

Photorealistic Video Generation with Diffusion Models

Agrim Gupta^{1,2,*} Lijun Yu² Kihyuk Sohn² Xiuye Gu² Meera Hahn² Li Fei-Fei¹
Irfan Essa^{2,3} Lu Jiang² José Lezama²

¹ Stanford University ² Google Research ³ Georgia Institute of Technology



A cute cat riding a yellow surfboard, fast waves.



A cute corgi wearing a blue sweater, walking towards the camera, slow motion.



A raccoon wearing a black jacket, dancing in slow motion in front of the pyramids.

Figure 1. **W.A.L.T samples for text-to-video generation.** Our approach can generate high-resolution, temporally consistent photorealistic videos from text prompts. The samples shown are 512×896 resolution over 3.6 seconds duration at 8 frames per second.

Abstract

We present W.A.L.T, a transformer-based approach for photorealistic video generation via diffusion modeling. Our approach has two key design decisions. First, we use a causal encoder to jointly compress images and videos within a unified latent space, enabling training and generation across modalities. Second, for memory and training efficiency, we use a window attention architecture tailored for joint spatial and spatiotemporal generative modeling. Taken together these design decisions enable us to achieve state-of-the-art performance on established video (UCF-101 and Kinetics-600) and image (ImageNet) generation benchmarks without using classifier free guidance. Finally, we also train a cascade of three models for the task of text-to-video generation consisting of a base latent video diffusion model, and two video super-resolution diffusion models to generate

videos of 512×896 resolution at 8 frames per second.

1. Introduction

Transformers [73] are highly scalable and parallelizable neural network architectures designed to win the *hardware lottery* [39]. This desirable property has encouraged the research community to increasingly favor transformers over domain-specific architectures in diverse fields such as language [26, 55–57], audio [1], speech [58], vision [18, 30], and robotics [5, 7, 89]. Such a trend towards unification allows researchers to share and build upon advancements in traditionally disparate domains. Thus, leading to a virtuous cycle of innovation and improvement in model design favoring transformers.

*Work partially done during an internship at Google.

A notable exception to this trend is generative modelling of videos. Diffusion models [67, 69] have emerged as a leading paradigm for generative modelling of images [16, 33] and videos [36]. However, the U-Net architecture [33, 62], consisting of a series of convolutional [46] and self-attention [73] layers, has been the predominant backbone in all video diffusion approaches [16, 33, 36]. This preference stems from the fact that the memory demands of full attention mechanisms in transformers scale quadratically with input sequence length. Such scaling leads to prohibitively high costs when processing high-dimensional signals like video.

Latent diffusion models (LDMs) [61] reduce computational requirements by operating in a lower-dimensional latent space derived from an autoencoder [20, 72, 75]. A critical design choice in this context is the type of latent space employed: spatial compression (per frame latents) versus spatiotemporal compression. Spatial compression is often preferred because it enables leveraging pre-trained image autoencoders and LDMs, which are trained on large paired image-text datasets. However, this choice increases network complexity and limits the use of transformers as backbones, especially in generating high-resolution videos due to memory constraints. On the other hand, while spatiotemporal compression can mitigate these issues, it precludes the use of paired image-text datasets, which are much larger and diverse than their video counterparts.

We present **Window Attention Latent Transformer** (W.A.L.T): a transformer-based method for latent video diffusion models (LVDMs). Our method consists of two stages. First, an autoencoder maps both videos and images into a unified, lower-dimensional latent space. This design choice enables training a single generative model *jointly* on image and video datasets and significantly reduces the computational burden for generating high resolution videos. Subsequently, we propose a new design of transformer blocks for latent video diffusion modeling which is composed of self-attention layers that alternate between non-overlapping, window-restricted spatial and spatiotemporal attention. This design offers two primary benefits: firstly, the use of local window attention significantly lowers computational demands. Secondly, it facilitates joint training, where the spatial layers independently process images and video frames, while the spatiotemporal layers are dedicated to modeling the temporal relationships in videos.

While conceptually simple, our method provides the first empirical evidence of transformers' superior generation quality and parameter efficiency in latent video diffusion on public benchmarks. Specifically, we report state-of-the-art results on class-conditional video generation (UCF-101 [70]), frame prediction (Kinetics-600 [9]) and class conditional image generation (ImageNet [15]) without using classifier free guidance. Finally, to showcase the scalability and ef-

ficiency of our method we also demonstrate results on the challenging task of photorealistic text-to-video generation. We train a cascade of three models consisting of a base latent video diffusion model, and two video super-resolution diffusion models to generate videos of 512×896 resolution at 8 frames per second and report state-of-the-art zero-shot FVD score on the UCF-101 benchmark.

2. Related Work

Video Diffusion Models. Diffusion models have shown impressive results in image [33, 38, 52, 61, 67, 68] and video generation [4, 24, 29, 34, 36, 66]. Video diffusion models can be categorized into pixel-space [34, 36, 66] and latent-space [4, 24, 31, 83] approaches, the later bringing important efficiency advantages when modeling videos. Ho et al. [36] demonstrated that the quality of text conditioned video generation can be significantly improved by jointly training on image and video data. Similarly, to leverage image datasets, latent video diffusion models inflate a pre-trained image model, typically a U-Net [62], into a video model by adding temporal layers, and initializing them as the identity function [4, 34, 66]. Although computationally efficient, this approach couples the design of video and image models, and precludes spatiotemporal compression. In this work, we operate on a unified latent space for images and videos, allowing us to leverage large scale image and video datasets while enjoying computational efficiency gains from spatiotemporal compression of videos.

