VIDEO-GPT VIA NEXT CLIP DIFFUSION

Shaobin Zhuang[1,*] **Zhipeng Huang**[2,*] **Ying Zhang**[4] **Fangyikang Wang**[3,*]
Canmiao Fu[4] **Binxin Yang**[4] **Chong Sun**[4] **Chen Li**[4] **Yali Wang**[5,6,†]

[1]Shanghai Jiao Tong University [2]University of Science and Technology of China
[3]Zhejiang University [4]WeChat Vision, Tencent Inc.
[5]Shenzhen Key Lab of Computer Vision and Pattern Recognition, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences [6]Shanghai AI Laboratory

{hahahahaha}@sjtu.edu.cn {ethanphuang, yinggzhang}@tencent.com
{kenelmfu, binxinyang, waynecsun, chaselli}@tencent.com
wangfangyikang@zju.edu.cn yl.wang@siat.ac.cn

ABSTRACT

GPT has shown its remarkable success in natural language processing. However, the language sequence is not sufficient to describe spatial-temporal details in the visual world. Alternatively, the video sequence is good at capturing such details. Motivated by this fact, we propose a concise Video-GPT in this paper by treating video as new language for visual world modeling. By analogy to next token prediction in GPT, we introduce a novel next clip diffusion paradigm for pretraining Video-GPT. Different from the previous works, this distinct paradigm allows Video-GPT to tackle both short-term generation and long-term prediction, by autoregressively denoising the noisy clip according to the clean clips in the history. Extensive experiments show our Video-GPT achieves the state-of-the-art performance on video prediction, which is the key factor towards world modeling (Physics-IQ Benchmark: **Video-GPT 34.97** ***vs.*** **Kling 23.64** ***vs.*** **Wan 20.89**). Moreover, it can be well adapted on 6 mainstream video tasks in both video generation and understanding, showing its great generalization capacity in downstream.

1 INTRODUCTION

Over the past few years, natural language processing has been mainly driven by training large language models (LLMs) from web-scale text data. In particular, GPT series Radford & Narasimhan (2018); Radford et al. (2019); Brown et al. (2020); Achiam et al. (2023) have demonstrated remarkable generalization capabilities on various downstream language tasks, which further opens the possibility towards general artificial intelligence. However, the language sequence is often good at expressing high-level abstractions, while it may not be sufficient to capture rich spatial-temporal details in the visual world Liu et al. (2024); Yang et al. (2024b). For example, it is difficult to describe how to tie a knot by language. Alternatively, the video sequence has been considered as a preferred candidate to describe such details, since it can record visual knowledge of our dynamical world at different spatial and temporal resolutions Bai et al. (2023); Liu et al. (2024). Hence, there is a natural question: *Can we treat video as new language for visual world modeling?*

The recent studies have shown that, video generation is a promising direction to achieve this goal OpenAI (2024); Bruce et al. (2024); Kondratyuk et al. (2023). The typical design is video diffusion by adding noise and denoising gradually Song et al. (2021); Ho et al. (2020). Although such a paradigm has achieved significant progress Bar-Tal et al. (2024); Kong et al. (2024); Kuaishou (2024); Wang et al. (2025), it often suffers from difficulty in long-term future prediction that is a critical factor of world modeling Ouyang et al. (2024); Li et al. (2024). To address this problem, the autoregressive attempts have been made for long-context video modeling, on analogy of next token prediction in LLMs Kondratyuk et al. (2023); Liu et al. (2024); Bai et al. (2023). However,

[*]Work done as interns at WeChat Vision, Tencent Inc.
[†]Corresponding author.

