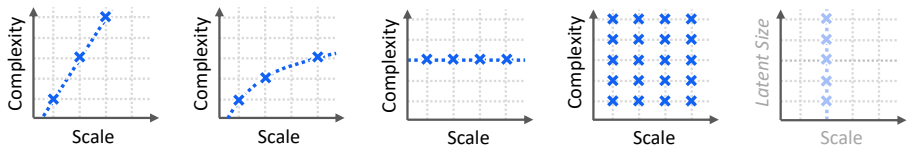
Decoupling Complexity from Scale in Latent Diffusion Model

Tianxiong Zhong, Xingye Tian, Xuebo Wang*, Boyuan Jiang, Xin Tao, and Pengfei Wan

Kling Team, Kuaishou Technology, Beijing, China
inkosizhong@gmail.com, {tianxingye,wangxuebo,jiangboyuan}@kuaishou.com
{taoxin,wanpengfei}@kuaishou.com

Abstract. Existing latent diffusion models typically couple scale with content complexity, using more latent tokens to represent higher-resolution images or higher–frame rate videos. However, the latent capacity required to represent visual data primarily depends on content complexity, with scale serving only as an upper bound. Motivated by this observation, we propose DCS-LDM, a novel paradigm for visual generation that decouples information complexity from scale. DCS-LDM constructs a hierarchical, scale-independent latent space that models sample complexity through multi-level tokens and supports decoding to arbitrary resolutions and frame rates within a fixed latent representation. This latent space enables DCS-LDM to achieve a flexible computation–quality tradeoff. Furthermore, by decomposing structural and detailed information across levels, DCS-LDM supports a progressive coarse-to-fine generation paradigm. Experimental results show that DCS-LDM delivers performance comparable to state-of-the-art methods while offering flexible generation across diverse scales and visual qualities.



(a) Vanilla LDM **(b)** DetailFlow **(c)** VFRTok **(d) Ours** **(e)** FlexTok

Fig. 1: Comparison of how different frameworks model content complexity and scale. (a–b) Couple scale and complexity, requiring larger latents for higher resolutions or frame rates. (c) Models only a fixed level of complexity. **(d) Our approach decouples complexity from scale, enabling flexible and efficient trade-offs.** (e) Utilizes arbitrary length latents to model fixed level of both scale and complexity.

* Corresponding author.

1 Introduction

Latent Diffusion Models (LDMs) have recently emerged as a powerful framework for visual generation [1, 23, 26, 30, 39, 43]. An LDM employs a tokenizer to embed high-dimensional images or videos into a compressed latent representation [6, 8, 9, 40, 50], followed by a diffusion model that performs denoising in the latent space [27, 29, 44, 46]. This latent space inherently provides a rate–distortion trade-off [4, 25]: larger latent representations possess higher capacity to preserve fine details of the visual data, but at the cost of increased computational overhead. Meanwhile, users exhibit diverse demands for generation scale, such as varying resolutions and frame rates. For example, mobile users may prefer high–frame rate, vertical videos optimized for smartphone displays, whereas professional creators may seek ultra–high resolution, horizontal videos tailored for large cinema screens.

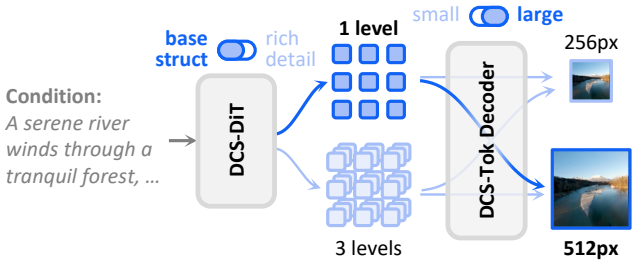
As illustrated in Figure 1, mainstream LDMs couple the scale of visual data with its content complexity. They allocate more latent tokens to higher-resolution or higher–frame rate data, assuming that scale directly reflects the complexity. However, this assumption overlooks the variability in information content across samples. While input size determines the number of pixels, the upper bound of representable information, the actual information content depends primarily on the semantic and structural complexity of the visual data. For example, a large-scale animation may contain far less information than a small-scale nature documentary. From an information–theoretic perspective, this implies that each sample should have its own rate–distortion curve, meaning that the number of latent tokens required to achieve comparable fidelity should vary across samples.

Motivated by these insights, we propose DCS-LDM, a **L**atent **D**iffusion **M**odel that **D**ecouples information **C**omplexity from the sample **S**cale. As illustrated in Figure 1d, DCS-LDM decomposes complexity and scale into two orthogonal dimensions, enabling flexible representation of diverse samples. Specifically, it employs a scale-independent latent space, aligning images and videos of different scales into a unified representation, and structures this space hierarchically, where additional token levels model higher information complexity. As shown in Figure 2, DCS-LDM comprises two key components: the tokenizer DCS-Tok and the transformer-based diffusion model DCS-DiT. During generation, quality is controlled by the number of latent token levels generated by DCS-DiT, while scale can be freely chosen during DCS-Tok decoding.

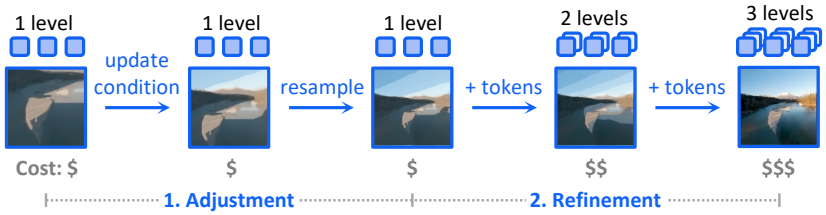
These designs enable DCS-LDM to achieve a flexible computation–quality tradeoff. We further structure the latent space causally, assigning basic structural information to earlier levels and fine-grained details to later ones. DCS-LDM thus supports a coarse-to-fine progressive generation paradigm, maintaining identical computational cost while reducing latency and enabling efficient refinement [18].

In summary, our main contributions are as follows:

1. We reveal the inherent differences in content complexity among visual samples and introduce a decoupling perspective between scale and complexity.
2. We propose DCS-LDM, a novel latent diffusion paradigm that naturally supports flexible computation–quality trade-offs and coarse-to-fine generation.