Transformers for Generative Modeling. Multiple classes of generative models have utilized Transformers [73] as backbone, such as, Generative adversarial networks [42, 47, 85], autoregressive [10, 11, 20, 21, 27, 59, 74, 77, 78, 80, 81] and diffusion [2, 22, 41, 50, 53, 87] models. Inspired by the success of autoregressive pretraining of large language models [55–57], Ramesh et al. [59] trained a text-to-image generation model by predicting the next visual token obtained from an image tokenizer. Subsequently, this approach was applied to multiple applications including class-conditional image generation [20, 79], text-to-image [17, 59, 76, 80] or image-to-image translation [21, 77]. Similarly, for video generation, transformer-based models were proposed to predict next tokens using 3D extensions of VQGAN [23, 37, 78, 81] or using per frame image latents [27]. Autoregressive sampling of videos is typically impractical given the very long sequences involved. To alleviate this issue, non-autoregressive sampling [10, 11], *i.e.* parallel token prediction, has been adopted as a more efficient solution for transformer-based video generation [27, 74, 81]. Recently, the community has started adopting transformers as the denoising backbone for diffusion models in place of U-Net [12, 38, 50, 53, 87]. To the best of our knowledge, our work is the first successful

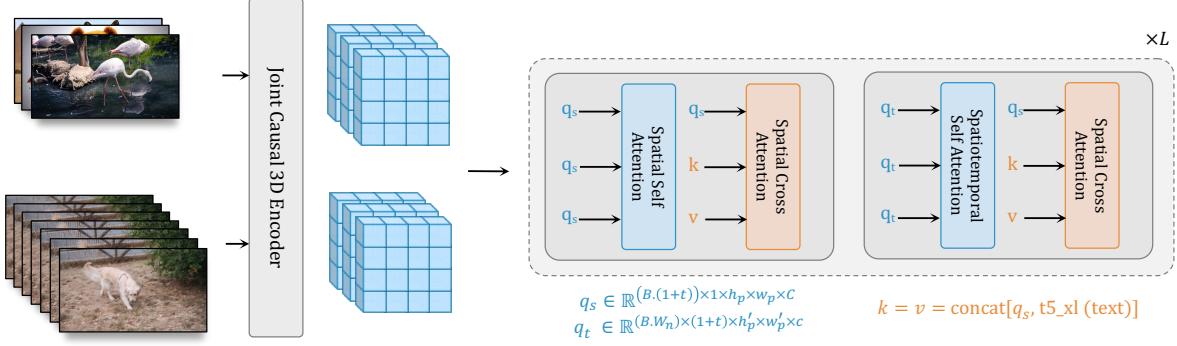


Figure 2. **W.A.L.T.** We encode images and videos into a shared latent space. The transformer backbone processes these latents with blocks having two layers of window-restricted attention: spatial layers capture spatial relations in both images and video, while spatiotemporal layers model temporal dynamics in videos and *passthrough* images via identity attention mask. Text conditioning is done via spatial cross-attention.

empirical demonstration (§ 5.1) of a transformer-based backbone for jointly training image and video latent diffusion models.

3. Background

Diffusion formulation. Diffusion models [33, 67, 69] are a class of generative models which learn to generate data by iteratively denoising samples drawn from a noise distribution. Gaussian diffusion models assume a forward noising process which gradually applies noise (ϵ) to real data ($x_0 \sim p_{\text{data}}$). Concretely,

$$x_t = \sqrt{\gamma(t)} x_0 + \sqrt{1 - \gamma(t)} \epsilon, \quad (1)$$

where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $t \in [0, 1]$, and $\gamma(t)$ is a monotonically decreasing function (noise schedule) from 1 to 0. Diffusion models are trained to learn the reverse process that inverts the forward corruptions:

$$\mathbb{E}_{x \sim p_{\text{data}}, t \sim \mathcal{U}(0, 1), \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} [\|\mathbf{y} - f_\theta(\mathbf{x}_t; \mathbf{c}, t)\|^2], \quad (2)$$

where f_θ is the denoiser model parameterized by a neural network, \mathbf{c} is conditioning information e.g., class labels or text prompts, and the target \mathbf{y} can be random noise ϵ , denoised input \mathbf{x}_0 or $\mathbf{v} = \sqrt{1 - \gamma(t)} \epsilon - \sqrt{\gamma(t)} \mathbf{x}_0$. Following [34, 63], we use \mathbf{v} -prediction in all our experiments.

Latent diffusion models (LDMs). Processing high-resolution images and videos using raw pixels requires considerable computational resources. To address this, LDMs operate on the low dimensional latent space of a VQ-VAE [20, 72]. VQ-VAE consists of an encoder $E(x)$ that encodes an input video $x \in \mathbb{R}^{T \times H \times W \times 3}$ into a latent representation $z \in \mathbb{R}^{t \times h \times w \times c}$. The encoder downsamples the video by a factor of $f_s = H/h = W/w$ and $f_t = T/t$, where $T = t = 1$ corresponds to using an image autoencoder. An important distinction from the original VQ-VAE is the absence of a codebook of quantized embeddings as

diffusion models can operate on continuous latent spaces. A decoder D is trained to predict a reconstruction of the video, \hat{x} , from z . Following VQ-GAN [20], reconstruction quality can be further improved by adding adversarial [25] and perceptual losses [43, 86].

4. W.A.L.T

4.1. Learning Visual Tokens

A key design decision in video generative modeling is the choice of latent space representation. Ideally, we want a shared and unified compressed visual representation that can be used for generative modeling of both images and videos [74, 82]. The unified representation is important because joint image-video learning is preferable due to a scarcity of labeled video data [34], such as text-video pairs. Concretely, given a video sequence $x \in \mathbb{R}^{(1+T) \times H \times W \times C}$, we aim to learn a low-dimensional representation $z \in \mathbb{R}^{(1+t) \times h \times w \times c}$ that performs spatial-temporal compression by a factor of $f_s = H/h = W/w$ in space and a factor of $f_t = T/t$ in time. To enable a unified representation for both videos and static images, the first frame is always encoded independently from the rest of the video. This allows static images $x \in \mathbb{R}^{1 \times H \times W \times C}$ to be treated as videos with a single frame, i.e. $z \in \mathbb{R}^{1 \times h \times w \times c}$.

We instantiate this design with the causal 3D CNN encoder-decoder architecture of the MAGVIT-v2 tokenizer [82]. Typically the encoder-decoder consists of regular 3D convolution layers which cannot process the first frame independently [23, 81]. This limitation stems from the fact that a regular convolutional kernel of size (k_t, k_h, k_w) will operate on $\lfloor \frac{k_t-1}{2} \rfloor$ frames before and $\lfloor \frac{k_t}{2} \rfloor$ frames after the input frames. *Causal* 3D convolution layers solve this issue as the convolutional kernel operates on only the past $k_t - 1$ frames. This ensures that the output for each frame is influenced solely by the preceding frames, enabling the model to tokenize the first frame independently.

