for efficient generation. Early vector quantization methods [19, 53] map continuous features to a discrete codebook, but suffer from limited scalability, prompting lookup-free approaches [39, 67] to enable larger vocabularies. Despite this, a performance gap to continuous representations remains. Recent works [2, 26] aim to close this gap by drastically scaling vocabularies, outperforming continuous VAEs. However, this gain comes at the cost of slower convergence and larger generative models, motivating more efficient quantization schemes.

2.2 Autoregressive Models

Inspired by large language models, [19, 28, 50, 55] explore visual generation via next-token prediction. MaskGIT [7] accelerates generation using parallel decoding, where it first generates high-confidence tokens and then iteratively fills in the remainder. VAR [52] shifts autoregression to next-scale prediction, improving quality and achieving over $10\times$ faster inference. Nevertheless, AR models remain limited by lossy discrete tokenization and error accumulation, and still lag behind diffusion methods. Although Infinity [26] introduces self-correction by randomly flipping bitwise tokens, its assumption of less than 30% diffuse errors covers limited patterns.

2.3 Adaptive-step Generation

Diffusion models dominate visual generation [4, 18, 31, 44, 54], but typically require tens of inference steps. Distillation methods [60, 61] reduce sampling steps substantially, yet still rely on predefined schedules with fixed steps. This “one-size-fits-all” strategy wastes computational resources on simple prompts. Recently, AdaDiff [65] employs an external network to determine instance-specific steps and uses a policy gradient method to maximize the reward. The sophisticated pipeline requires an additional network and reward signals.

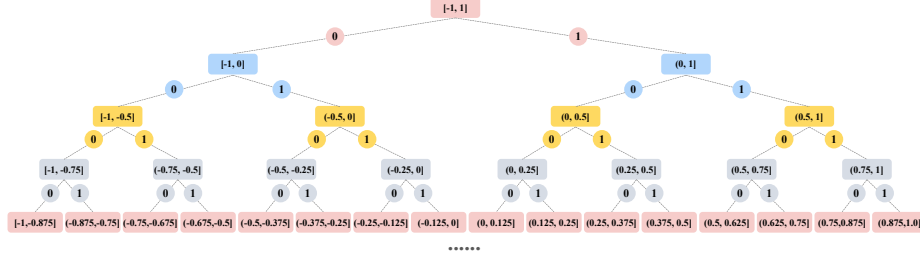


Fig. 2: Hierarchical Binary Quantization. Each element from the VAE encoded features undergoes several rounds of hierarchical binary quantization. The quantization error decays exponentially with the number of rounds, theoretically enabling lossless quantization to be achieved rapidly.

3 Method

3.1 Visual Tokenizer

The visual tokenizer plays a vital role in learning a compact latent space to compress high-dimensional realistic data. We adopt the 3D causal VAE design proposed in Wan 2.1 [54] so that images and videos can be tokenized in a unified framework. Specifically, given an image or a video $X \in R^{(1+4T) \times H \times W \times 3}$, the tokenizer encodes its spatio-temporal information into dimensions $[1+T, H/16, W/16]$ while expanding the number of channels to C . Since our goal is lossless discrete compression, and VAE features are continuous signals, we frame feature quantization as a signal transformation problem. Inspired by Harr Wavelet [24] in signal processing, we introduce Hierarchical Binary Quantization to transform VAE features into discrete ones.

Hierarchical Binary Quantization. We first append a $\tanh(\cdot)$ non-linear activation function after the VAE encoder to map the feature representation F from an unbounded range to the closed interval $(-1, +1)$. As illustrated in Fig. 2, each element in F undergoes several rounds of binary quantization based on a binary tree of buckets with center c as defined in Eq. 1 and Eq. 2.

$$c_i = \sum_{j=1}^{i-1} \frac{\delta[q_j]}{2^j} \quad (1)$$

$$q_i = \begin{cases} 0 & \text{if } F \leq c_i, \\ 1 & \text{if } F > c_i. \end{cases} \quad (2)$$

where $\delta(\cdot)$ is a delta function with -1 when $q_i = 0$ and 1 otherwise. Then we obtain the quantized binary labels $\{q_1, q_2, \dots, q_M\}$, where $q_j \in \{0, 1\}^{[1+T, H/16, W/16, C]}$. Here M is the total round of hierarchical binary quantization. In this way, we perform quantization from coarse to fine to represent information of different frequencies, and the quantization error e_j for the round j is less than $\frac{1}{2^j}$. The



Fig. 3: An example of Hierarchical Binary Quantization ($M=4$). For q_1 , q_2 , and q_3 , we truncate the complete sequence and take the truncated parts for reconstruction.

upper bound of the quantization error decays exponentially with the number of rounds, theoretically enabling lossless quantization to be achieved rapidly. Fig. 3 shows the images reconstructed from the quantized intermediate results, revealing the coarse-to-fine property.

$$\hat{F} = \delta[q_1] \cdot 2^{-1} + \delta[q_2] \cdot 2^{-2} + \dots + \delta[q_M] \cdot 2^{-M} \quad (3)$$

Subsequently, the quantized feature $\hat{F}$ can be derived according to Eq. 3. The detailed algorithm for HBQ is provided in Appendix 1. During the training phase of the visual tokenizer, the quantized feature $\hat{F}$ is taken as input to the decoder to reconstruct the raw image or video X . Following the common practice for training discrete visual tokenizers, we adopt the Straight-Through Estimator (STE) to backpropagate gradients to the encoder. The training loss is a weighted combination of the reconstruction loss (λ_{recons}), LPIPS perceptual loss (λ_{LPIPS}), and the GAN loss (λ_{GAN}) from a PatchGAN discriminator.

After tokenization, we obtain binary outputs with the size $[1+T, \frac{H}{16}, \frac{W}{16}, C, M]$. However, it is nearly impossible to encode outputs into INT scalars to perform generation by merging C and M dimensions due to its equivalence to introducing a codebook with the large size 2^{CM} . Inspired by bitwise tokens [26], we propose two variants of GRN, *i.e.* GRN_{ind} and GRN_{bit} to support generation. For GRN_{ind} we simply encode the M dimension to INT scalars, resulting in $Y_{ind} \in \{0, \dots, 2^M - 1\}^{[1+T, H/16, W/16, C]}$. For GRN_{bit} we concatenate the last two dimensions and predict $Y_{bit} \in \{0, 1\}^{[1+T, H/16, W/16, CM]}$. For both variants, we flatten the spatiotemporal dimensions and predict the entire channel dimension in parallel for each token via multi-token prediction [34].

3.2 Generative Refinement Network

Inspired by the intuition of human drawing, we propose an elegantly simple autoregressive refinement framework for visual generation, which commences with a random token map. Let F_t represent the state of the token map in step t . The objective is to predict the drawing map Y_{t+1} , based on the current state F_t . To explicitly formulate this process, we define F_t as a composition of three components: a random map Y_{rand} , a drawing map Y_t , and a binary selection map S_t . The relationship is formally expressed in Eq. 4, where F_t is constructed by selecting from Y_t or Y_{rand} based on the values in S_t .