(a) DCS-LDM decouples information complexity from data scale. The latents generated by DCS-DiT are hierarchical, capturing information ranging from global structure to fine details. The DCS-Tok decoder can decode data from these latents at arbitrary scales.



(b) DCS-LDM enables coarse-to-fine progressive generation, allowing users to flexibly adjust content and structure with lower cost and latency, and incrementally increase tokens to obtain finer details.

Fig. 2: Overview of DCS-LDM.

3. Experiments demonstrate that DCS-LDM achieves comparable reconstruction and generation performance to state-of-the-art methods, while supporting more diverse and scalable functionalities.

2 Related Work

2.1 Vanilla Latent Diffusion Models

LDMs [27,29,30,44,46] were introduced to improve the efficiency of pixel-space diffusion models [16,33]. An LDM typically consists of two core components: a visual tokenizer and a diffusion model. This design has been widely adopted in both image [2,6,7,23,24,30,37,39,44] and video [1,14,21,26,43,49,50] generation, driving rapid progress in visual generative modeling. In LDMs, the capacity of the latent space plays a crucial role in balancing efficiency and visual quality. The tokenizer bridges the pixel space and the latent space, providing diverse strategies for information compression and representation. Vanilla LDMs [23,30,43,44] commonly employ CNN-based tokenizers to perform fixed-ratio downsampling. For example, in video generation, it is typical to downsample the spatial and temporal dimensions by factors of 16 and 4, respectively. Consequently, as the data scale increases, the number of tokens expands linearly with scale (Figure 1a).

2.2 Novel Visual Tokenizers

Recently, several 1D tokenizers have been proposed [6,7,24,45,50], which map grid-based visual patches into a 1D sequence, removing explicit spatial relationships. These methods typically adopt ViT-based architectures. During encoding, the input image or video is divided into patches, learnable query tokens are concatenated, and only these tokens are retained as latents. During decoding, the query tokens are concatenated to the latent tokens to reconstruct each patch.

This flexible formulation allows for the construction of tokenizers under different information assumptions, controlled by the number of patch or latent tokens [2,24,50]. For instance, VFRTok [50] aligns a varying number of patch tokens into a fixed-length latent, enabling variable frame-rate generation. FlexTok [2] and SEMANTICIST [42] instead model fixed-size images using latents of arbitrary length. However, their flow-based decoding paradigm introduces randomness, meaning that increasing the number of tokens may not refine the details but entirely alter image content. DetailFlow [24] further explores the sub-linear relationship between scale and complexity from an entropy perspective. However, the existing visual tokenizers overlook the variation in content complexity across samples, which limits their ability to adapt latent capacity.

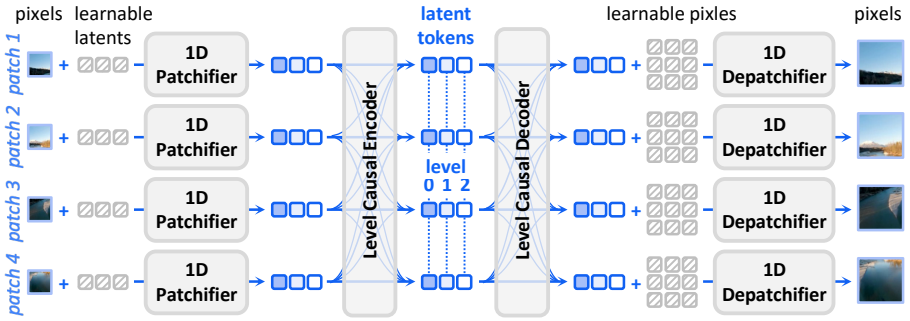
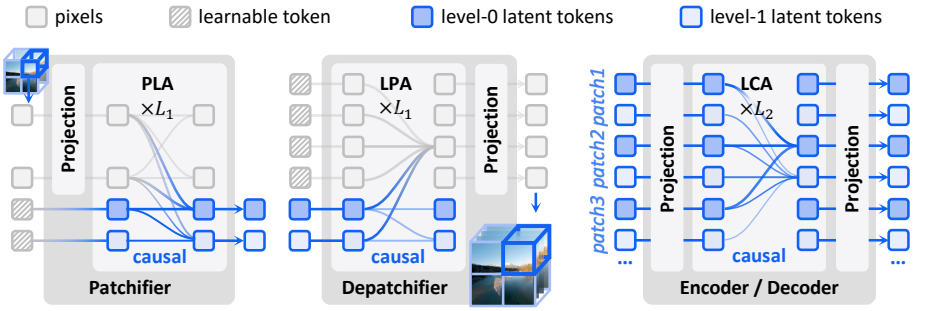


Fig. 3: DCS-Tok adopts a pure Transformer architecture, comprising a 1D patchifier–depatchifier and a causal encoder–decoder. The patchifier–depatchifier align scale-aware spatial–temporal patches into a causal sequence, where lower-level tokens capture structure and higher-level tokens encode fine-grained textures. The encoder–decoder facilitates cross-patch interaction while preserving causal dependencies across levels.

3 Method

To construct the hierarchical, scale-independent latent space, we introduce a carefully designed tokenizer (Figure 3) together with a corresponding DiT architecture (Figure 5a). The tokenizer incorporates a novel 1D patchifier that operates on a fixed number of patches, enabling the model to encode inputs of



(a) Each pixel is projected as an independent token. Lines indicate information flow: pixels to latents in the patchifier, latents to pixels in the depatchifier.

(b) Lines indicate information flow: a token can attend only to tokens at its own level or lower across all patches.

Fig. 4: Detailed architectures of DCS-Tok. (a) Patchifier and depatchifier. (b) Encoder and decoder. PLA, LPA, and LCA denote Pixel-to-Latent, Latent-to-Pixel, and Level Causal Attention, respectively. L_i indicates the number of blocks.

varying scales into a unified multi-level latent representation. More levels of latent tokens capture fine-grained details for complex samples, while fewer levels suffice for simpler ones, thereby effectively decoupling scale from complexity.