After this stage, the input to our model is a batch of latent tensors $z \in \mathbb{R}^{(1+t) \times h \times w \times c}$ representing a single video or a stack of $1 + t$ independent images (Fig. 2). Different from [82], our latent representation is real-valued and quantization-free. In the section below we describe how our model jointly processes a mixed batch of images and videos.

4.2. Learning to Generate Images and Videos

Patchify. Following the original ViT [18], we “patchify” each latent frame independently by converting it into a sequence of non-overlapping $h_p \times w_p$ patches where $h_p = h/p$, $w_p = w/p$ and p is the patch size. We use learnable positional embeddings [73], which are the sum of space and time positional embeddings. Position embeddings are added to the linear projections [18] of the patches. Note that for images, we simply add the temporal position embedding corresponding to the first latent frame.

Window attention. Transformer models composed entirely of global self-attention modules incur significant compute and memory costs, especially for video tasks. For efficiency and for processing images and videos jointly we compute self-attention in windows [27, 73], based on two types of non-overlapping configurations: spatial (S) and spatiotemporal (ST), cf. Fig. 2. *Spatial Window (SW)* attention is restricted to all the tokens within a latent frame of size $1 \times h_p \times w_p$ (the first dimension is time). SW models the spatial relations in images and videos. *Spatiotemporal Window (STW)* attention is restricted within a 3D window of size $(1+t) \times h'_p \times h'_w$, modeling the temporal relationships among video latent frames. For images, we simply use an *identity* attention mask ensuring that the *value* embeddings corresponding to the image frame latents are passed through the layer as is. Finally, in addition to absolute position embeddings we also use relative position embeddings [49].

Our design, while conceptually straightforward, achieves computational efficiency and enables joint training on image and video datasets. In contrast to methods based on frame-level autoencoders [4, 24, 27], our approach does not suffer from flickering artifacts, which often result from encoding and decoding video frames independently. However, similar to Blattmann et al. [4], we can also potentially leverage pre-trained image LDMs with transformer backbones by simply interleaving STW layers.

4.3. Conditional Generation

To enable controllable video generation, in addition to conditioning on timestep t , diffusion models are often conditioned on additional conditional information c such as class labels, natural language, past frames or low resolution videos. In our transformer backbone, we incorporate three types of conditioning mechanisms as described in what follows:

Cross-attention. In addition to self-attention layers in our window transformer blocks, we add a cross-attention

layer for text conditioned generation. When training models on just videos, the cross-attention layer employs the same window-restricted attention as the self-attention layer, meaning S/ST blocks will have SW/STW cross-attention layers (Fig. 2). However, for joint training, we only use SW cross-attention layers. For cross-attention we concatenate the input signal (query) with the conditioning signal (key, value) as our early experiments showed this improves performance.

AdaLN-LoRA. Adaptive normalization layers are an important component in a broad range of generative and visual synthesis models [16, 19, 44, 52–54]. A simple way to incorporate adaptive layer normalization is to include for each layer i , an MLP layer to regress a vector of conditioning parameters $A^i = \text{MLP}(c + t)$, where $A^i = \text{concat}(\gamma_1, \gamma_2, \beta_1, \beta_2, \alpha_1, \alpha_2)$, $A^i \in \mathbb{R}^{6 \times d_{\text{model}}}$, and $c \in \mathbb{R}^{d_{\text{model}}}$, $t \in \mathbb{R}^{d_{\text{model}}}$ are the condition and timestep embeddings. In the transformer block, γ and β scale and shift the inputs of the multi-head attention and MLP layers, respectively, while α scales the output of both the multi-head attention and MLP layers. The parameter count of these additional MLP layers scales linearly with the number of layers and quadratically with the model’s dimensional size ($\text{num_blocks} \times d_{\text{model}} \times 6 \times d_{\text{model}}$). For instance, in a ViT-g model with 1B parameters, the MLP layers contribute an additional 475M parameters. Inspired by [40], we propose a simple solution dubbed *AdaLN-LoRA*, to reduce the model parameters. For each layer, we regress conditioning parameters as

$$A^1 = \text{MLP}(c + t), \quad A^i = A^1 + W_b^i W_a^i(c + t) \quad \forall i \neq 1, \quad (3)$$

where $W_b^i \in \mathbb{R}^{d_{\text{model}} \times r}$, $W_a^i \in \mathbb{R}^{r \times (6 \times d_{\text{model}})}$. This reduces the number of trainable model parameters significantly when $r \ll d_{\text{model}}$. For example, a ViT-g model with $r = 2$ reduces the MLP parameters from 475M to 12M.

Self-conditioning. In addition to being conditioned on external inputs, iterative generative algorithms can also be conditioned on their own previously generated samples during inference [3, 13, 65]. Specifically, Chen et al. [13] modify the training process for diffusion models, such that with some probability p_{sc} the model first generates a sample $\tilde{z}_0 = f_\theta(z_t; \mathbf{0}, c, t)$ and then refines this estimate using another forward pass conditioned on this initial sample: $f_\theta(z_t; \text{stopgrad}(\tilde{z}_0), c, t)$. With probability $1 - p_{\text{sc}}$, only a single forward pass is done. We concatenate the model estimate with the input along the channel dimension and found this simple technique to work well when used in conjunction with *v-prediction*.

4.4. Autoregressive Generation

For generating long videos via autoregressive prediction we also train our model *jointly* on the task of *frame prediction*. This is achieved by conditioning the model on past frames with a probability of p_{fp} during training. Specifically, the

Method	K600 FVD↓	UCF FVD↓	params.	steps
TrIVD-GAN-FP [51]	25.7 _{±0.7}	—	—	1
Video Diffusion [36]	16.2 _{±0.3}	—	1.1B	256
RIN [41]	10.8	—	411M	1000
TATS [23]	—	332 _{±18}	321M	1024
Phenaki [74]	36.4 _{±0.2}	—	227M	48
MAGVIT [81]	9.9 _{±0.3}	76 _{±2}	306M	12
MAGVITv2 [82]	4.3 _{±0.1}	58 _{±2}	307M	24
W.A.L.T-L <i>Ours</i>	3.3 _{±0.0}	46 _{±2}	313M	50
W.A.L.T-XL <i>Ours</i>	—	36 _{±2}	460M	50

Table 1. **Video generation** evaluation on frame prediction on Kinetics-600 and class-conditional generation on UCF-101.

model is conditioned using $c_{fp} = \text{concat}(m_{fp} \circ z_t, m_{fp})$, where m_{fp} is a binary mask. The binary mask indicates the number of past frames used for conditioning. We condition on either 1 latent frame (image to video generation) or 2 latent frames (video prediction). This conditioning is integrated into the model through concatenation along the channel dimension of the noisy latent input. During inference, we use standard classifier-free guidance with c_{fp} as the conditioning signal.