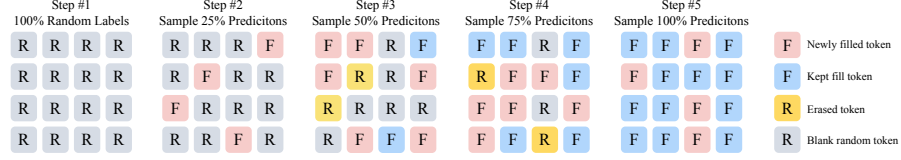


Fig. 4: Generative Refinement Framework. Starting from a random token map, GRN randomly selects more predictions at each step and refines all input tokens. For example, compared to the second step, the third step filled six new tokens (pink), kept two tokens (blue), erased two tokens (yellow), and left six tokens blank (gray).

$$F_t = S_t \cdot Y_t \oplus \overline{S_t} \cdot Y_{rand} \quad (4)$$

Intuitively, $S_t \cdot Y_t$ represents the current drawing while $\overline{S_t} \cdot Y_{rand}$ corresponds to the blank area without any information to mimic the intermediate step during the human drawing procedure. S is designed to make the accumulation statistic l , namely the proportion of ones in S , increase monotonically from 0% to 100% during the refinement steps. Therefore, F gradually converges to ideal token maps. In order to obtain Y_{t+1} , we employ a transformer $\Phi(\cdot)$ and approximate $p(Y_{t+1})$ as

$$p(Y_{t+1}) = \Phi(F_t, cond) \quad (5)$$

to model next-step drawing by fitting real token maps, where *cond* denotes the generation condition, such as class embeddings or texts. S_{t+1} is constructed following Eq. 6 based on Y_{t+1} . An alternative approach based on prediction confidence was also investigated for constructing S_{t+1} . However, it produced inferior results, as detailed in Appendix 5.3.

$$S_t = RandLike(Y_t) < l_t \quad (6)$$

We introduce a complexity-aware sampling strategy to control l_t to not only maintain its monotonicity, but also consider the uncertainty of Y . We will discuss the details below. Based on the prediction Y_{t+1} and selection map S_{t+1} , the state F_{t+1} is updated accordingly as mentioned in Eq. 4. Thus, a coherent loop of progressive generation and refinement is formed. The autoregressive mechanism allows us not only to improve more and more tokens with high certainty, but also to erase obvious errors with more context included as the drawing proceeds. Ideally, the process converges to the best result as more and more information accumulates. The toy process is demonstrated in Fig. 4.

Training. For each iteration in the training stage, we randomly sample random tokens Y_{rand} from a uniform distribution $\{0, 1, \dots, 2^M - 1\}$ for GRN_{ind} and $\{0, 1\}$ for GRN_{bit} . The binary map S_t is also uniformly sampled with varying selection ratios that control how many real tokens are used as input. Therefore, the input F_t to the transformer consists of $N \cdot l_t$ tokens sampled from ground-truth tokens Y_{gt} and $N \cdot (1 - l_t)$ tokens sampled from random tokens Y_{rand} . Note that token sampling is randomly conducted along the spatial, temporal, and channel dimensions, with no additional priors (Eq. 6). N refers to the total number of tokens, which equals $(1 + T) \cdot H/16 \cdot W/16 \cdot (C \text{ or } CM)$. Taking F_t constructed

by Eq. 4 as input with partial information, our goal is to predict ground-truth similar to x-prediction in diffusion setting using the simple Cross-Entropy loss as illustrated in Eq. 7. Here y_i denotes the ground truth token.

$$\mathcal{L} = -\mathbf{E}[\frac{1}{N} \sum_{i=0}^N \log p(y_i | F_t, \text{cond})] \quad (7)$$

The detailed training and inference process for GRN is provided in Appendix 2. Additional comparisons between GRN and other autoregressive models are provided in Appendix 3.

Complexity-Aware Sampling. We propose an entropy-guided scheduling function to determine l_t , where t refers to the index of the refinement step. In particular, we calculate the average entropy $H(Y_t)$ for step t as

$$H(Y_t) = \frac{1}{N} \cdot \frac{1}{\log_2 K} \cdot \sum_{i=0}^N \sum_{j=0}^K -p(y_{(i,j)} | F_{t-1}, \text{cond}) \cdot \log_2 p(y_{(i,j)} | F_{t-1}, \text{cond}). \quad (8)$$

In Eq. 8, we denote i as the token index and $j \in \{1, 2, \dots, K\}$ as the category index, where K is the total number of categories. Note that $K = 2^M$ for GRN_{ind} and $K = 2$ for GRN_{bit} . Generation complexity is measured by the entropy $H(Y_t)$, bounded between 0 and 1. Given that a smaller $H(Y_t)$ denotes greater predictive confidence, we allocate fewer refinement steps alongside a steeper increase in l_t , thereby retaining more information from Y_t . Conversely, when high entropy indicates substantial complexity, we apply more refinement steps and a moderate progression of l_t . Specifically, we formulate l_t as

$$l_t = l(Y, t) = \frac{t}{\alpha} \mathbb{1}_{t \leq t_0} + \left(\frac{t_0}{\alpha} + \frac{\alpha - t_0}{\alpha} \cdot \frac{(t - t_0)}{k \cdot H(Y_{(t_0+1)}) + b} \right) \mathbb{1}_{t > t_0}. \quad (9)$$

Here, $H(Y_{(t_0+1)})$ represents the average entropy calculated from a specific step. We set a warm-up period with $t_0 = 5$ and $\alpha = 50$, as we observed that the entropy values are unstable during the initial steps. The hyperparameters k control the dynamic range of adaptive steps, and b is the bias. We also clip the value of $k \cdot H(Y_{(t_0+1)}) + b$ to ensure the total number of inference steps remains within the range of $[T_{min}, T_{max}]$.

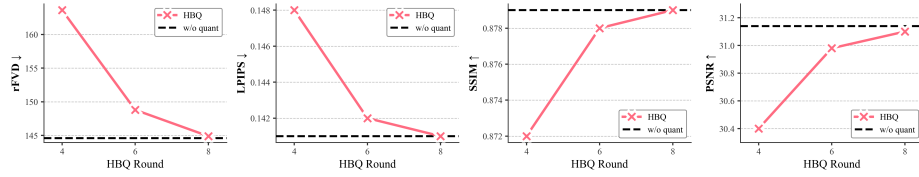


Fig. 5: Effect of HBQ rounds. 8-round HBQ matches the continuous baseline.

Table 1: Reconstruction performance comparison of image tokenizers on ImageNet (256×256).