To better illustrate how the model structure aligns with our design principles, we detail each component according to its objective in the following sections. We first describe how data of different scales are mapped into a unified latent space in Section 3.1. Then, we present the construction of hierarchical representations in Section 3.2. Finally, we discuss the coarse-to-fine generation paradigm in detail in Section 3.3. Additional implementation details, such as positional embeddings, attention masks, and temporal causality, are provided in the supplementary material (Section A).

3.1 Scale-independent Latent Space

Scale-aware Patchify. As detailed in Figure 3, instead of fixing the patch size [23, 30, 39], we fix the number of patches, and determine the size based on its resolution and frame rate. Specifically, for a video with resolution $h \times w$ and frame rate f, the spatial and temporal patch sizes, $p_{h,w}$ and p_t, are defined as:

$$p_{h,w} = \frac{\min(h, w)}{k}, \quad p_t = \frac{f}{k_t}, \tag{1}$$

where k is the number of patches along the shorter spatial dimension and k_t is the number of patches within a second.

For example, as illustrated in Figure 4a, the same video is patchified at different scales with $k = 2$ and $k_t = 1$. The smaller one (top-left) has a resolution of 16×16 and frame rate 2 fps, resulting in $p_{h,w} = 8$ and $p_t = 2$. The larger copy (bottom-right) has a resolution of 32×32 and frame rate 4 fps, giving

$p_{h,w} = 16$ and $p_t = 4$. This approach aligns the same spatial–temporal regions across scales. For images, we treat them as single-frame videos with $p_t = 1$.

Asymmetric scale alignment. To enable decoding at arbitrary resolutions and frame rates from a unified latent representation, DCS-Tok aligns patches across different scales. As shown in Figure 3, we adopt asymmetric training [50] where the input and output of DCS-Tok are randomly chosen scale variants of the same sample during training.

3.2 Hierarchical Representation

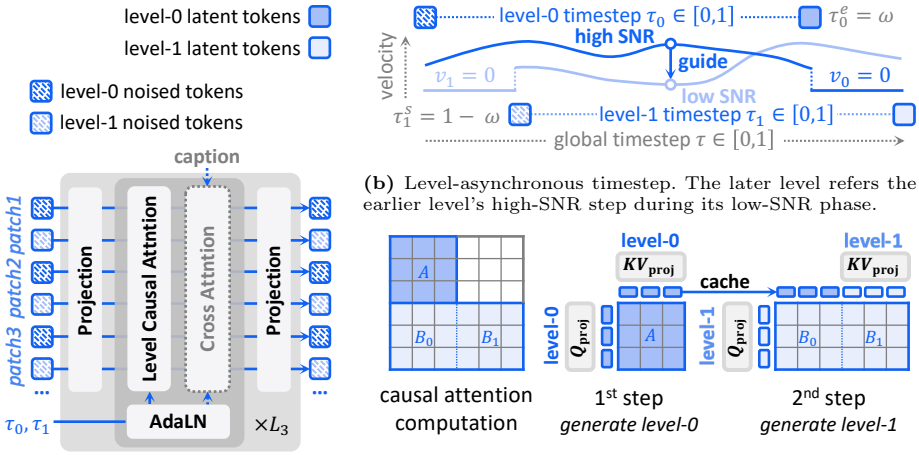
1D patchifier. As detailed in Figure 4a, we implement the patchifier and depatchifier in a 1D tokenizer style [6, 45, 50], using learnable tokens to query information from the pixels in each patch. All pixels in a patch are treated as independent tokens and concatenated with n learnable latent tokens, numbered sequentially from 0 to $n-1$. Symmetrically, in the depatchifier, learnable pixel tokens are concatenated before the latent tokens to reconstruct pixel information. This architecture enables DCS-Tok to encode scale-aware patches into a hierarchical representation.

Autoencoder. Since the patchifier processes each patch independently, the autoencoder is responsible for modeling inter-patch interactions. As shown in Figure 4b, the encoder and decoder share the same architecture. The resulting latent representation preserves the spatio-temporal structure of the input while introducing an additional hierarchical dimension. For a video with t frames and spatial resolution $h \times w$, the representation has the shape $\frac{t}{p_t} \times \frac{h}{p_{h,w}} \times \frac{w}{p_{h,w}} \times n$, where n denotes the number of levels.

DiT. DCS-DiT models the transformation between the hierarchical latents and pure noise, with the number of generated levels determined by the user-specified noise level. To support hierarchical latent token generation, DCS-DiT extends conventional DiTs [27, 29, 44]. As shown in Figure 5a, it adopts an architecture similar to the autoencoder in DCS-Tok and introduces optional cross-attention blocks for injecting conditional information.

3.3 Coarse-to-fine Generation

Level causality. All components of DCS-LDM are designed with level causality, ensuring that each latent level depends only on the preceding one. As shown in Figure 4, we illustrate three key mechanisms: Pixel-to-Latent Attention (PLA), Latent-to-Pixel Attention (LPA), and Level Causal Attention (LCA). We also highlight the information flow within different modules. Unlike prior 1D tokenizer designs [6, 7, 45, 50] that use full attention across all tokens, PLA and LPA compute full attention only among pixel tokens. Information exchange between pixel and latent tokens is strictly unidirectional, preventing violation of level causality among latent tokens. For LCA, information is allowed to flow across patches, but strictly in a unidirectional manner: information from earlier levels can influence later levels, but not vice versa.



(a) Architecture. DCS-DiT extends autoencoder of the DCS-Tok by adding Adaptive Layer Normalization (AdaLN) and Cross Attention blocks.

(b) Level-asynchronous timestep. The later level refers the earlier level's high-SNR step during its low-SNR phase.

(c) Coarse-to-fine generation. Causality separates computation by token level, and caching keys and values from earlier levels keeps progressive generation as efficient as non-progressive generation.

Fig. 5: DCS-DiT enables hierarchical generation.

Structured representation. To enable reconstruction from an arbitrary number of token levels, our objective is to encode essential structural information in the earlier levels and reserve later levels for fine-grained details. To achieve this, we randomly assign each patch between 1 and n token levels during training. Because DCS-Tok is optimized to reconstruct patches even when later level tokens are missing, it naturally learns the desired coarse-to-fine encoding scheme. Operationally, when a patch is assigned $1 \leq m \leq n$ token levels, an attention mask is applied to deactivate the remaining $n - m$ tail-level tokens. During inference (batch size $= 1$), this simply amounts to discarding the unused tokens. More details are provided in Section A.1.