4.5. Video Super Resolution

Generating high-resolution videos with a single model is computationally prohibitive. Following [35], we use a cascaded approach with three models operating at increasing resolutions. Our base model generates videos at 128×128 resolution which are subsequently upsampled twice via two super resolution stages. We first spatially upscale the low resolution input z^{lr} (video or image) using a depth-to-space convolution operation. Note that, unlike training where ground truth low-resolution inputs are available, inference relies on latents produced by preceding stages (*cf.* teaching-forcing). To reduce this discrepancy and improve the robustness of the super-resolution stages in handling artifacts generated by lower resolution stages, we use noise conditioning augmentation [35]. Concretely, noise is added in accordance with $\gamma(t)$, by sampling a noise level as $t_{sr} \sim \mathcal{U}(0, t_{\max_noise})$ and is provided as input to our *AdaLN-LoRA* layers.

Aspect-ratio finetuning. To simplify training and leverage broad data sources with different aspect ratios, we train our base stage using a square aspect ratio. We fine-tune the base stage on a subset of data to generate videos with a 9 : 16 aspect ratio by interpolating position embeddings.

5. Experiments

In this section, we evaluate our method on multiple tasks: class-conditional image and video generation, frame prediction and text conditioned video generation and perform extensive ablation studies of different design choices. For

Method	Cost (Iter×BS)	FID↓	IS↑	params.	steps
BigGAN-deep [6]	-	6.95	171.4	160M	1
LDM-4 [61]	178k×1200	10.56	103.5	400M	250
DiT-XL/2 [53]	7000k×256	9.62	121.5	675M	250
ADM [16]	-	7.49	127.5	608M	2000
MDT [22]	6500k×256	6.23	143.0	676M	250
MaskDiT [87]	1200k×1024	5.69	178.0	736M	40
RIN [41]	600k×1024	3.42	182.0	410M	1000
simple diffusion [38]	500K×2048	2.77	211.8	2B	512
VDM++ [45]	-	2.40	225.3	2B	512
W.A.L.T-L <i>Ours</i>	437k×1024	2.56	215.1	460M	50

Table 2. **Class-conditional image generation on ImageNet 256×256.** We adopt the evaluation protocol and implementation of ADM [16] and report results without classifier free guidance.

qualitative results, see Fig. 1, Fig. 3, Fig. 4 and videos on our project website. See appendix for additional details.

5.1. Visual Generation

Video generation. We consider two standard video benchmarks, UCF-101 [70] for class-conditional generation and Kinetics-600 [9] for video prediction with 5 conditioning frames. We use FVD [71] as our primary evaluation metric. Across both datasets, W.A.L.T significantly outperforms all prior works (Tab. 1). Compared to prior video diffusion models, we achieve state-of-the-art performance with less model parameters, and require 50 DDIM [68] inference steps.

Image generation. To verify the modeling capabilities of W.A.L.T on the image domain, we train a version of W.A.L.T for the standard ImageNet class-conditional setting. For evaluation, we follow ADM [16] and report the FID [32] and Inception [64] scores calculated on 50K samples generated in 50 DDIM steps. We compare (Table 2) W.A.L.T with state-of-the-art image generation methods for 256×256 resolution. Our model outperforms prior works without requiring specialized schedules, convolution inductive bias, improved diffusion losses, and classifier free guidance. Although VDM++ [45] has slightly better FID score, the model has significantly more parameters (2B).

5.2. Ablation Studies

We ablate W.A.L.T to understand the contribution of various design decisions with the default settings: model L, patch size 1, $1 \times 16 \times 16$ spatial window, $5 \times 8 \times 8$ spatiotemporal window, $p_{sc} = 0.9$, $c = 8$ and $r = 2$.

Patch size. In various computer vision tasks utilizing ViT[18]-based models, a smaller patch size p has been shown to consistently enhance performance [8, 18, 28, 84]. Similarly, our findings also indicate that a reduced patch size improves performance (Table 3a).

Window attention. We compare three different STW window configurations with full self-attention (Table 3b). We find that local self-attention can achieve competitive (or bet-

patch size p	FVD \downarrow	IS \uparrow	st window	FVD \downarrow	IS \uparrow	sps	p_{sc}	FVD \downarrow	IS \uparrow
			$5 \times 4 \times 4$	56.9	87.3	2.24			
1	60.7	87.2	$5 \times 8 \times 8$	59.6	87.4	2.00	0.0	109.9	82.6
2	134.4	82.2	$5 \times 16 \times 16$	55.3	87.4	1.75	0.3	76.0	86.5
4	461.8	63.9	full self attn.	59.9	87.8	1.20	0.6	60.0	86.8
							0.9	61.4	87.1

(a) **Patch size.** Lower patch size is significantly better.

r	FVD \downarrow	IS \uparrow	params
2	60.7	87.2	313 M
4	56.6	87.3	314 M
16	55.5	88.0	316 M
64	54.4	87.9	324 M
256	52.5	88.5	357 M

(d) **AdaLN-LoRA.** Bigger r is better.

(b) **Spatiotemporal window size.** Full self-attention is not essential for good performance. sps is steps per sec.

	FVD \downarrow	IS \uparrow
w/o qk norm [14]	59.0	86.8
w/o latent norm	67.9	87.1
w/o zero snr [48]	91.0	84.2
full method	60.7	87.2

(e) **Other improvements.** See text for details.

(c) **Self-conditioning.** Higher p_{sc} is better.

c	rFVD \downarrow	FVD \downarrow	IS \uparrow
4	37.7	86.4	84.9
8	17.1	75.4	86.3
16	8.2	67.0	86.0
32	3.5	83.4	82.9

(f) **Latent dimension** c . Higher c is better.

Table 3. **Ablation experiments** on UCF-101 [70]. We compare FVD and inception scores to ablate important design decisions with the default setting: L model, $1 \times 16 \times 16$ spatial window, $5 \times 8 \times 8$ spatiotemporal (st) window, $p_{sc} = 0.9$, $c = 8$ and $r = 2$.