Method	Tokenizer Type	Spatial Ratio	Latent Channel	Channel Bits	Compress Ratio	rFID↓	LPIPS↓	SSIM↑	PSNR↑
SD-VAE [47]	Continuous	16	16	16	24	0.87	-	0.68	24.08
RAE [68]	Continuous	16	768	16	0.5	0.62	0.25	0.44	19.20
VAR [†] [52]	Discrete	16	N/A	N/A	193	0.85	0.15	0.64	22.47
LlamaGen [50]	Discrete	16	1	14	439	2.19	-	0.68	20.79
Open-MAGVIT2 [36]	Discrete	16	1	18	341	1.17	-	-	22.64
HBQ (M=4)	Discrete	16	16	4	96	0.56	0.13	0.71	23.01

Table 2: Reconstruction performance of HBQ tokenizers. To evaluate reconstruction quality, we curated a challenging validation set of 160 high-motion videos. For continuous tokenizers, we consider each latent channel to hold 16 bits of information. An HBQ tokenizer with M rounds introduces M bits within each latent channel. We train HBQ*(M=4) for more iterations and apply an optimized GAN loss weight.

Method	Tokenizer Type	Latent Channel	Spatial Stride	Temporal Stride	Channel Bits	Compress Ratio	rFVD↓	LPIPS↓	SSIM↑	PSNR↑
Wan 2.1 (patchify)	Continuous	64	16	4	16	24	19.5	0.058	0.929	34.10
Wan 2.2 (patchify)	Continuous	48	16	4	16	27	22.6	0.052	0.932	34.54
HBQ (w/o quant)	Continuous	16	16	4	16	96	144.6	0.141	0.879	31.14
HBQ (M=4)	Discrete	16	16	4	4	384	163.6	0.148	0.872	30.40
HBQ (M=6)	Discrete	16	16	4	6	256	148.8	0.142	0.878	30.98
HBQ (M=8)	Discrete	16	16	4	8	192	144.9	0.141	0.879	31.10
HBQ (w/o quant)	Continuous	64	16	4	16	24	43.2	0.078	0.935	34.79
HBQ (M=4)	Discrete	64	16	4	4	96	50.6	0.084	0.930	33.97
HBQ* (M=4)	Discrete	64	16	4	4	96	26.3	0.068	0.938	34.73

4 Experiments

4.1 Visual Tokenizer

Implementation. We introduce two visual tokenizers: an image-only tokenizer tailored for class-conditional image generation, and a joint image-video tokenizer designed for text-to-image and text-to-video generation tasks. Both tokenizers adopt the 3D causal encoder and decoder architecture from Wan 2.1 [54] and are trained from scratch. Specifically, the image-only tokenizer is trained on the OpenImages dataset [30], while the joint tokenizer is trained on a combination of publicly available image and video datasets. During training, the overall objective comprises reconstruction, perceptual, and adversarial (GAN) losses. The respective loss weights are set to 1.0, 1.0, and 0.3 for the image-only tokenizer, and 1.0, 0.2, and 0.005 for the joint image-video tokenizer.

Results. As shown in Tab. 1, our tokenizer demonstrates state-of-the-art reconstruction performance on the 256×256 ImageNet benchmark. Utilizing four HBQ rounds, it achieves a remarkable rFID of 0.56. This result significantly surpasses not only the continuous SD-VAE (0.87 rFID) while operating at a 4× higher compression rate, but also substantially outperforms other leading methods, including RAE (0.62), VAR (0.85), LlamaGen (2.19), and Open-MAGVIT-

v2 (1.17). These results underscore our method’s superior ability to achieve high fidelity reconstruction under stringent compression.

In Tab. 2, we present a series of joint image-video tokenizers, exploring the impact of varying HBQ rounds and latent channel dimensions. We first analyze the effect of HBQ rounds and observe a clear trend: reconstruction metrics such as rFVD and PSNR consistently improve as the number of HBQ rounds increases. While four to six rounds already yield strong performance, an eight-round configuration achieves reconstruction quality nearly identical to that of the continuous baseline as depicted in Tab. 2 and Fig. 5. This demonstrates that our HBQ tokenizer can match its continuous counterpart’s fidelity while operating at a higher compression rate. Crucially, this is achieved without increasing the latent channels. While other methods (e.g., Infinity [26], BitDance [2]) can also bridge the gap to continuous models, they typically rely on expanding the latent dimension. As recent studies [10, 18, 26] suggest, such an approach often slows convergence and necessitates larger models. In contrast, our primary results are achieved without increasing the latent dimension.

We also experimented with expanding the latent channels from 16 to 64. This single change boosts the PSNR from 30.40 to an impressive 33.97. Notably, this performance is comparable to the state-of-the-art Wan 2.1 tokenizer, but is achieved at a $4\times$ higher compression rate. By carefully tuning the GAN loss weight and training for more iterations, HBQ tokenizer beats Wan 2.1 and Wan 2.2 in terms of SSIM and PSNR.

4.2 Class-to-Image Results

Implementation. Following JiT [32], we incorporate SwiGLU, RMSNorm, RoPE, qk-norm, and in-context class conditioning to the original transformer. We train GRN_{ind} with four different model sizes: 130M, 458M, 952M, and 2B, denoted as GRN-B, GRN-L, GRN-H, GRN-G, respectively. We train them for 600 epochs on the ImageNet [16] dataset. The learning rate is set to $2e-4$ and is constant during training. We randomly discard 10% conditions for Classifier-Free Guidance. During the inference stage, we grid search the best decoding hyperparameters. Additional implementation details are provided in Appendix 4.

Results. As shown in Tab. 3, GRN is benchmarked against state-of-the-art diffusion, hybrid, and autoregressive models on the ImageNet 256×256 class-conditional generation task. With nearly half the parameters of MaskGIT [7], our GRN-B model achieves a superior FID of 3.56 (vs. 6.18), demonstrating remarkable efficiency. Our largest variant, GRN-G, achieves a state-of-the-art FID of 1.81, rivaling top diffusion and hybrid models. Notably, GRN-G surpasses foundational models like DiT [43] and SiT [37] in both FID and Inception Score. This is significant as they form the backbone of many current industrial T2I and T2V models. Furthermore, GRN-G outperforms the autoregressive model LlamaGen [50] and VAR [52]. We attribute this advantage to our proposed global refinement generation, which effectively mitigates error propagation. The high-quality visual samples generated by GRN-G, shown in Fig. 1, also confirm its capabilities. Please refer to Fig. 4 in the Appendix for additional uncured

Table 3: Reference results on ImageNet 256×256 . FID [27] and IS [48] of 50K samples are evaluated. Tokenizer “D” refers to a discrete tokenizer, while “C” refers to a continuous tokenizer.