Level-asynchronous timestep. Since fine-grained details inherently depend on the structural priors established at earlier levels, we introduce the level-asynchronous timestep strategy for DCS-DiT generation. The key intuition is to enforce a temporal ordering in the signal-to-noise ratio (SNR), such that structural information consistently achieves higher SNR before detailed information. As illustrated in Figure 5b, we construct independent yet staggered timelines for each level [28]. During both training and inference, we first sample a global timestep τ. We then introduce a scaling factor ω, where $1/n \leq \omega \leq 1$, to control the ratio between the per-level time span and the global time range. The timeline of each level is uniformly allocated as:

$$\tau_i^s = \frac{1-\omega}{n-1} \cdot i, \quad \tau_i^e = \tau_i^s + \omega, \quad 0 \leq i \leq n-1, \tag{2}$$

where τ_i^s and τ_i^e denote the start and end timesteps of the i-th level, respectively. Notably, $\omega = 1$ degenerates to SNR-consistent generation across all lev-

els, whereas smaller ω values enforce stronger temporal separation, resulting in higher SNR for earlier levels at the same global timestep. By adjusting ω, we achieve a flexible trade-off between computational parallelism and generation quality. The velocity v_i at level i is predicted only when the sampled global timestep τ falls within $[\tau_i^s, \tau_i^e]$, ensuring that each level is activated within its designated temporal window. As shown in Figure 5, the per-level timestep condition is re-normalized as $\tau_i = \text{clamp}(\tau - \tau_i^s)/\omega, 0, 1)$, which is used to interpolate both noise and token representations. Correspondingly, the predicted velocity is scaled by ω to maintain consistent magnitude across levels. More details are shown in Section A.3.

Cache-based acceleration. With the above components in place, DCS-LDM naturally supports coarse-to-fine generation. By introducing a caching mechanism, the model can switch from generating all m token levels at once to generating them sequentially, while keeping the total computational cost unchanged. Using two token levels as an example, Figure 5c illustrates the attention operations within each DCS-DiT block and the DCS-Tok decoder. Let x_i denote all tokens at level-i. The tokens in each level are first projected into query, key, and value features as:

$$q_0 = Q_{\text{proj}}(x_0),\ k_0 = K_{\text{proj}}(x_0),\ v_0 = V_{\text{proj}}(x_0), \tag{3}$$

$$q_1 = Q_{\text{proj}}(x_1),\ k_1 = K_{\text{proj}}(x_1),\ v_1 = V_{\text{proj}}(x_1). \tag{4}$$

As highlighted in Figure 5c, the effective attention computations include parts A, B_0, and B_1, defined as:

$$A : \text{attn}(q_0, k_0, v_0), \tag{5}$$

$$B_1 : \text{attn}(q_1, k_0, v_0),\ B_2 : \text{attn}(q_1, k_1, v_1). \tag{6}$$

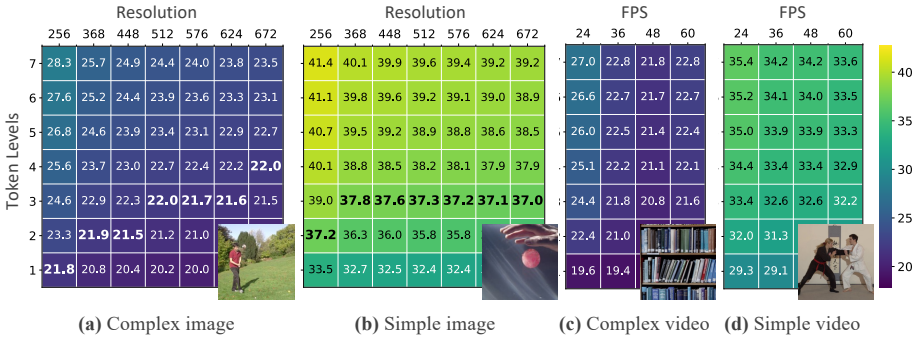
During coarse-to-fine generation, level-0 tokens are first generated using Equations (3) and (5), followed by level-1 tokens using Equations (4) and (6). Since the computation of B_0 depends on k_0 and v_0, these features are cached [10] from the first step, enabling efficient progressive generation.

On the other hand, as illustrated in Figure 5b, when $\tau > \tau_i^e$, the velocity prediction for the i-th level is set to $v_i = 0$, and the corresponding branch degenerates to an identity mapping after DCS-DiT. The representation of this level remains unchanged and can be directly reused. When such a level provides conditional features for subsequent levels, we further cache its key and value tensors to eliminate redundant computations, thereby maintaining consistent computational cost across different ω settings. Additional implementation details are provided in Section A.3.

4 Experiments

4.1 Setup

Implementation. We employ ViT-B [12] as the backbone for the encoder and decoder ($L_2 = 12$), a tiny ViT ($L_1 = 3$) for the patchifier and depatchifier in



(a) Complex image **(b)** Simple image **(c)** Complex video **(d)** Simple video

Fig. 6: Image and video reconstruction via DCS-Tok across samples with varying information complexity, scale, and latent token levels. Brighter background indicates higher PSNR. Configurations with comparable PSNR are highlighted in bold.