Model	AdaLN	FVD \downarrow	IS \uparrow	params.	final loss
L	separate	34.6	90.2	458M	0.274
XL	LoRA-2	36.7	89.4	460M	0.268

Table 4. **Parameter matched** comparison between AdaLN-LoRA and per layer adaln layers. See text for details.

ter) performance while being significantly faster (up to $2\times$) and requiring less accelerator memory.

Self-conditioning. In Table 3c we study the influence of varying the self-conditioning rate p_{sc} on generation quality. We notice a clear trend: increasing the self conditioning rate from 0.0 (no self-conditioning) to 0.9 improves the FVD score substantially (44%).

AdaLN-LoRA. An important design decision in diffusion models is the conditioning mechanism. We investigate the effect of increasing the bottleneck dimension r in our proposed AdaLN-LoRA layers (Table 3d). This hyperparameter provides a flexible way to trade off between number of model parameters and generation performance. As shown in Table 3d, increasing r improves performance but also increases model parameters. This highlights an important model design question: given a fixed parameter budget, how should we allocate parameters - either by using *separate* AdaLN layers, or by increasing base model parameters while using *shared* AdaLN-LoRA layers? We explore this in Table 4 by comparing two model configurations: W.A.L.T-L with separate AdaLN layers and W.A.L.T-XL with AdaLN-LoRA and $r = 2$. While both configurations yield similar FVD and Inception scores, W.A.L.T-XL achieves a lower final loss value, suggesting the advantage of allocating more parameters to the base model and choosing an appropriate r value within accelerator memory limits.

Noise schedule. Common latent diffusion noise schedules [61] typically do not ensure a zero signal-to-noise ra-

tio (SNR) at the final timestep, i.e., at $t = 1, \gamma(t) > 0$. This leads to a mismatch between training and inference phases. During inference, models are expected to start from purely Gaussian noise, whereas during training, at $t = 1$, a small amount of signal information remains accessible to the model. This is especially harmful for video generation as videos have high temporal redundancy. Even minimal information leakage at $t = 1$ can reveal substantial information to the model. Addressing this mismatch by enforcing a zero terminal SNR [48] significantly improves performance (Table 3e). Note that this approach was originally proposed to fix over-exposure problems in image generation, but we find it effective for video generation as well.

Autoencoder. Finally, we investigate one critical but often overlooked hyperparameter in the first stage of our model: the channel dimension c of the autoencoder latent z . As shown in Table 3f, increasing c significantly improves the reconstruction quality (lower rFVD) while keeping the same spatial f_s and temporal compression f_t ratios. Empirically, we found that both lower and higher values of c lead to poor FVD scores in generation, with a sweet spot of $c = 8$ working well across most datasets and tasks we evaluated. We also normalize the latents before processing them via transformer which further improves performance.

In our transformer models, we use query-key normalization [14] as it helps stabilize training for larger models. Finally, we note that some of our default settings are not optimal, as indicated by ablation studies. These defaults were chosen early on for their robustness across datasets, though further tuning may improve performance.

5.3. Text-to-video

We train W.A.L.T for text-to-video jointly on text-image and text-video pairs (Sec. 4.2). We used a dataset of ~ 970 M



A polar bear swimming.



Pouring beer into an empty glass, low angle shot, bar in the background.



A cat eating food out of a bowl, in the style of Van Gogh.



Pouring chocolate sauce over vanilla ice cream in a cone, studio lighting.



A robot ballerina dancing gracefully, highly detailed, studio lighting.



A cute panda skateboarding in the sky, over snow covered mountains, with dreamy whimsical atmosphere.



An astronaut riding a horse.

Figure 3. **Qualitative evaluation.** Example videos generated by W.A.L.T from natural language prompts at 512×896 resolution over 3.6 seconds duration at 8 frames per second. The W.A.L.T model is able to generate temporally consistent photorealistic videos that align with the textual prompt.

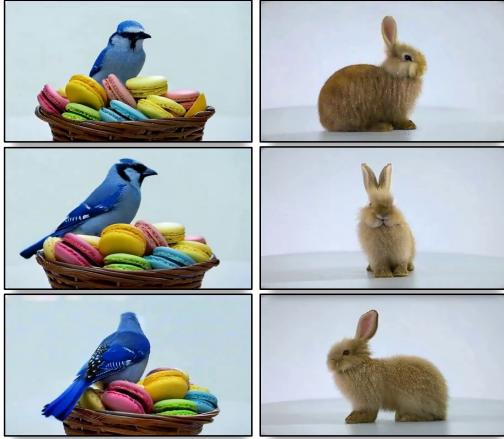


Figure 4. **Examples of consistent 3D camera motion (5.1 secs).** Prompts: *camera turns around a {blue jay, bunny}, studio lighting, 360° rotation.* Best viewed in video format.

text-image pairs and $\sim 89M$ text-video pairs from the public internet and internal sources. We train our base model at resolution $17 \times 128 \times 128$ (3B parameters), and two $2\times$ cascaded super-resolution models for $17 \times 128 \times 224 \rightarrow 17 \times 256 \times 448$ (L, 1.3B, $p = 2$) and $17 \times 256 \times 448 \rightarrow 17 \times 512 \times 896$ (L, 419M, $p = 2$) respectively. We fine-tune the base stage for the 9 : 16 aspect ratio to generate videos at resolution 128×224 . We use classifier free guidance for all our text-to-video results.

5.3.1 Quantitative Evaluation

Evaluating text-conditioned video generation systems scientifically remains a significant challenge, in part due to the absence of standardized training datasets and benchmarks. So far we have focused our experiments and analyses on the standard academic benchmarks, which use the same training data to ensure controlled and fair comparisons. Nevertheless, to compare with prior work on text-to-video, we also report results on the UCF-101 dataset in the zero-shot evaluation protocol in Table 5 [24, 37, 66]. Also see supplement.

Joint training. A primary strength of our framework is its ability to train simultaneously on both image and video datasets. In Table 5 we ablate the impact of this joint training approach. Specifically, we trained two versions of W.A.L.T-L (each with 419M params.) models using the default settings specified in § 5.2. We find that joint training leads to a notable improvement across both metrics. Our results align with the findings of Ho et al. [36], who demonstrated the benefits of joint training for pixel-based video diffusion models with U-Net backbones.