Type	Model	Tokenizer	Loss	Param	Gflops	FID↓	IS↑
Diffusion	DiT-L/2 [43]	C	MSE	458M	-	5.02	167.2
Diffusion	DiT-XL/2 [43]	C	MSE	675M	119	2.27	278.2
Flow	SiT-XL/2 [37]	C	MSE	675M	119	2.06	277.5
Flow	REPA [62], SiT-XL/2	C	MSE	675M	119	1.42	305.7
Flow	RAE [68], DiT ^{DH} -XL/2	C	MSE	839M	146	1.13	262.6
Flow	JiT-B/16 [32]	C	MSE	131M	25	3.66	275.1
Flow	JiT-L/16 [32]	C	MSE	459M	88	2.36	298.5
Flow	JiT-H/16 [32]	C	MSE	953M	182	1.86	303.4
Flow	JiT-G/16 [32]	C	MSE	2B	383	1.82	292.6
Hybrid	MAR [33]	C	MSE	943M	-	1.55	303.7
Hybrid	BitDance-H-1x [2]	D	MSE	1B	-	1.24	304.4
AR	LlamaGen-L [50]	D	CE	343M	-	3.07	256.1
AR	LlamaGen-XL [50]	D	CE	775M	-	2.62	244.1
AR	LlamaGen-XXL [50]	D	CE	1.4B	-	2.34	253.9
AR	MaskGIT [7]	D	CE	227M	-	6.18	182.1
AR	VAR-d20 [52]	D	CE	600M	-	2.57	302.6
AR	VAR-d24 [52]	D	CE	1B	-	2.09	312.9
AR	VAR-d30 [52]	D	CE	2B	-	1.92	323.1
AR	RandAR-XXL [42]	D	CE	1.4B	-	2.15	322.0
AR	GRN-B	D	CE	130M	25	3.56	280.3
AR	GRN-L	D	CE	458M	88	2.64	314.8
AR	GRN-H	D	CE	952M	182	2.06	316.1
AR	GRN-G	D	CE	2B	383	1.81	299.0

qualitative results. These strong results establish GRN as a powerful and scalable baseline for high-fidelity visual generation, motivating its application to more complex text-to-image and text-to-video tasks.

4.3 Text-to-Image Results

Implementation. We train GRN_{bit} for the text-to-image task with 2B parameters from scratch. In contrast to C2I models, the T2I model leverages in-context self-attention instead of adaln-zero to inject conditions. The model was pre-trained on large-scale public datasets and subsequently fine-tuned on a small, high-quality proprietary dataset. We first train GRN on the pre-training dataset with 256 resolution for 150K iterations using a batch size of around 15400 and a learning rate of $2e-4$. Then we fine-tune GRN at 1024 resolution with a smaller, high-quality dataset. In this stage, we train GRN for 60K iterations using a batch size of 2048 and a learning rate of $2e-5$. Additional implementation details are provided in Appendix 4. Furthermore, we observe that GRN significantly outperforms Infinity in generating fine details like small faces.

Results. As shown in Tab. 4, our model, augmented with a re-writer, achieves an overall score of 0.76 on the GenEval benchmark [21]. While our model is surpassed by larger-scale methods such as Z-Image-Turbo [5], HiDream [6], Qwen-

Table 4: Evaluation of Text-to-Image generation on GenEval [21].

Model	#Param	#Data	Single Obj.	Two Obj.	Count	Colors	Pos.	Color Attri.	Overall↑
<i>Proprietary Models</i>									
GPT Image 1 [40]	N/A	N/A	0.99	0.92	0.85	0.92	0.75	0.61	0.84
Seedream 3.0 [20]	N/A	N/A	0.99	0.96	0.91	0.93	0.47	0.80	0.84
<i>Diffusion Models</i>									
PixArt- α [9]	0.6B	N/A	0.98	0.50	0.44	0.80	0.08	0.07	0.48
SD3 Medium [18]	2B	N/A	0.98	0.74	0.63	0.67	0.34	0.36	0.62
JanusFlow [38]	1.3B	N/A	0.97	0.59	0.45	0.83	0.53	0.42	0.63
FLUX.1-Dev [31]	12B	N/A	0.98	0.81	0.74	0.79	0.22	0.45	0.66
SD3.5-Large [18]	8B	N/A	0.98	0.89	0.73	0.83	0.34	0.47	0.71
Lumina-Image-2.0 [45]	2.6B	111M	-	0.87	0.67	-	-	0.62	0.73
Show-o2 [57]	N/A	N/A	1.00	0.87	0.58	0.92	0.52	0.62	0.76
Z-Image-Turbo [5]	6B	N/A	1.00	0.95	0.77	0.89	0.65	0.68	0.82
HiDream-I1-Full [6]	17B	N/A	1.00	0.98	0.79	0.91	0.60	0.72	0.83
Z-Image [5]	6B	N/A	1.00	0.94	0.78	0.93	0.62	0.77	0.84
Qwen-Image [46]	20B	N/A	0.99	0.92	0.89	0.88	0.76	0.77	0.87
BAGEL [13]	14B	N/A	0.98	0.95	0.84	0.95	0.78	0.77	0.88
<i>Autoregressive Models</i>									
Emu3-Gen [56]	8B	N/A	0.98	0.71	0.34	0.81	0.17	0.21	0.54
Infinity ¹ [26]	2B	160M	-	0.85	-	-	0.49	0.57	0.73
Janus-Pro [11]	N/A	N/A	0.99	0.89	0.59	0.90	0.79	0.66	0.80
Tar [25]	N/A	N/A	0.98	0.92	0.83	0.85	0.80	0.65	0.84
NextStep-1 [51]	14B	N/A	-	-	-	-	-	-	0.73
BitDance [3]	14B	N/A	1.00	0.96	0.71	0.95	0.72	0.83	0.86
GRN [†]	2B	80M	0.99	0.90	0.72	0.84	0.52	0.60	0.76

Image [46], and BitDance [3], it is crucial to note the significant disparity in model size: these models utilize 6B to 20B parameters, whereas GRN is a far more compact 2B model. When compared at an equivalent scale, GRN demonstrates superior performance, significantly outperforming models of a similar 2B size like SD3 Medium [18] (0.62) and Infinity [26] (0.71). The qualitative results in Fig. 5 in the Appendix showcase GRN’s strong capability to generate high-fidelity and diverse images that accurately follow user prompts.