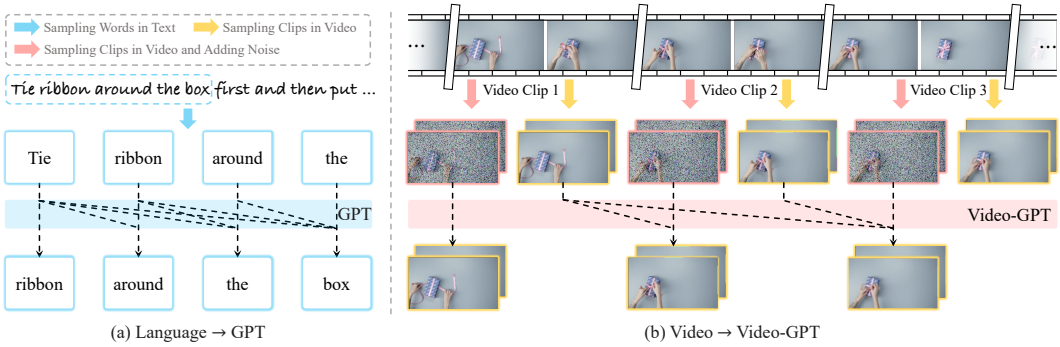


Figure 1: **Next clip diffusion.** We draw an analogy with GPT's next token prediction and model each video clip as a visual word by denoising the next noisy clip, conditioning on the previous video.

compared with its diffusion counterpart, the generation performance of this paradigm still needs to be further improved. In order to leverage advantages from both paradigms, several approaches combine diffusion and autogregressive modeling in a unified transformer style Xiao et al. (2024); Deng et al. (2024); Zhou et al. (2024); Huang et al. (2025b); Hu et al. (2024); Deng et al. (2025). But these design mainly works on the image domain, without insightful analogy between language and video.

To fill this gap, we propose a concise Video-GPT, by analogy with generative pretraining design in GPT. Inspired by next token prediction in GPT, we introduce a novel *next clip diffusion* paradigm for our Video-GPT. Specifically, we creatively treat a *clip* in the video as the role of a *word* in the language, since both of them describe local temporal information, respectively, in the video and language sequence. However, different from a discrete word token, it is often challenging to predict a continuous video clip in the next step. To address this difficulty, we introduce a flexible diffusion design within an interleaved sequence of noisy and clean clips in the temporal order. As shown in Fig. 1, one can randomly sample a number of clips from a training video. For each clip, we generate its noisy clip by adding noise on it, based on the forward process of diffusion. Then, we re-arrange noisy and clean clips in temporal order, constructing an interleaved clip sequence. Consequently, for each noisy clip, we leverage the clean clips in the previous as context, and reconstruct the corresponding clean clip by the diffusion loss. In this case, our Video-GPT inherits advantages from both GPT and diffusion, but within the basic unit of a video clip. This allows it to capture long-term video prediction as well as short-term video generation for effective generative pretraining on videos.

Our contributions can be summarized as follows. *First*, we introduce a concise and self-supervised pretrained Video-GPT, by analogy to GPT. Different from GPT, our Video-GPT aims at modeling rich spatial-temporal information in the visual world. *Second*, we design a novel next clip diffusion paradigm for pretraining Video-GPT. Different from the previous works mentioned above, this hybrid paradigm allows Video-GPT to tackle both short-term generation and long-term prediction, via integrating diffusion and autoregressive design in the video clip level. Finally, our pretrained Video-GPT achieves state-of-the-art performance on Physics-IQ Benchmark and Kinetics-600, fully demonstrating its potential for visual world modeling. After fine-tuning on 6 downstream video generation and understanding tasks, it also achieved preferable performance for generalization.

2 Related Work

Video Diffusion Models. With diffusion models demonstrating remarkable performance in image generation Saharia et al. (2022); Rombach et al. (2021); Dhariwal & Nichol (2021); Ramesh et al. (2022), video diffusion models Ho et al. (2022) also surpasses traditional methods Clark et al. (2019); Saito et al. (2018); Kahembwe & Ramamoorthy (2019) in video generation. With the emergence of large-scale text–video datasets Chen et al. (2024b); Wang et al. (2024c; 2023c), text-to-video diffusion models Blattmann et al. (2023); Zhuang et al. (2024); Yang et al. (2024c); OpenAI (2024); Wang et al. (2025); Kong et al. (2024); Wang et al. (2023b) exhibit realistic generation capabilities and strong transferability to downstream tasks Yue et al. (2025); Hu & Xu (2023). However, the supervised paradigm based on text annotations faces two challenges as it scales up. **1) Quantity**