Scaling. Transformers are known for their ability to scale effectively in many tasks [5, 14, 55]. In Table 5 we show the benefits of scaling our transformer model for video diffusion.

Method	IS (\uparrow)	FVD (\downarrow)
Cog Video (Chinese) [37]	23.6	751.3
Cog Video (English) [37]	25.3	701.6
MagicVideo [88]	-	699.0
Make-A-Video [66]	33.0	367.2
Video LDM [4]	33.5	550.6
PYoCo [24]	47.8	355.2
W.A.L.T (<i>Ours</i>) 419M (video only)	26.8	598.8
W.A.L.T (<i>Ours</i>) 419M (video + image)	31.7	344.5
W.A.L.T (<i>Ours</i>) 3B (video + image)	35.1	258.1

Table 5. **UCF-101 text-to-video generation.** Joint training on image and video datasets in conjunction with scaling the model parameters is essential for high quality video generation.

Scaling our base model size leads to significant improvements on both the metrics. It is important to note, however, that our base model is considerably smaller than leading text-to-video systems. For instance, Ho et al. [34] trained base model of 5.7B parameters. Hence, we believe scaling our models further is an important direction of future work.

Comparison with prior work. In Table 5, we present a system-level comparison of various text-to-video generation methods. Our results are promising; we surpass all previous work in the FVD metric. In terms of the IS, our performance is competitive, outperforming all but PYoCo [24]. A possible explanation for this discrepancy might be PYoCo’s use of stronger text embeddings. Specifically, they utilize both CLIP [57] and T5-XXL [60] encoders, whereas we employ a T5-XL [60] text encoder only.

5.3.2 Qualitative Results

As mentioned in § 4.4, we jointly train our model on the task of frame prediction conditioned on 1 or 2 latent frames. Hence, our model can be used for animating images (**image-to-video**) and generating longer videos with consistent camera motion (Fig. 4). See videos on our [project website](#).

6. Conclusion

In this work, we introduce W.A.L.T, a simple, scalable, and efficient transformer-based framework for latent video diffusion models. We demonstrate state-of-the-art results for image and video generation using a transformer backbone with windowed attention. We also train a cascade of three W.A.L.T models jointly on image and video datasets, to synthesize high-resolution, temporally consistent photorealistic videos from natural language descriptions. While generative modeling has seen tremendous recent advances for images, progress on video generation has lagged behind. We hope that scaling our unified framework for image and video generation will help close this gap.

Acknowledgements

We thank Bryan Seybold, Dan Kondratyuk, David Ross, Hartwig Adam, Huisheng Wang, Jason Baldridge, Mauricio Delbracio and Orly Liba for helpful discussions and feedback.

References

- [1] Andrea Agostinelli, Timo I Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, et al. MusicLM: Generating music from text. *arXiv:2301.11325*, 2023. 1
- [2] Fan Bao, Chongxuan Li, Yue Cao, and Jun Zhu. All are worth words: a vit backbone for score-based diffusion models. In *NeurIPS 2022 Workshop on Score-Based Methods*, 2022. 2
- [3] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in neural information processing systems*, 28, 2015. 4
- [4] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *CVPR*, 2023. 2, 4, 8
- [5] Konstantinos Bousmalis, Giulia Vezzani, Dushyant Rao, Coline Devin, Alex X Lee, Maria Bauza, Todor Davchev, Yuxiang Zhou, Agrim Gupta, Akhil Raju, et al. Robocat: A self-improving foundation agent for robotic manipulation. *arXiv preprint arXiv:2306.11706*, 2023. 1, 8
- [6] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *ICLR*, 2018. 5
- [7] Anthony Brohan, Noah Brown, Justice Carbalal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022. 1
- [8] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 5
- [9] Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about Kinetics-600. *arXiv:1808.01340*, 2018. 2, 5
- [10] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. MaskGIT: Masked generative image transformer. In *CVPR*, 2022. 2
- [11] Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, José Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy, William T Freeman, Michael Rubinstein, et al. Muse: Text-to-image generation via masked generative transformers. In *ICML*, 2023. 2
- [12] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, et al. Pixart-\alpha: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*, 2023. 2
- [13] Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. *arXiv preprint arXiv:2208.04202*, 2022. 4
- [14] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning*, pages 7480–7512. PMLR, 2023. 6, 8
- [15] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 2
- [16] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *NeurIPS*, 2021. 2, 4, 5, 1
- [17] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, et al. CogView: Mastering text-to-image generation via transformers. In *NeurIPS*, 2021. 2
- [18] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020. 1, 4, 5
- [19] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. *arXiv preprint arXiv:1610.07629*, 2016. 4
- [20] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, 2021. 2, 3
- [21] Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. Make-a-scene: Scene-based text-to-image generation with human priors. In *European Conference on Computer Vision*, pages 89–106. Springer, 2022. 2
- [22] Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. Masked diffusion transformer is a strong image synthesizer. *arXiv:2303.14389*, 2023. 2, 5, 1
- [23] Songwei Ge, Thomas Hayes, Harry Yang, Xi Yin, Guan Pang, David Jacobs, Jia-Bin Huang, and Devi Parikh. Long video generation with time-agnostic VQGAN and time-sensitive transformer. In *ECCV*, 2022. 2, 3, 5
- [24] Songwei Ge, Seungjun Nah, Guilin Liu, Tyler Poon, Andrew Tao, Bryan Catanzaro, David Jacobs, Jia-Bin Huang, Ming-Yu Liu, and Yogesh Balaji. Preserve your own correlation: A noise prior for video diffusion models. *arXiv preprint arXiv:2305.10474*, 2023. 2, 4, 8, 1
- [25] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 3
- [26] Google. PaLM 2 technical report. *arXiv:2305.10403*, 2023. 1
- [27] Agrim Gupta, Stephen Tian, Yunzhi Zhang, Jiajun Wu, Roberto Martín-Martín, and Li Fei-Fei. MaskViT: Masked visual pre-training for video prediction. In *ICLR*, 2022. 2, 4