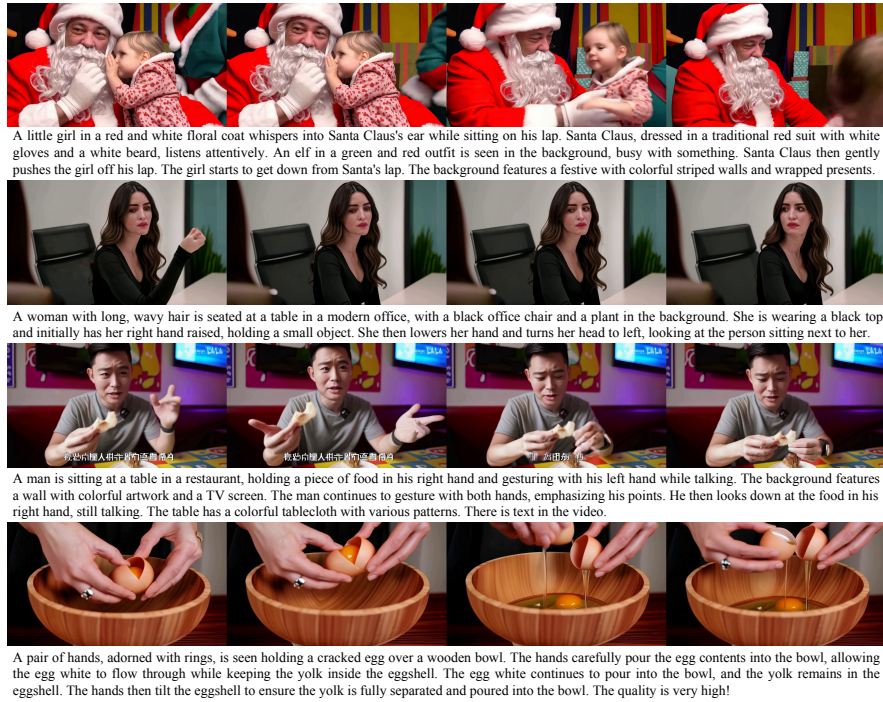
4.4 Text-to-Video Results

Implementation. Beyond the class-to-image and text-to-image generation tasks, we extend GRN to the most challenging text-to-video synthesis task. The T2V variant of GRN shares the same architecture as its T2I counterpart but is trained exclusively on video data. For this purpose, we curated a training dataset of approximately 40 million video clips, each with a resolution of at least 256×256 and a duration of 2 to 10 seconds. The training process consists of two stages. First, we train GRN at a 192p resolution for 150K iterations with a batch size of 4096 and a learning rate of $2e-4$. Subsequently, we switch to a 480p resolution for fine-tuning, training for an additional 9K iterations with a reduced batch size of 1350 and a learning rate of $2e-5$. Additional implementation details are provided in Appendix 4.

Results. As demonstrated in Tab. 5, GRN exhibits superior performance in generating videos from textual prompts. When benchmarked against contemporary diffusion and flow-based models—including AnimateDiff-V2 [23], VideoCraft-2.0 [8], OpenSora V1.2 [69], Show-1 [64], and CogVideoX-5B [59]—GRN achieves significantly higher scores across quality, semantic, and overall scores. Notably, despite having only 2B parameters, GRN surpasses the much larger CogVideoX-5B [59] model, highlighting its exceptional parameter efficiency. Furthermore,

Table 5: VBench evaluation results. [†] result is with prompt rewriting.

Models	# Params	Human Action	Scene	Multiple Objects	Appear. Style	Quality Score	Semantic Score	Overall
Diffusion / Flow Models								
AnimateDiff-V2 [23]	1.5B	92.60	50.19	36.88	22.42	82.90	69.75	80.27
VideoCrafter-2.0 [8]	1.5B	95.00	55.29	40.66	25.13	82.20	73.42	80.44
OpenSora V1.2 [69]	1.1B	85.80	42.47	58.41	23.89	80.71	73.30	79.23
Show-1 [64]	6B	95.60	47.03	45.47	23.06	80.42	72.98	78.93
URSA [15]	1.7B	-	52.30	70.60	-	83.40	78.50	82.40
CogVideoX-5B [59]	5B	99.40	53.20	62.11	24.91	82.75	77.04	81.61
HunyuanVideo [29]	13B	94.40	53.88	68.55	19.80	85.09	75.82	83.24
Wan 2.1 [54]	14B	98.80	53.67	81.44	21.13	85.64	80.95	84.70
AutoRegressive Models								
Nova [†] [14]	0.6B	95.20	54.06	77.52	20.92	80.39	79.05	80.12
Emu3 [56]	8B	77.71	37.11	44.64	20.92	84.09	68.43	80.96
Lumos-1 [63]	3.6B	-	-	-	-	79.50	73.50	78.30
InfinityStar [†] [35]	8B	96.43	52.08	78.66	21.81	84.73	79.78	83.74
GRN[†]	2B	93.75	50.44	70.83	21.30	84.41	77.35	82.99

**Fig. 6: Qualitative results of GRN (2B) on the text-to-video task.**

our approach outperforms URSA [15], a discrete diffusion model of comparable size. The performance advantage of GRN becomes even more pronounced when compared to autoregressive counterparts such as Nova [14], Emu3 [56], and Lumos-1 [49]. While the 8B parameter model, InfinityStar [35], currently

holds a higher overall score of 83.74, we are confident that the performance gap can be bridged by scaling up the size of GRN. We present qualitative results in Fig. 6. Please refer to Fig. 6 and Fig. 7 in the Appendix for additional results. The generated videos not only accurately capture the semantic details of the user prompts but also maintain a high degree of aesthetic and visual quality.

4.5 Ablation Studies

Predict Indices vs. Predict Bits As detailed in Sec. 3, GRN supports predicting either discrete indices (GRN_{ind}) or their binary representations (GRN_{bit}). We compare these two prediction targets on the 256×256 class-conditional image generation task using two model scales: GRN-B (130M) and GRN-L (458M). For each variant, we performed a grid search to identify the optimal decoding parameters, including CFG, CFG interval, and the temperature τ . The results, presented in Tab. 6, indicate that both approaches achieve comparable performance. Specifically, for the smaller GRN-B model, predicting indices yields a slightly better FID score. Conversely, for the larger GRN-L model, predicting bits proves superior, achieving a lower FID of 2.47 compared to 2.64. This suggests that GRN is well-suited for both prediction formats on the class-to-image generation task.

We further extend this comparison to the more challenging T2V generation task. As illustrated in Fig. 7, we observe that the bit prediction approach generates better videos with fewer artifacts. We hypothesize that this is because predicting bits provides a more explicit supervisory signal and mitigates the token aliasing effect inherent in index prediction, thus demonstrating superior performance on complex generation tasks. While some prior works argue that bit prediction assumes independence between bits, leading to suboptimal results, our global refinement mechanism effectively addressed this issue.



Fig. 7: Predict Indices vs. Predict Bits on the T2V task. We hypothesize that this is because predicting bits provides a more explicit supervisory signal and mitigates the token aliasing effect inherent in index prediction, thus demonstrating superior performance on complex generation tasks. While some prior works argue that bit prediction assumes independence between bits, leading to suboptimal results, our global refinement mechanism effectively addressed this issue.