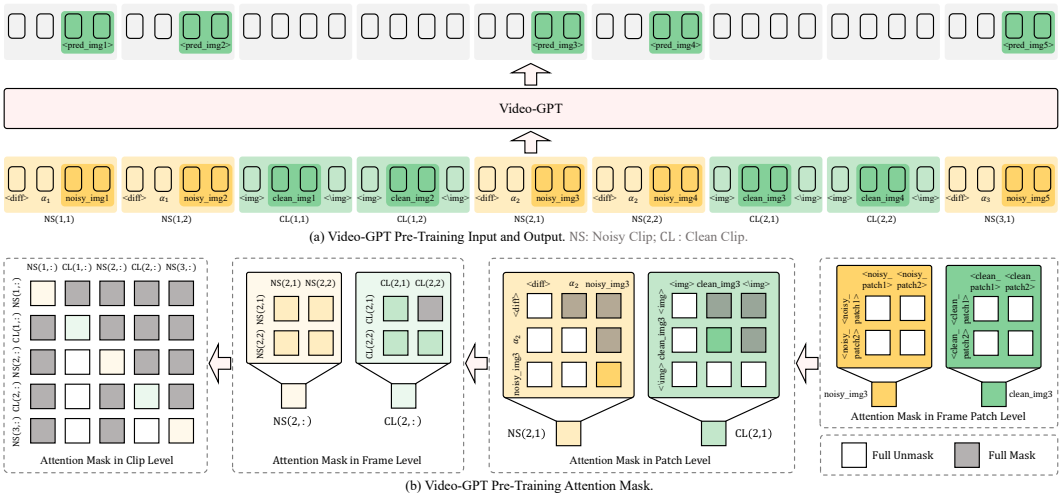


Figure 2: **Video-GPT pretraining framework.** The full attention mask is shown in Fig. 10

Table 1: **Progressive training strategy.**

Stage	Resolution	Frame Num.	Frame Interval	Clip Num.	Train Steps
1	Flexible	16	4	16	300000
2	Flexible	48	4	$\sim$ Uniform$\{2, 3, \cdots, 48\}$	25000
3	Flexible	48	$\sim$ Uniform$\{4, 5, \cdots, 12\}$	$\sim$ Uniform$\{2, 3, \cdots, 48\}$	40000
4	Flexible	80	$\sim$ Uniform$\{4, 5, \cdots, 12\}$	$\sim$ Uniform$\{2, 3, \cdots, 80\}$	20000

of Annotated Data: Current methods require meticulous data filtering and annotation processes Menapace et al. (2024); Polyak et al. (2024), resulting substantially smaller data scale compared to LLM Dubey et al. (2024); Yang et al. (2024a); Kaplan et al. (2020). **2) Quality of Annotated Data**: Text cannot fully capture the meaning of a video. While increasing the granularity of descriptions Chen et al. (2023a); Betker et al.; Segalis et al. (2023) alleviates this issue, the difficulty of annotating such descriptions significantly increases as video length scales up.

Autoregressive Video Models. ImageTransformer Parmar et al. (2018) migrated next token prediction to image generation in pixel space. Chameleon Team (2024) and LlamaGen Sun et al. (2024) perform pretraining on images in latent space. LVM Bai et al. (2023) and VideoWorld Ren et al. (2025) reveal that models can acquire world knowledge from videos. LWM Liu et al. (2024) compresses videos into latent space for pretraining, but focuses on video understanding capabilities. In general, next token prediction based visual generation Wang et al. (2024d) still underperform compared with the most advanced diffusion models Labs (2024); Liao et al. (2025); Wang et al. (2025); Kong et al. (2024). In contrast, our Video-GPT adopts a next clip diffusion paradigm, which diffusion generation within video clips and applies autoregressive generation between clips. It enhances video generative modeling while utilizing historical information as a self-supervised condition.

Video Generation Models with Hybrid Modeling. Recently, video generation models that hybridize autoregressive and diffusion modeling have begun to emerge. Self-forcing Huang et al. (2025a) is based on the paradigm of next frame prediction, using diffusion for generation within each frame, and employing autoregression between frames. APT2 Lin et al. (2025) introduces the autoregressive adversarial post-training to achieve real-time video generation. Video-GPT differs from them in two fundamental ways: 1) **Self-Supervised Pre-Training:** Video-GPT is pre-trained through self-supervised pre-training similar to LLM, does not require any annotations other than videos in this process. Through pre-training on the open source Panda-70M dataset Chen et al. (2024b), we achieve and surpass the physical world modeling capabilities of the top closed-source models like Gen3 Runway (2024) and Sora OpenAI (2024). 2) **Next Clip Diffusion:** Video-GPT is autoregressive at the clip level. This is more efficient and flexible as it allows parallel, bidirectional processing (diffusion) within a multi-frame clip, but maintains temporal order between clips.