- [28] Agrim Gupta, Jiajun Wu, Jia Deng, and Li Fei-Fei. Siamese masked autoencoders. *arXiv preprint arXiv:2305.14344*, 2023. 5
- [29] William Harvey, Saeid Naderiparizi, Vaden Masrani, Christian Weilbach, and Frank Wood. Flexible diffusion modeling of long videos. *Advances in Neural Information Processing Systems*, 35:27953–27965, 2022. 2
- [30] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 1
- [31] Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. Latent video diffusion models for high-fidelity long video generation. *arXiv preprint arXiv:2211.13221*, 2023. 2
- [32] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017. 5
- [33] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 2, 3
- [34] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv:2210.02303*, 2022. 2, 3, 8
- [35] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *JMLR*, 23(1): 2249–2281, 2022. 5
- [36] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. In *ICLR Workshops*, 2022. 2, 5, 8
- [37] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. CogVideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv:2205.15868*, 2022. 2, 8
- [38] Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In *ICML*, 2023. 2, 5, 1
- [39] Sara Hooker. The hardware lottery. *Communications of the ACM*, 64(12):58–65, 2021. 1
- [40] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *ICLR*, 2021. 4
- [41] Allan Jabri, David J Fleet, and Ting Chen. Scalable adaptive computation for iterative generation. In *ICML*, 2023. 2, 5
- [42] Yifan Jiang, Shiyu Chang, and Zhangyang Wang. Transgan: Two pure transformers can make one strong gan, and that can scale up. *Advances in Neural Information Processing Systems*, 34:14745–14758, 2021. 2
- [43] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016. 3
- [44] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 4
- [45] Diederik P Kingma and Ruiqi Gao. Understanding the diffusion objective as a weighted integral of elbos. *arXiv:2303.00848*, 2023. 5, 1
- [46] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 2
- [47] Kwonjoon Lee, Huiwen Chang, Lu Jiang, Han Zhang, Zhuowen Tu, and Ce Liu. Vitgan: Training gans with vision transformers. *arXiv preprint arXiv:2107.04589*, 2021. 2
- [48] Shanchuan Lin, Bingchen Liu, Jiashi Li, and Xiao Yang. Common diffusion noise schedules and sample steps are flawed. *arXiv preprint arXiv:2305.08891*, 2023. 6
- [49] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 4
- [50] Haoyu Lu, Guoxing Yang, Nanyi Fei, Yuqi Huo, Zhiwu Lu, Ping Luo, and Mingyu Ding. Vdt: General-purpose video diffusion transformers via mask modeling. *arXiv preprint arXiv:2305.13311*, 2023. 2
- [51] Pauline Luc, Aidan Clark, Sander Dieleman, Diego de Las Casas, Yotam Doron, Albin Cassirer, and Karen Simonyan. Transformation-based adversarial video prediction on large-scale data. *arXiv:2003.04035*, 2020. 5
- [52] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021. 2, 4
- [53] William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv:2212.09748*, 2022. 2, 5, 1
- [54] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, 2018. 4
- [55] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018. 1, 2, 8
- [56] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
- [57] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 1, 2, 8
- [58] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*, pages 28492–28518. PMLR, 2023. 1
- [59] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, 2021. 2

- [60] Adam Roberts, Colin Raffel, Katherine Lee, Michael Matena, Noam Shazeer, Peter J Liu, Sharan Narang, Wei Li, and Yanqi Zhou. Exploring the limits of transfer learning with a unified text-to-text transformer. 2019. 8
- [61] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 2, 5, 6, 1
- [62] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 2
- [63] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022. 3
- [64] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *NeurIPS*, 2016. 5
- [65] Nikolay Savinov, Junyoung Chung, Mikolaj Binkowski, Erich Elsen, and Aaron van den Oord. Step-unrolled denoising autoencoders for text generation. *arXiv preprint arXiv:2112.06749*, 2021. 4
- [66] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv:2209.14792*, 2022. 2, 8
- [67] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015. 2, 3
- [68] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2, 5
- [69] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *NeurIPS*, 2019. 2, 3
- [70] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv:1212.0402*, 2012. 2, 5, 6
- [71] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv:1812.01717*, 2018. 5
- [72] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. In *NeurIPS*, 2017. 2, 3
- [73] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 1, 2, 4
- [74] Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. Phenaki: Variable length video generation from open domain textual description. *arXiv:2210.02399*, 2022. 2, 3, 5
- [75] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008. 2
- [76] Chenfei Wu, Lun Huang, Qianxi Zhang, Binyang Li, Lei Ji, Fan Yang, Guillermo Sapiro, and Nan Duan. Godiva: Generating open-domain videos from natural descriptions. *arXiv preprint arXiv:2104.14806*, 2021. 2
- [77] Chenfei Wu, Jian Liang, Lei Ji, Fan Yang, Yuejian Fang, Daxin Jiang, and Nan Duan. Nüwa: Visual synthesis pre-training for neural visual world creation. In *European conference on computer vision*, pages 720–736. Springer, 2022. 2
- [78] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. VideoGPT: Video generation using VQ-VAE and transformers. *arXiv:2104.10157*, 2021. 2
- [79] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved VQGAN. In *ICLR*, 2022. 2
- [80] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv:2206.10789*, 2022. 2
- [81] Lijun Yu, Yong Cheng, Kihyuk Sohn, José Lezama, Han Zhang, Huiwen Chang, Alexander G Hauptmann, Ming-Hsuan Yang, Yuan Hao, Irfan Essa, et al. MAGVIT: Masked generative video transformer. In *CVPR*, 2023. 2, 3, 5
- [82] Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Agrim Gupta, Xiuye Gu, Alexander G Hauptmann, et al. Language model beats diffusion–tokenizer is key to visual generation. *arXiv preprint arXiv:2310.05737*, 2023. 3, 4, 5, 1
- [83] Sihyun Yu, Kihyuk Sohn, Subin Kim, and Jinwoo Shin. Video probabilistic diffusion models in projected latent space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18456–18466, 2023. 2
- [84] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12104–12113, 2022. 5
- [85] Bowen Zhang, Shuyang Gu, Bo Zhang, Jianmin Bao, Dong Chen, Fang Wen, Yong Wang, and Baining Guo. Styleswin: Transformer-based gan for high-resolution image generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11304–11314, 2022. 2
- [86] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 3
- [87] Hongkai Zheng, Weili Nie, Arash Vahdat, and Anima Anandkumar. Fast training of diffusion models with masked transformers. *arXiv:2306.09305*, 2023. 2, 5, 1
- [88] Daquan Zhou, Weimin Wang, Hanshu Yan, Weiwei Lv, Yizhe Zhu, and Jiashi Feng. Magicvideo: Efficient video generation with latent diffusion models. *arXiv preprint arXiv:2211.11018*, 2022. 8
- [89] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *CoRL*, 2023. 1

	T2V (base)
Input	$5 \times 16 \times 28$
Spatial window	$1 \times 16 \times 28$
Spatiotemporal window	$5 \times 8 \times 14$
Training steps	250000
Batch size	512
lr schedule	Constant
Optimizer	Adafactor
lr	0.00005

Table 6. Hyperparameters for aspect-ratio finetuning.