Global Refinement Mechanism In the ablation study, we validate the effectiveness of our global refinement mechanism, termed Refine. We contrast its performance with a conventional mask-based generation pipeline like MaskGIT [7] or BERT [17], where previously generated tokens are fixed. The results are striking: using identical decoding hyperparameters, the mask-based approach collapses into generating nonsensical outputs (FID=185.62),

Table 6: Predict Indices vs. Predict Bits on the C2I generation task.

		CFG interval	τ	FID↓	IS↑
GRN_{ind} -B	2.4	[0.4,1]	1.33	3.56	280.3
GRN_{bit} -B	2.4	[0.44,1]	1.23	3.63	285.5
GRN_{ind} -L	2.0	[0.40,1]	1.30	2.64	314.8
GRN_{bit} -L	1.9	[0.45,1]	1.20	2.47	287.0

Table 7: Ablation on Global Refinement Mechanism.

		CFG interval	τ	FID↓	IS↑
Refine	2.4	[0.44,1]	1.23	3.63	285.5
Mask	2.4	[0.44,1]	1.23	185.62	4.3
Mask	8.0	[0.00,1]	0.50	18.13	220.2



Fig. 8: Complexity-Aware Sampling: T2I Qualitative Results.

as detailed in Tab. 7. Even with optimal decoding parameters found via grid search (higher CFG, lower temperature τ), the mask-based method (FID=18.13) still lags significantly behind our approach (FID=3.63). This experiment clearly demonstrates that our refinement paradigm effectively mitigates error propagation, a critical weakness in standard AR, and thus achieves superior performance.

Complexity-Aware Sampling We evaluate the efficacy of our complexity-aware sampling on GRN_{bit}-B, with hyperparameters set to $k = 700, b = -620$. Following standard settings in diffusion models, we set the maximum number of refinement steps to $T_{max} = 50$. To strike a balance between performance and efficiency, we empirically set the minimum number of steps to $T_{min} = 20$. We synthesize 63K images and plot the distribution of their allocated generation steps in a histogram (Fig. 9). The results demonstrate that our proposed method enables GRN to dynamically allocate computational resources based on varying levels of complexity. As observed, different examples are assigned different numbers of refinement steps, ranging from 20 to 50. Over 97.9% of samples require fewer than 50 refinement steps. Intriguingly, around 6.3% images are generated using a minimum of 20 steps, which suggests the model possesses high confidence in these particular predictions. Compared to the fixed-step baseline (50 steps for all samples), our complexity-aware sampling reduces FID from 3.56 to

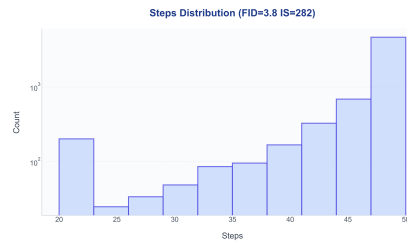


Fig. 9: Complexity-Aware Sampling.

3.47 while offering significant speedup (max $2.5\times$, avg $1.25\times$). Furthermore, we apply complexity-aware sampling to the text-to-image generation task and set $T_{min} = 10$. The qualitative results in Fig. 8 visually confirm that our method effectively enables complexity-aware, adaptive-step generation. For future work, we plan to explore refinement-step distillation, which is naturally compatible with complexity-aware sampling and enable more efficient visual generation.

Training and Inference Cost As shown in Tab. 3, we report Gflops for one forward pass in C2I. Training cost is comparable to one forward pass while inference cost scales with steps. For T2I, GRN-2B takes 2-8s per 1024×1024 image with an average of 4s (vs. Infinity 1s, FluxDev 12s, Qwen-Image 33s). For T2V, GRN takes 125s for a 480p, 81-frame video (vs. Wan2.1 480s, Wan2.2 437s). Note that these baselines use Diffusers optimizations, whereas GRN runs in bare PyTorch eager mode. We test these results using the same devices.

5 Conclusion

In this paper, we introduce GRN, a next-generation visual synthesis framework characterized by a global refinement mechanism and complexity-aware generation. We propose Hierarchical Binary Quantization (HBQ) to develop a series of discrete image and video tokenizers that are on par with their continuous counterparts while using the same number of latent channels and offering a significantly higher compression rate. In the generation phase, GRN sets new state-of-the-art results in both image reconstruction and class-conditional image generation. Extensive experiments demonstrate that, at equivalent scales, GRN surpasses existing autoregressive and diffusion-based approaches in both text-to-image and text-to-video generation tasks.

Moreover, as an autoregressive framework built entirely on discrete tokens, we believe GRN can be more naturally integrated into large language models. Unified learning over discrete text and visual tokens could substantially promote multimodal understanding and generation. At a fundamental level, GRN resolves the issues of quantization loss and error accumulation that have long limited visual autoregressive generative models. We believe it has the potential to emerge as a strong competitor to the currently dominant Transfusion [70] architecture.

6 Limitation

Due to limited computational resources, we have not scaled up the training compute or model size to the level of leading visual generation models. In addition, for the text-to-video generation task, we observe that GRN performs better in human-related scenarios. The generated videos may sometimes lack rich visual details and exhibit distortions. We will try to address these in the future.

7 Acknowledgements

We would like to thank Ruibiao Lu for his contributions to data collection and the video demo, and Hui Wu for his valuable advice on infrastructure.