3 VIDEO-GPT

In this section, we first introduce our Video-GPT via next clip diffusion. Subsequently, we describe how to adapt it for typically downstream tasks in both video generation and understanding. For better understanding, we also present the pseudocode of Video-GPT in Sup. A.

3.1 INPUT: CLIP SEQUENCE CONSTRUCTION

As mentioned before, we notice that a clip in the video plays a similar role as a word in the language. Hence, we first divide a training video into a number of clips to construct the basic processing unit in our Video-GPT. Specifically, we uniformly sample N frames from each training video. Then, we randomly divide these frames into K clips, where K is sampled by $K \sim \text{Uniform}\{2, 3, \cdots, N\}$.

Forward Diffusion Process on Each Clip. Since diffusion has shown the powerful capacity of video generation, we propose to leverage it to predict the next video clip based on the previous videos. According to this reason, we perform the forward diffusion process on latent of each clip by adding Gaussian noise. Specifically, for the k-th clip, we first apply continues VAE Rombach et al. (2021) to compress each frame in this clip, and patchify it to obtain the latent feature of each frame (i.e., the feature matrix of all the patch tokens in this frame). Next, we choose flow matching Lipman et al. (2022) for diffusion, due to its efficiency. According to flow matching, we add the noise on the latent feature in the following way,

$$\Psi(k, i, \alpha_k) = \alpha_k \Phi(k, i) + (1 - \alpha_k)\varepsilon_{k,i}, \tag{1}$$

where $\Phi(k, i)$ is the latent feature of the i-th frame in the k-th clip, and $\Psi(k, i)$ is the noisy feature of this frame. The weight α_k is sampled by $\alpha_k \sim \text{Uniform}[0, 1]$, and the noise $\varepsilon_{k,i}$ is sampled by $\varepsilon_{k,i} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Note that, in order to inference the content in the video clip in parallel, we apply the same α_k for all frames in the video clip during training.

Noise-Clean Interleaved Clip Sequence. After getting the noisy clips, we next arrange the input sequence for our Video-GPT. By analogy of GPT, we leverage the historical clips as temporal context to denoise the noisy clip in the next step. But instead of using the noisy clips in the history Chen et al. (2024a); Yi et al. (2025), we leverage the original clean clips in the history, in order to denoise the next clip conditioned on the clean and correct context. Hence, we feed both noisy and clean clips as input to our Video-GPT. To distinguish different frames in these clips, we add extra tokens for specializing them. **1) Clip-Level** Token Form of Clean Clip: For each clean clip, we add the boundary tokens with the latent feature of each frame in this clip,

$$\mathbf{CL}(k, i) = [\text{<img>}, \Phi(k, i), \text{<\img>}]. \tag{2}$$

Hence, the clip-level token form of the k-th clean clip is $\mathbf{CL}(k, :) = [\mathbf{CL}(k, 1), ..., \mathbf{CL}(k, N_k)]$, where N_k is the number of frames in this clip. **2) Clip-Level** Token Form of Noisy Clip: For each noisy clip, we add two extra tokens with the noisy latent feature of each frame in this clip,

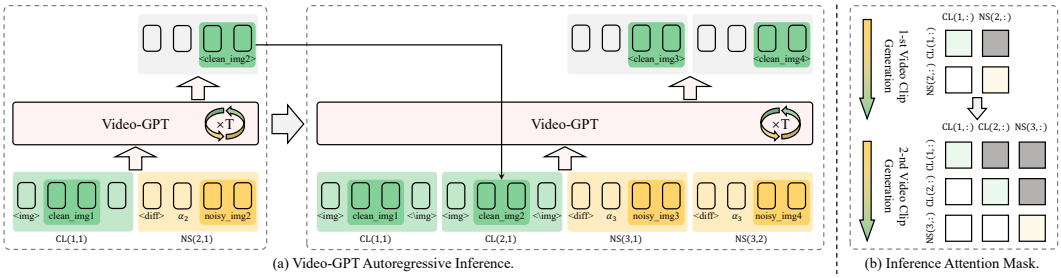
$$\mathbf{NS}(k, i) = [\text{<diff>}, \alpha_k, \Psi(k, i, \alpha_k)]. \tag{3}$$