DCS-Tok, and ViT-XL for DCS-DiT ($L_3 = 28$). Following LightningDiT [44], we adopt modern Transformer components such as SwiGLU FFN [32] and RMSNorm [47]. We set $k = 16$, with $k_t = 1$ for images and $k_t = 6$ for videos, corresponding to a spatial downsampling factor of $p_{h,w} = 16$ at 256×256 resolution and a temporal factor of $p_t = 4$ at 24 fps [39, 44]. Both label-guided and text-guided generation are implemented using Classifier-Free Guidance [17] (CFG), where the cross-attention blocks are dropped in the label-guided image and video generation. We train DCS-Tok as a deterministic tokenizer with reconstruction objectives:

$$\mathcal{L}_{\text{Tok}} = \mathcal{L}_{\text{recon}} + \lambda_1 \mathcal{L}_{\text{percept}} + \lambda_2 \cdot \lambda_\nabla \mathcal{L}_{\text{adv}}, \tag{7}$$

where $\mathcal{L}_{\text{recon}}, \mathcal{L}_{\text{percept}}, \mathcal{L}_{\text{adv}}$ are the L1 reconstruction loss, perceptual loss [19, 22], and adversarial loss [13], respectively. The factors are set to $\lambda_1 = 1$, $\lambda_2 = 0.2$, and λ_∇ represents adaptive weight [6, 50]. The entire training process is divided into multiple stages: low resolution image initialization, spatial asymmetric training, and spatial-temporal asymmetric training. DCS-DiT follows the sampling method and training objectives of LightningDiT [44]. Unless otherwise specified, we set $\omega = 1/n$. All models are trained on four nodes, each equipped with eight GPUs. Training DCS-Tok takes approximately one week, while DCS-DiT requires about four days. More details are given in the Section A.4.

Datasets. The training data for DCS-Tok comprise the ImageNet [11], Koala-36M [41], and LAVIB [35] datasets. The Koala-36M and LAVIB dataset provide videos with relatively high resolutions and frame rates, allowing DCS-Tok to be trained across diverse configurations ranging from 128p to 720p and 12 fps to 60 fps. For DCS-DiT, due to computational constraints, the qualitative experiments are conducted on a text-to-image model trained using the ImageNet [11] and SA-1B [20] datasets, with captions generated by Qwen-VL 2.5 [3]. Moreover, to ensure fair comparison with existing methods [1, 2, 5, 6, 24, 27, 29, 36, 40, 44, 46, 50], we additionally train label-based image and video generation models. For

Fig. 7: Text-to-image generation across samples with varying information complexity and latent token counts.

label-to-image generation, the ImageNet dataset [11] is used for both training and evaluation, while for label-to-video generation, the UCF101 dataset [34] is incorporated for both training and evaluation.

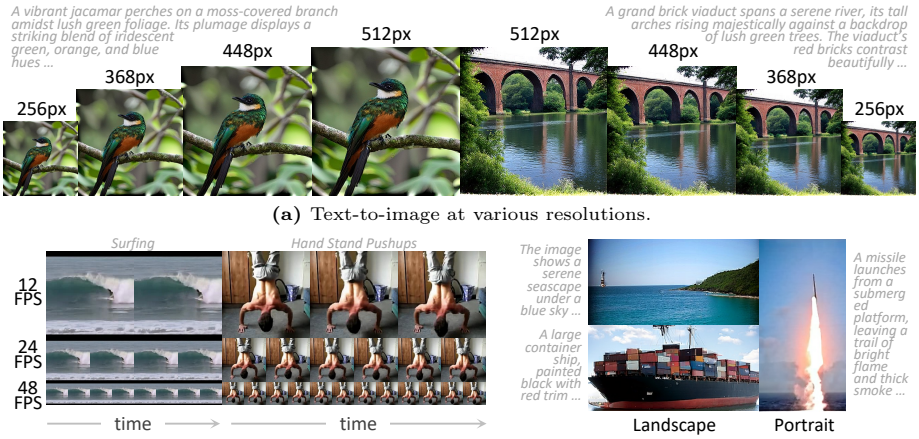
Metrics. We evaluate reconstruction quality using PSNR and SSIM for pixel-level assessment, and Fréchet Inception Distance (rFID) [15] and Fréchet Video Distance (rFVD) [38] for perceptual quality on images and videos, respectively. For generation, with and without Classifier-Free Guidance (CFG) [17], we employ gFID [15] and gFVD [38] to measure generation quality, and Inception Score (IS) [31] to assess semantic diversity.

4.2 Complexity Matters

Figure 6 shows the reconstruction quality of different samples at different scales with various token levels. We can draw the following conclusions:

Statistically sublinear scaling between scale and latent capacity. For a given sample, to maintain similar reconstruction quality (PSNR), the growth trend of the scale and number of latents exhibits a sublinear relationship. In contrast, the most widely used linear assumption [23, 30, 39, 43, 44] leads to increasingly better reconstruction quality as the scale and latent capacity increase. On the other hand, fixed-latent strategies similar to VFRTok [50] result in an inverse relationship between reconstruction quality and scale. The assumption of DetailFlow [24] is reasonable in the statistics of a large number of samples.

Content complexity dominates reconstruction quality. Different samples with varying content complexity will result in different rate-distortion curves [4, 25]. For example, at the same scale, simple samples can reach reconstruction quality comparable to complex ones using significantly fewer latent tokens. Similarly, under a fixed latent budget, a large but simple sample may achieve higher



(a) Text-to-image at various resolutions.

(b) Label-to-video at various frame rates.

(c) Text-to-image at various aspect ratios.

Fig. 8: DCS-LDM generates data at varying resolutions and frame rates via scale-independent latents and supports flexible aspect ratios.

reconstruction quality than a small yet complex sample. Similarly, we can observe the impact of content complexity in generation tasks. As illustrated in Figure 7, for relatively simple content, DCS-LDM achieves satisfactory generation results using only first-level tokens. For complex samples, DCS-LDM uses four levels of tokens to render rich details. This also demonstrates that DCS-LDM provides highly flexible efficiency-quality control by decoupling complexity and scale.

4.3 Variable Scale Generation

Resolution & frame rate. As shown in Figure 8, DCS-LDM supports generating data at multiple scales via the scale-independent latent space. Given a latent representation, the DCS-Tok decoder can provide outputs at arbitrary resolutions and frame rates. For image generation, we fix DCS-DiT to produce four levels of latent tokens and decode them into images ranging from 256px to 512px. (Figure 8a) For video generation, DCS-Tok decodes a single level of generated tokens to synthesize results at 12 fps, 24 fps, and 48 fps. (Figure 8b) Across all scales, the generated outputs preserve consistent semantics and structure.

Aspect ratio. Figure 8c illustrates the generation results with varying aspect ratios, enabled by the preservation of spatio-temporal structures in the latent space. For instance, to generate a landscape image with a $h : w = 1 : 2$ aspect ratio, setting the number of patches on the vertical side to $k = 16$ yields 32 patches on the horizontal side. This design enables DCS-DiT to infer the desired aspect ratio through noise-shape perception.