References

1. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023)
2. Ai, Y., Han, J., Zhuang, S., Mao, W., Hu, X., Yang, Z., Yang, Z., Huang, H., Yue, X., Chen, H.: Bitdance: Scaling autoregressive generative models with binary tokens. arXiv preprint arXiv:2602.14041 (2026)
3. Ai, Y., Han, J., Zhuang, S., Mao, W., Hu, X., Yang, Z., Yang, Z., Huang, H., Yue, X., Chen, H.: Bitdance: Scaling autoregressive generative models with binary tokens. arXiv preprint arXiv:2602.14041 (2026)
4. Brooks, T., Peebles, B., Holmes, C., DePue, W., Guo, Y., Jing, L., Schnurr, D., Taylor, J., Luhman, T., Luhman, E., Ng, C., Wang, R., Ramesh, A.: Video generation models as world simulators. OpenAI (2024), <https://openai.com/research/video-generation-models-as-world-simulators>
5. Cai, H., Cao, S., Du, R., Gao, P., Hoi, S., Hou, Z., Huang, S., Jiang, D., Jin, X., Li, L., et al.: Z-image: An efficient image generation foundation model with single-stream diffusion transformer. arXiv preprint arXiv:2511.22699 (2025)
6. Cai, Q., Li, Y., Pan, Y., Yao, T., Mei, T.: Hidream-1l: An open-source high-efficient image generative foundation model. In: Proceedings of the 33rd ACM International Conference on Multimedia. pp. 13636–13639. ACM (2025)
7. Chang, H., Zhang, H., Jiang, L., Liu, C., Freeman, W.T.: Maskgit: Masked generative image transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11315–11325 (2022)
8. Chen, H., Xia, M., He, Y., Zhang, Y., Cun, X., Yang, S., Xing, J., Liu, Y., Chen, Q., Wang, X., et al.: Videocrafter1: Open diffusion models for high-quality video generation. arXiv preprint arXiv:2310.19512 (2023)
9. Chen, J., Yu, J., Ge, C., Yao, L., Xie, E., Wu, Y., Wang, Z., Kwok, J., Luo, P., Lu, H., et al.: Pixart: Fast training of diffusion transformer for photorealistic text-to-image synthesis. arXiv preprint arXiv:2310.00426 (2023)
10. Chen, J., Zou, D., He, W., Chen, J., Xie, E., Han, S., Cai, H.: Dc-ae 1.5: Accelerating diffusion model convergence with structured latent space. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 19628–19637 (2025)
11. Chen, X., Wu, Z., Liu, X., Pan, Z., Liu, W., Xie, Z., Yu, X., Ruan, C.: Janus-pro: Unified multimodal understanding and generation with data and model scaling. arXiv preprint arXiv:2501.17811 (2025)
12. Dehghani, M., Mustafa, B., Djolonga, J., Heek, J., Minderer, M., Caron, M., Steiner, A., Puigcerver, J., Geirhos, R., Alabdulmohsin, I.M., et al.: Patch n’pack: Navit, a vision transformer for any aspect ratio and resolution. *Advances in Neural Information Processing Systems* **36**, 2252–2274 (2023)
13. Deng, C., Zhu, D., Li, K., Gou, C., Li, F., Wang, Z., Zhong, S., Yu, W., Nie, X., Song, Z., Shi, G., Fan, H.: Emerging properties in unified multimodal pretraining. arXiv preprint arXiv:2505.14683 (2025)
14. Deng, H., Pan, T., Diao, H., Luo, Z., Cui, Y., Lu, H., Shan, S., Qi, Y., Wang, X.: Autoregressive video generation without vector quantization. arXiv preprint arXiv:2412.14169 (2024)
15. Deng, H., Pan, T., Zhang, F., Liu, Y., Luo, Z., Cui, Y., Wang, W., Shen, C., Shan, S., Zhang, Z., et al.: Uniform discrete diffusion with metric path for video generation. arXiv preprint arXiv:2510.24717 (2025)

16. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
17. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
18. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al.: Scaling rectified flow transformers for high-resolution image synthesis. In: Forty-first International Conference on Machine Learning (2024)
19. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12873–12883 (2021)
20. Gao, Y., Gong, L., Guo, Q., Hou, X., Lai, Z., Li, F., Li, L., Lian, X., Liao, C., Liu, L., et al.: Seedream 3.0 technical report. arXiv preprint arXiv:2504.11346 (2025)
21. Ghosh, D., Hajishirzi, H., Schmidt, L.: Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems* **36** (2024)
22. Guo, Y., Gan, Q., Zhang, Y., Liu, J., Hu, Y., Xie, P., Qian, D., Zhang, Y., Li, R., Zhang, Y., Lu, R., Mei, X., Han, B., Yin, X., Peng, B., Yuan, Z.: Alive: Animate your world with lifelike audio-video generation. arXiv preprint arXiv:2602.08682 (2026)
23. Guo, Y., Yang, C., Rao, A., Liang, Z., Wang, Y., Qiao, Y., Agrawala, M., Lin, D., Dai, B.: Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. arXiv preprint arXiv:2307.04725 (2023)
24. Haar, A.: Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen* **69**(3), 331–371 (1910)
25. Han, J., Chen, H., Zhao, Y., Wang, H., Zhao, Q., Yang, Z., He, H., Yue, X., Jiang, L.: Vision as a dialect: Unifying visual understanding and generation via text-aligned representations. arXiv preprint arXiv:2506.18898 (2025)
26. Han, J., Liu, J., Jiang, Y., Yan, B., Zhang, Y., Yuan, Z., Peng, B., Liu, X.: Infinity: Scaling bitwise autoregressive modeling for high-resolution image synthesis. arXiv preprint arXiv:2412.04431 (2024)
27. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems* **30** (2017)
28. Kondratyuk, D., Yu, L., Gu, X., Lezama, J., Huang, J., Schindler, G., Hornung, R., Birodkar, V., Yan, J., Chiu, M.C., Somandepalli, K., Akbari, H., Alon, Y., Cheng, Y., Dillon, J., Gupta, A., Hahn, M., Hauth, A., Hendon, D., Martinez, A., Minnen, D., Sirotenko, M., Sohn, K., Yang, X., Adam, H., Yang, M.H., Essa, I., Wang, H., Ross, D.A., Seybold, B., Jiang, L.: Videopoet: A large language model for zero-shot video generation (2024)
29. Kong, W., Tian, Q., Zhang, Z., Min, R., Dai, Z., Zhou, J., Xiong, J., Li, X., Wu, B., Zhang, J., et al.: Hunyuanvideo: A systematic framework for large video generative models. arXiv preprint arXiv:2412.03603 (2024)
30. Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Mallocci, M., Kolesnikov, A., et al.: The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision* **128**(7), 1956–1981 (2020)
31. Labs, B.F.: Flux. <https://blackforestlabs.ai/announcing-black-forest-labs/> (2024)

32. Li, T., He, K.: Back to basics: Let denoising generative models denoise. arXiv preprint arXiv:2511.13720 (2025)
33. Li, T., Tian, Y., Li, H., Deng, M., He, K.: Autoregressive image generation without vector quantization. *Advances in Neural Information Processing Systems* **37**, 56424–56445 (2024)
34. Liu, A., Feng, B., Xue, B., Wang, B., Wu, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C., et al.: Deepseek-v3 technical report. arXiv preprint arXiv:2412.19437 (2024)
35. Liu, J., Han, J., Yan, B., Wu, H., Zhu, F., Wang, X., Jiang, Y., Peng, B., Yuan, Z.: Infinitystar: Unified spacetime autoregressive modeling for visual generation. arXiv preprint arXiv:2511.04675 (2025)
36. Luo, Z., Shi, F., Ge, Y., Yang, Y., Wang, L., Shan, Y.: Open-magvit2: An open-source project toward democratizing auto-regressive visual generation. arXiv preprint arXiv:2409.04410 (2024)
37. Ma, N., Goldstein, M., Albergo, M.S., Boffi, N.M., Vanden-Eijnden, E., Xie, S.: Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In: *European Conference on Computer Vision*. pp. 23–40. Springer (2024)
38. Ma, Y., Liu, X., Chen, X., Liu, W., Wu, C., Wu, Z., Pan, Z., Xie, Z., Zhang, H., Yu, X., et al.: Janusflow: Harmonizing autoregression and rectified flow for unified multimodal understanding and generation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 7739–7751 (2025)
39. Mentzer, F., Minnen, D., Agustsson, E., Tschannen, M.: Finite scalar quantization: Vq-vae made simple. arXiv preprint arXiv:2309.15505 (2023)
40. OpenAI: Introducing gpt-4o image generation. <https://openai.com/zh-Hans-CN/index/introducing-4o-image-generation/> (2025), accessed: 2025-03-05
41. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al.: Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems* **35**, 27730–27744 (2022)
42. Pang, Z., Zhang, T., Luan, F., Man, Y., Tan, H., Zhang, K., Freeman, W.T., Wang, Y.X.: Randar: Decoder-only autoregressive visual generation in random orders. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. pp. 45–55 (2025)
43. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 4195–4205 (2023)
44. Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., Rombach, R.: Sdxl: Improving latent diffusion models for high-resolution image synthesis. arXiv preprint arXiv:2307.01952 (2023)
45. Qin, Q., Zhuo, L., Xin, Y., Du, R., Li, Z., Fu, B., Lu, Y., Yuan, J., Li, X., Liu, D., et al.: Lumina-image 2.0: A unified and efficient image generative framework. arXiv preprint arXiv:2503.21758 (2025)
46. Qwen-Image Team: Qwen-image technical report. arXiv preprint arXiv:2508.02324 (2025)
47. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 10684–10695 (2022)
48. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. *Advances in neural information processing systems* **29** (2016)