where <diff> is a denoising hint token to indicate that $\Psi(k, i)$ is noisy. Moreover, we also add the weight α_k for the k-th noisy clip, in order to provide the timestep information in the flow matching. As a result, the clip-level token form of the k-th noisy clip is $\mathbf{NS}(k, :) = [\mathbf{NS}(k, 1), ..., \mathbf{NS}(k, N_k)]$. **3) Noise-Clean Interleaved Input**: As shown in Fig. 2 (a), after formulating both clean and noisy clips, we arrange them together as the input sequence to Video-GPT. Since we aim at sequential modeling, we take a pair of noisy and clean clips as a group, and arrange these groups in the temporal order. As a result, we obtain an interleaved sequence of noisy and clean clips as input, i.e.,

$$\mathbf{Input} = [\mathbf{NS}(1, :), \mathbf{CL}(1, :), ..., \mathbf{NS}(k, :), \mathbf{CL}(k, :), ..., \mathbf{NS}(K, :)]. \tag{4}$$

3.2 PRETRAINING: NEXT CLIP DIFFUSION

After obtaining the input sequence, we feed it into our Video-GPT for generative pretraining. By analogy to GPT, we use the vanilla transformer architecture for conciseness. The next question is how to build up relations between noisy and clean clips for next clip diffusion. As shown in Fig. 2 (b), we introduce a hierarchical masking method to indicate such relations in the attention operation.



(a) Video-GPT Autoregressive Inference. (b) Inference Attention Mask.

Figure 3: **Video-GPT inference framework.** We iteratively denoise the 2nd noisy clip $\mathbf{NS}(\mathbf{2},:)$ to $\mathbf{CL}(\mathbf{2},:)$, and use it along with the 1st clean clip $\mathbf{CL}(\mathbf{1},:)$ to condition the prediction of the 3rd noisy clip $\mathbf{NS}(\mathbf{3},:)$. The number of frames in each clip can also vary during inference.

Clip-Level Mask. As our input is a clip sequence, we first need to define the dependence among clips. By analogy to the dependence of words in GPT, we basically apply a causal relation as clip dependence. **1) Clean Clip Mask**. As shown in Fig. 2 (b), the k-th clean clip depends on itself and the previous $(k-1)$ clean clips. **2) Noisy Clip Mask**. Alternatively, the k-th noisy clip depends on itself for sure. However, instead of depending on the previous $(k-1)$ noisy clips, it depends on the previous $(k-1)$ clean clips. The reason is that, these clean clips in the history provide the correct temporal context for denoising the k-th noisy clip. It is the core idea in our next clip diffusion.

Frame-Level Mask. Since each clip consists of several frames, we need to further specify the dependence among frames in the defined clip-level mask. **1) Clean Frame Mask**. As shown in Fig. 2 (b), for the i-th frame in the k-th clean clip, it depends on itself, the $(i-1)$ frames in this clean clip, and all the frames in the previous $(k-1)$ clean clips. **2) Noisy Frame Mask**. Alternatively, for the i-th frame in the k-th noisy clip, it depends on itself, all the other frames in this noisy clip, and all the frames in the previous $(k-1)$ clean clips. Note that, different from the causal design for the clean frame, the i-th noisy frame depends on all the other frames in the same noisy clip. The reason is that iterative denoising turns noisy clips into clean clip as history condition, the mask of noisy frames will not affect later inference and bidirectional attention can better enhance generation quality Yin et al. (2024); Kim et al. (2021).

Patch-Level Mask. As shown in Eq. (2) and (3), each frame (clean or noisy) is formulated by extra hint tokens, and the latent feature of this frame which actually refers to the feature matrix of all patch tokens in this frame. Hence, we need to further specify the dependence among these hint tokens and patch tokens in the defined frame-level mask. **1) Clean Patch Mask**. As shown in Fig. 2 (b), we follow the similar design of GPT, the dependence is causal among tokens, i.e., <img>, $\Phi(k,i)$, and <\img>. But differently, the clean frame feature $\Phi(k,i)$ is actually the feature matrix of all the patch tokens in this frame. Since the patch tokens describe spatial relation rather than temporal one, we define the full dependence among these patch tokens. **2) Noisy Patch Mask**. Similar to clean patch mask, we define the causal dependence among tokens, i.e., <diff>, α_k, and $\Psi(k,i)$. We also define the full dependence among patch tokens in noisy frame as the same as the one in clean frame.