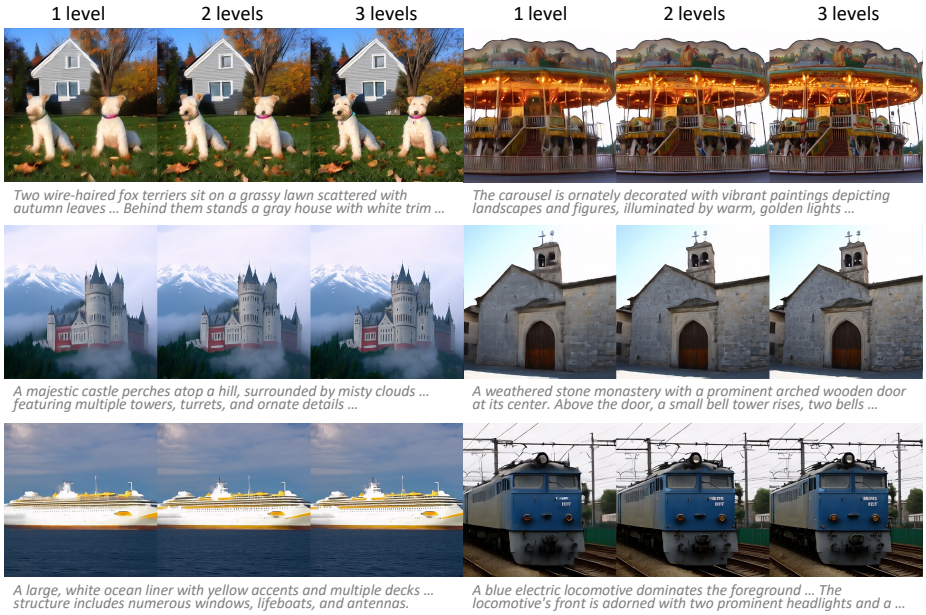


Fig. 9: Coarse-to-fine text-to-image generation on ImageNet. Increasing token levels progressively enriches image details. Users can start with fewer token levels for a quick preview and stop once the desired detail is reached.

4.4 Coarse-to-fine Generation

DCS-LDM supports level-by-level, coarse-to-fine generation, prioritizing the synthesis of essential structures before progressively adding finer details. Figure 9 shows samples generated using one to three token levels. For each row, adding more tokens enhances the visual details of the sample. This paradigm allows users to quickly generate draft ideas at lower computational cost and latency, progressively refining them by adding tokens until the desired quality is reached.

We also conduct an ablation study of the level-asynchronous timestep scaling factor. As shown in Table 1, when $\omega < 1$, adding more levels can improve the generation quality, and the result is optimal when $\omega = 1/4$. In contrast, level-synchronous generation ($\omega = 1$) causes the image quality to decrease as the level increases. As shown in Figure 10, the lack of guidance from earlier layers to later layers in the latents ultimately leads to misaligned details.

4.5 System-level Comparison

We also compare DCS-LDM with existing image and video generation frameworks [24, 29, 40, 44, 50]. To ensure fairness, we follow the experimental settings commonly adopted in prior work [6, 44, 50], evaluating image reconstruction and label-to-image generation at a resolution of 256×256 on ImageNet [11],

ω	gFID↓ w/o CFG			
	1 level	2 levels	3 levels	4 levels
1	**40.42**	48.26	62.23	71.25
1/2	42.31	40.30	38.48	38.03
1/4	42.50	**40.23**	**38.39**	**37.80**

Table 1: Ablation study of level-asynchronous timestep scaling factor ω using DCS-DiT/B.

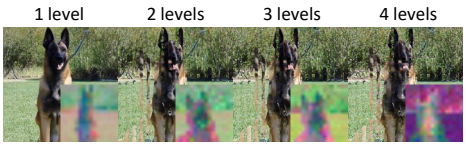


Fig. 10: Level-synchronous timestep $\omega = 1$ causes detail misalignment. Bottom right: top-3 PCs of generated latents.

Table 2: System-level comparison on ImageNet 256 × 256 label-to-image task. †DCS-LDM can incorporate semantic alignment techniques to further improve generation.

Method	Reconstruction					w/o CFG		w/ CFG	
	#Tok	#Dim	PSNR↑	SSIM↑	rFID↓	gFID↓	IS↑	gFID↓	IS↑
AutoRegressive									
MaskGiT [5]	256	256	-	-	2.28	6.18	182.1	-	-
LlamaGen [36]	256	8	20.79	0.675	2.19	9.38	112.9	2.18	263.3
FlexTok [2]	256	6	-	-	1.45	2.50	-	1.86	-
DetailFlow [24]	256	8	20.08	0.670	0.80	-	-	2.62	-
Latent Diffusion Model									
DiT [29]						9.62	121.5	2.77	278.2
SiT [27]	1024	4	26.27	0.759	0.60	8.61	131.7	2.06	270.3
REPA† [46]						5.90	157.8	1.42	305.7
LightningDiT† [44]	256	32	27.63	0.811	0.28	2.17	205.6	1.35	295.3
Ours (1 level)	256	16	26.36	0.766	0.79	7.26	119.0	2.44	277.3
+ REPA†						4.26	147.1	1.39	311.6

and video reconstruction and label-to-video generation at 256 × 256 and 24 fps on UCF101 [34]. Across all tasks, DCS-LDM only uses the first level tokens. DCS-LDM delivers reconstruction and generation quality on par with vanilla LDMs [1, 27, 33, 40, 50], as detailed in Tables 2 and 3. Furthermore, incorporating techniques such as representation alignment in RePA [46], LightningDiT [44], and MAETok [6] into DCS-LDM may further enhance the quality of the generation. These results demonstrate that DCS-LDM's novel structural design not only introduces flexibility but also maintains competitive performance in both reconstruction and generation.

Table 3: System-level comparison on UCF101 24fps and 256 × 256 label-to-video task.