49. Shenoy, A., Lu, Y., Jayakumar, S., Chatterjee, D., Moslehpour, M., Chuang, P., Harpale, A., Bhardwaj, V., Xu, D., Zhao, S., Zhao, L., Ramchandani, A., Dong, X.L., Kumar, A.: Lumos : Empowering multimodal llms with scene text recognition (2024), <https://arxiv.org/abs/2402.08017>
50. Sun, P., Jiang, Y., Chen, S., Zhang, S., Peng, B., Luo, P., Yuan, Z.: Autoregressive model beats diffusion: Llama for scalable image generation. arXiv preprint arXiv:2406.06525 (2024)
51. Team, N., Han, C., Li, G., Wu, J., Sun, Q., Cai, Y., Peng, Y., Ge, Z., Zhou, D., Tang, H., et al.: Nextstep-1: Toward autoregressive image generation with continuous tokens at scale. arXiv preprint arXiv:2508.10711 (2025)
52. Tian, K., Jiang, Y., Yuan, Z., Peng, B., Wang, L.: Visual autoregressive modeling: Scalable image generation via next-scale prediction. arXiv preprint arXiv:2404.02905 (2024)
53. Van Den Oord, A., Vinyals, O., et al.: Neural discrete representation learning. *Advances in neural information processing systems* **30** (2017)
54. Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.W., Chen, D., Yu, F., Zhao, H., Yang, J., Zeng, J., et al.: Wan: Open and advanced large-scale video generative models. arXiv preprint arXiv:2503.20314 (2025)
55. Wang, J., Chen, Y., Yu, J., Lu, G., Pei, W.: Editinfinity: Image editing with binary-quantized generative models. arXiv preprint arXiv:2510.20217 (2025)
56. Wang, X., Zhang, X., Luo, Z., Sun, Q., Cui, Y., Wang, J., Zhang, F., Wang, Y., Li, Z., Yu, Q., et al.: Emu3: Next-token prediction is all you need. arXiv preprint arXiv:2409.18869 (2024)
57. Xie, J., Yang, Z., Shou, M.Z.: Show-o2: Improved native unified multimodal models. arXiv preprint arXiv:2506.15564 (2025)
58. Yan, W., Zhang, Y., Abbeel, P., Srinivas, A.: Videogpt: Video generation using vq-vae and transformers. arXiv preprint arXiv:2104.10157 (2021)
59. Yang, Z., Teng, J., Zheng, W., Ding, M., Huang, S., Xu, J., Yang, Y., Hong, W., Zhang, X., Feng, G., et al.: Cogvideox: Text-to-video diffusion models with an expert transformer. arXiv preprint arXiv:2408.06072 (2024)
60. Yin, T., Gharbi, M., Park, T., Zhang, R., Shechtman, E., Durand, F., Freeman, B.: Improved distribution matching distillation for fast image synthesis. *Advances in neural information processing systems* **37**, 47455–47487 (2024)
61. Yin, T., Gharbi, M., Zhang, R., Shechtman, E., Durand, F., Freeman, W.T., Park, T.: One-step diffusion with distribution matching distillation, 2024. URL <https://arxiv.org/abs/2311.18828>
62. Yu, S., Kwak, S., Jang, H., Jeong, J., Huang, J., Shin, J., Xie, S.: Representation alignment for generation: Training diffusion transformers is easier than you think. arXiv preprint arXiv:2410.06940 (2024)
63. Yuan, H., Chen, W., Yu, H., Liang, J., Chang, S., Lin, Z., Feng, T., Liu, P., Xing, J., Luo, H., et al.: Lumos-1: On autoregressive video generation with discrete diffusion from a unified model perspective. In: *The Fourteenth International Conference on Learning Representations*
64. Zhang, D.J., Wu, J.Z., Liu, J.W., Zhao, R., Ran, L., Gu, Y., Gao, D., Shou, M.Z.: Show-1: Marrying pixel and latent diffusion models for text-to-video generation. *International Journal of Computer Vision* pp. 1–15 (2024)
65. Zhang, H., Wu, Z., Xing, Z., Shao, J., Jiang, Y.G.: Adadiff: adaptive step selection for fast diffusion models. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. vol. 39, pp. 9914–9922 (2025)

66. Zhang, Y., Yang, H., Zhang, Y., Hu, Y., Zhu, F., Lin, C., Mei, X., Jiang, Y., Peng, B., Yuan, Z.: Waver: Wave your way to lifelike video generation. arXiv preprint arXiv:2508.15761 (2025)
67. Zhao, Y., Xiong, Y., Krähenbühl, P.: Image and video tokenization with binary spherical quantization. arXiv preprint arXiv:2406.07548 (2024)
68. Zheng, B., Ma, N., Tong, S., Xie, S.: Diffusion transformers with representation autoencoders. arXiv preprint arXiv:2510.11690 (2025)
69. Zheng, Z., Peng, X., Yang, T., Shen, C., Li, S., Liu, H., Zhou, Y., Li, T., You, Y.: Open-sora: Democratizing efficient video production for all. arXiv preprint arXiv:2412.20404 (2024)
70. Zhou, C., Yu, L., Babu, A., Tirumala, K., Yasunaga, M., Shamis, L., Kahn, J., Ma, X., Zettlemoyer, L., Levy, O.: Transfusion: Predict the next token and diffuse images with one multi-modal model. arXiv preprint arXiv:2408.11039 (2024)