A. Implementation Details

For the first stage, we follow the architecture and hyperparameters from Yu et al. [82]. We report hyperparameters specific for training our model in Table 8. To train the second stage transformer model, we use the default settings of $1 \times 16 \times 16$ spatial window, $5 \times 8 \times 8$ spatiotemporal window, $p_{sc} = 0.9$, $c = 8$ and $r = 2$. We summarize additional training and inference hyperparameters for all tasks in Table 8. The UCF-101 model results reported in Tables 1 and 4 are trained for 60,000 steps. We perform all ablations on UCF-101 with 35,000 training steps.

Aspect-ratio finetuning. To simplify training and leverage broad data sources with different aspect ratios, we train the base stage using a square aspect ratio. We fine-tune the base stage on a subset of data to generate videos with a $9 : 16$ aspect ratio. We interpolate the absolute and relative position embeddings and scale the window sizes. We summarize the finetuning hyperparameters in Table 6.

Long video generation. As described in § 4.4, we train our model jointly on the task of frame prediction. During inference, we generate videos as follows: Given a natural language description of a video, we first generate the initial 17 frames using our base model. Next, we encode the last 5 frames into 2 latent frames using our causal 3D encoder. Providing 2 latent frames as input for subsequent autoregressive generation helps ensure that our model can maintain continuity of motion and produce temporally consistent videos.

UCF-101 Text-to-Video. We follow the evaluation protocol of prior work [24], and adapt their prompts to better describe the UCF-101 classes.

B. Additional Results

B.1. Image Generation

We compare (Table 7) W.A.L.T with state-of-the-art image generation methods for 256×256 resolution with classifier free guidance. Unlike, prior work [22, 53, 87] using Transformer for diffusion modelling, we did not observe significant benefit of using vanilla classifier free guidance. Hence, we report results using the power cosine schedule proposed by Gao et al. [22]. Our model performs better than prior works on the Inception Score metric, and achieves

Method	Cost (Iter×BS)	FID↓	IS↑	Params.	Steps
LDM-4 [61]	178k×1200	3.60	247.7	400M	250
DiT-XL/2 [53]	7000k×256	2.27	278.2	675M	250
ADM [16]	-	3.94	215.8	608M	2000
MDT [22]	6500k×256	1.79	283.0	676M	250
MaskDiT [87]	1200k×1024	2.28	276.6	736M	40
simple diffusion [38]	500K×2048	2.44	256.3	2B	512
VDM++ [45]	-	2.12	267.7	2B	512
W.A.L.T-L <i>Ours</i>	437k×1024	2.40	290.5	460M	50

Table 7. Class-conditional image generation on ImageNet 256×256 . We adopt the evaluation protocol and implementation of ADM [16] and report results with classifier free guidance.



Figure 5. ImageNet class-conditional generation samples.

competitive FID scores. Fig. 5 shows qualitative samples.

B.2. Video Generation

We show samples for Kinetics-600 frame prediction in Fig. 6.

B.3. Image-to-Video

As noted in Section 4.4, we train our model jointly on the task of frame prediction, where we condition on 1 latent frame. This allows us to leverage the high quality first frame from the image generator as context for predicting subsequent frames. For qualitative results see videos on our [project website](#).

	ImageNet	UCF-101	K600	T2V (base)	T2V (2×)	T2V (2×2×)
<i>First Stage</i>						
Input	$1 \times 256 \times 256$	$17 \times 128 \times 128$	$17 \times 128 \times 128$	$17 \times 128 \times 128$	$17 \times 256 \times 448$	$17 \times 512 \times 896$
f_s, f_t	8, -	8, 4	8, 4	8, 4	8, 4	8, 4
Channels	128	128	128	128	128	128
Channel multiplier	1,1,2,4	1, 2, 2, 4	1, 2, 2, 4	1, 2, 2, 4	1, 2, 2, 4	1, 2, 2, 4
Training duration	270 epochs	2000 epochs	270000 steps	1000000 steps	1000000 steps	1000000 steps
Batch size	256	256	256	256	256	256
lr schedule	Cosine	Cosine	Cosine	Cosine	Cosine	Cosine
Optimizer	Adam	Adam	Adam	Adam	Adam	Adam
<i>Second Stage</i>						
Input	$1 \times 32 \times 32$	$5 \times 16 \times 16$	$5 \times 16 \times 16$	$5 \times 16 \times 16$	$5 \times 32 \times 56$	$5 \times 64 \times 112$
Layers	24	28	24	52	40	24
Hidden size	1024	1152	1024	9216	1408	1024
Heads	16	16	16	16	16	16
Training duration	350 epochs	60000 steps	360 epochs	550000 steps	675000 steps	275000 steps
Batch size	1024	256	512	512	512	512
lr schedule	Cosine	Cosine	Cosine	Cosine	Cosine	Cosine
Optimizer	AdamW	AdamW	AdamW	Adafactor	Adafactor	Adafactor
lr	0.0005	0.0005	0.0005	0.0002	0.0005	0.0005
EMA	✓	✓	✓	✗	✗	✗
Patch size	1	1	1	1	2	4
AdaLN-LoRA	✗	2	2	2	2	2
<i>Diffusion</i>						
Diffusion Steps	1000	1000	1000	1000	1000	1000
Noise schedule	Linear	Linear	Linear	Linear	Linear	Linear
β_0	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
β_{1000}	0.02	0.02	0.02	0.02	0.02	0.02
Sampler	DDIM	DDIM	DDIM	DDIM	DDIM	DDIM
Sampling steps	50	50	50	50	50	50
Guidance	✗	✗	✗	✓	✓	✓

Table 8. Training and evaluation hyperparameters.



Figure 6. Frame prediction samples on Kinetics-600. Top: ground-truth, where unobserved frames are shaded. Bottom: generation.