Method	Reconstruction					w/o CFG	w/ CFG
	#Tok	#Dim	PSNR↑	SSIM↑	rFVD↓	gFVD↓	gFVD↓
OmniTokenizer [40]	4096	8	29.36	0.931	13.29	480.60	88.89
Cosmos [1]	2048	16	31.67	0.908	14.96	497.01	85.22
VFRTok [50]	512	32	31.68	0.923	15.34	377.50	71.34
Ours (1 level)	1024	16	29.88	0.914	18.43	421.66	77.10

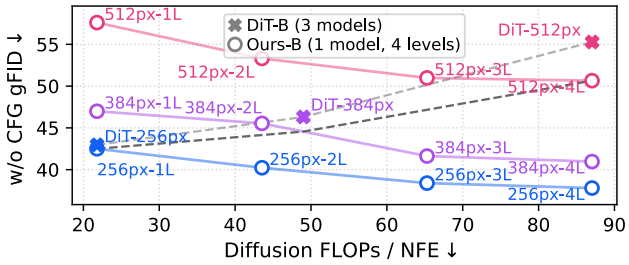


Fig. 11: Efficiency-quality trade-off, where nL indicates n token levels.

Table 4: End-to-end efficiency metrics, where batch=16 for throughput and batch=4 for latency and peak memory.

	Method	Tokenizer Decode			Diffusion XL 50NFE			Total
		Throughput↑	Latency↓	Memory↓	Throughput↑	Latency↓	Memory↓	Latency↓
256px	DiT [29]	149.4	30.42ms	0.67G	3.126	1572ms	2.57G	1602ms
	Ours (1 level)	52.32	87.04ms	3.54G	3.124	1573ms	2.68G	1660ms
512px	DiT [29]	32.46	131.5ms	2.18G	0.657	6612ms	3.01G	6743ms
	Ours (1 level)	9.665	370.4ms	14.9G	3.124	1573ms	2.68G	**1943ms**
	Ours (2 levels)	9.595	391.1ms	15.1G	1.468	3280ms	3.18G	**3671ms**
	Ours (4 levels)	9.427	414.2ms	15.4G	0.664	6972ms	3.46G	7386ms

4.6 Efficiency-Quality Trade-off

We compare DCS-LDM with independently trained DiT-B baselines at 256px, 384px, and 512px. DCS-LDM uses a single unified model with $\omega = 1/4$ and generates different resolutions by decoding 1-4 token levels. As shown in Figure 11, DiT scales diffusion tokens and FLOPs with resolution, whereas DCS-LDM decouples output scale from complexity by varying only the token level. Fewer levels enable faster generation with baseline-comparable quality, while additional levels progressively improve quality. DCS-LDM therefore forms a better quality-efficiency frontier and outperforms the DiT baselines under comparable diffusion budgets (dashed lines). Table 4 further reports end-to-end throughput, latency, and peak memory. Because diffusion cost is controlled by token levels rather than resolution, DCS-LDM can generate higher-resolution images without a proportional latency increase. For example, with one token level, 256px and 512px generation have similar latency despite the doubled resolution. At 512px, DiT requires 6743ms, while DCS-LDM takes only 1943ms and 3671ms with one and two levels, respectively. These results show that DCS-LDM substantially reduces end-to-end latency while maintaining competitive generation quality. DCS-Tok incurs extra memory overhead since its 1D depatchifier tokenizes pixels independently. This can be reduced via patch-parallel decoding across GPUs or grouped depatchification over smaller patch batches.

Fig. 12: Content-adaptive token allocation. PSNR is measured under an average of four tokens per patch. Bluer patches have higher entropy and receive more tokens.

4.7 Content-adaptive Token Allocation

DCS-LDM captures the information inhomogeneity among samples by assigning different token levels to different instances. Moreover, the information distribution within a single sample is often non-uniform [48]. For instance, in the example shown in Figure 6b, the foreground hand contains most of the visual information, whereas the background remains relatively simple. This observation suggests that leveraging such inner-sample non-uniformity can further enhance the efficiency of visual representation. Since DCS-Tok allocates token levels independently for each patch during training (Section A.4), it naturally supports modeling this non-uniform distribution. As illustrated in Figure 12, we evaluate this property on an image reconstruction task. Specifically, we compute the entropy of each patch and allocate more tokens to patches with higher entropy. Under a fixed average token budget of four levels, uniform allocation achieves a PSNR of 31.59, whereas entropy-based allocation attains 32.42 on the validation set [11]. More details are provided in Section A.5. Further investigation is required to determine the optimal number of tokens during generation.

5 Conclusion

We explore the independence between information complexity and data scale in images and videos. To this end, we propose DCS-LDM, a latent diffusion model that uses variable-level latents to represent information complexity while enabling decoding at arbitrary scales. This design introduces flexible computation–quality tradeoffs into the generation process. In addition, DCS-LDM supports a coarse-to-fine progressive generation paradigm, which delivers results more efficiently with the same total computational cost and allows generation to terminate early as needed. Under this paradigm, DCS-LDM enables rapid semantic and structural previewing of generated samples, followed by fine-grained refinement. Extensive experiments demonstrate that DCS-LDM achieves performance comparable to state-of-the-art models while offering substantially greater flexibility and functionality.

References

1. Agarwal, N., Ali, A., Bala, M., Balaji, Y., Barker, E., Cai, T., Chattopadhyay, P., Chen, Y., Cui, Y., Ding, Y., et al.: Cosmos world foundation model platform for physical ai. arXiv preprint arXiv:2501.03575 (2025)
2. Bachmann, R., Allardice, J., Mizrahi, D., Fini, E., Kar, O.F., Amirloo, E., El-Nouby, A., Zamir, A., Dehghan, A.: Flextok: Resampling images into 1d token sequences of flexible length. In: Forty-second International Conference on Machine Learning (2025)
3. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., et al.: Qwen2. 5-vl technical report. arXiv preprint arXiv:2502.13923 (2025)
4. Berger, T.: Rate-distortion theory. Wiley Encyclopedia of Telecommunications (2003)
5. 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)
6. Chen, H., Han, Y., Chen, F., Li, X., Wang, Y., Wang, J., Wang, Z., Liu, Z., Zou, D., Raj, B.: Masked autoencoders are effective tokenizers for diffusion models. arXiv preprint arXiv:2502.03444 (2025)
7. Chen, H., Wang, Z., Li, X., Sun, X., Chen, F., Liu, J., Wang, J., Raj, B., Liu, Z., Barsoum, E.: Softvq-vae: Efficient 1-dimensional continuous tokenizer. arXiv preprint arXiv:2412.10958 (2024)
8. Chen, J., Cai, H., Chen, J., Xie, E., Yang, S., Tang, H., Li, M., Lu, Y., Han, S.: Deep compression autoencoder for efficient high-resolution diffusion models. arXiv preprint arXiv:2410.10733 (2024)
9. 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)
10. Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q.V., Salakhutdinov, R.: Transformer-xl: Attentive language models beyond a fixed-length context. arXiv preprint arXiv:1901.02860 (2019)
11. 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)
12. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
13. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. Communications of the ACM **63**(11), 139–144 (2020)
14. HaCohen, Y., Chiprut, N., Brazowski, B., Shalem, D., Moshe, D., Richardson, E., Levin, E., Shiran, G., Zabari, N., Gordon, O., et al.: Ltx-video: Realtime video latent diffusion. arXiv preprint arXiv:2501.00103 (2024)
15. 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)

16. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020)
17. Ho, J., Salimans, T.: Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598 (2022)
18. Hu, J., Zhong, T., Wang, X., Jiang, B., Tian, X., Yang, F., Wan, P., Zhang, D.: Vivid-10m: A dataset and baseline for versatile and interactive video local editing. arXiv preprint arXiv:2411.15260 (2024)
19. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14. pp. 694–711. Springer (2016)
20. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4015–4026 (2023)
21. 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)
22. Larsen, A.B.L., Sønderby, S.K., Larochelle, H., Winther, O.: Autoencoding beyond pixels using a learned similarity metric. In: International conference on machine learning. pp. 1558–1566. PMLR (2016)
23. Li, Z., Zhang, J., Lin, Q., Xiong, J., Long, Y., Deng, X., Zhang, Y., Liu, X., Huang, M., Xiao, Z., et al.: Hunyuan-dit: A powerful multi-resolution diffusion transformer with fine-grained chinese understanding. arXiv preprint arXiv:2405.08748 (2024)
24. Liu, Y., Qu, L., Zhang, H., Wang, X., Jiang, Y., Gao, Y., Ye, H., Li, X., Wang, S., Du, D.K., et al.: Detailflow: 1d coarse-to-fine autoregressive image generation via next-detail prediction. arXiv preprint arXiv:2505.21473 (2025)
25. Lu, G., Zhong, T., Geng, J., Hu, Q., Xu, D.: Learning based multi-modality image and video compression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6083–6092 (2022)
26. Ma, G., Huang, H., Yan, K., Chen, L., Duan, N., Yin, S., Wan, C., Ming, R., Song, X., Chen, X., et al.: Step-video-t2v technical report: The practice, challenges, and future of video foundation model. arXiv preprint arXiv:2502.10248 (2025)
27. 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)
28. Pan, Y., Feng, R., Dai, Q., Wang, Y., Lin, W., Guo, M., Luo, C., Zheng, N.: Semantics lead the way: Harmonizing semantic and texture modeling with asynchronous latent diffusion. arXiv preprint arXiv:2512.04926 (2025)
29. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4195–4205 (2023)
30. 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)
31. 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)
32. Shazeer, N.: Glu variants improve transformer. arXiv preprint arXiv:2002.05202 (2020)
33. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502 (2020)

34. Soomro, K., Zamir, A.R., Shah, M.: Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402 (2012)
35. Stergiou, A.: Lavib: A large-scale video interpolation benchmark. arXiv preprint arXiv:2406.09754 (2024)
36. 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)
37. Sun, X., Chen, Y., Huang, Y., Xie, R., Zhu, J., Zhang, K., Li, S., Yang, Z., Han, J., Shu, X., et al.: Hunyuan-large: An open-source moe model with 52 billion activated parameters by tencent. arXiv preprint arXiv:2411.02265 (2024)
38. Unterthiner, T., van Steenkiste, S., Kurach, K., Marinier, R., Michalski, M., Gelly, S.: Fvd: A new metric for video generation. In: DGS@ICLR (2019), https://api.semanticscholar.org/CorpusID:198489709
39. Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.W., Chen, D., Yu, F., Zhao, H., Yang, J., et al.: Wan: Open and advanced large-scale video generative models. arXiv preprint arXiv:2503.20314 (2025)
40. Wang, J., Jiang, Y., Yuan, Z., Peng, B., Wu, Z., Jiang, Y.G.: Omnitokenizer: A joint image-video tokenizer for visual generation. Advances in Neural Information Processing Systems **37**, 28281–28295 (2024)
41. Wang, Q., Shi, Y., Ou, J., Chen, R., Lin, K., Wang, J., Jiang, B., Yang, H., Zheng, M., Tao, X., et al.: Koala-36m: A large-scale video dataset improving consistency between fine-grained conditions and video content. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 8428–8437 (2025)
42. Wen, X., Zhao, B., Elezi, I., Deng, J., Qi, X.: " principal components" enable a new language of images. arXiv preprint arXiv:2503.08685 (2025)
43. 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)
44. Yao, J., Yang, B., Wang, X.: Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. arXiv preprint arXiv:2501.01423 (2025)
45. Yu, Q., Weber, M., Deng, X., Shen, X., Cremers, D., Chen, L.C.: An image is worth 32 tokens for reconstruction and generation. Advances in Neural Information Processing Systems **37**, 128940–128966 (2024)
46. 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)
47. Zhang, B., Sennrich, R.: Root mean square layer normalization. Advances in neural information processing systems **32** (2019)
48. Zhang, Z., Wu, R., Sun, L., Zhang, L.: Gpstoken: Gaussian parameterized spatially-adaptive tokenization for image representation and generation. arXiv preprint arXiv:2509.01109 (2025)
49. 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)
50. Zhong, T., Tian, X., Jiang, B., Wang, X., Tao, X., Wan, P., Zhang, Z.: Vfrtok: Variable frame rates video tokenizer with duration-proportional information assumption. arXiv preprint arXiv:2505.12053 (2025)