Training Target. Based on this hierarchical masking, our Video-GPT can effectively leverage the masked attention to denoise the k-th noisy clip in the next step, according to the previous $(k-1)$ clean clips of input sequence. Subsequently, we compute the $\mathcal{L}_2$ loss between each denoised clip feature and its corresponding clean clip feature to pretrain our Video-GPT. It is worth mentioning that, different from previous diffusion works, we employ Video-GPT to predict the video clip directly instead of noise Chen et al. (2023a); Wang et al. (2023b) or velocity Esser et al. (2024); Yang et al. (2024c), to keep the training setting as simple as possible, allowing to easily adapt the pretrained Video-GPT for various downstream tasks.

Progressive Training. As the length of the video frame increases, the computational cost of the attention operation will increase quadratically. To alleviate such computation, we leverage a simple but effective progressive training strategy. Specifically, we start training from short videos to long videos. As shown in Tab. 1, the reason is that in the initial pretraining stage we use 16 frames—with each clip containing only one frame for next-frame prediction—then gradually increase both the number of clips and frames per clip for Video-GPT pretraining.

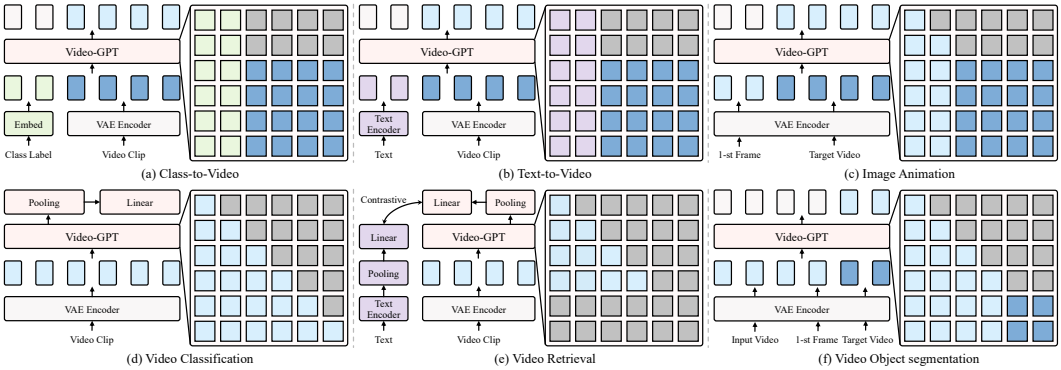


Figure 4: **Fine-tuning Video-GPT on downstream tasks.**

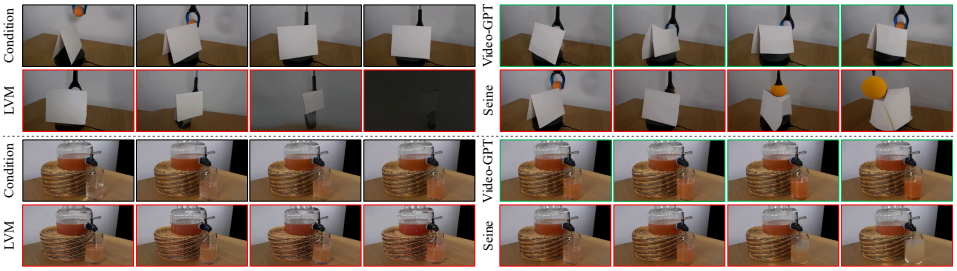


Figure 5: **Qualitative results on Physics-IQ Benchmark.** The videos predicted by our Video-GPT based on condition frames are more consistent with physical laws than other methods.

3.3 Inference: Autoregressive Video Prediction

After pretraining Video-GPT, one can naturally leverage it for video prediction, by analogy to GPT. As shown in Fig. 3 (a), video prediction can be autoregressively performed in the inference phase. Specifically, we treat the previous k denoised clips (outputs of Video-GPT in the history) as the clean clips, and leverage them as temporal context to denoise the $(k+1)$ noisy clip,

$$\mathbf{DNS}(k+1,:) = \textbf{Video-GPT}\Big(\ \mathbf{DNS}(1,:), ..., \mathbf{DNS}(k,:),\ \ \mathbf{NS}(k+1,:)\ \Big), \tag{5}$$

where $\mathbf{DNS}(k+1,:)$ is the $(k+1)$-th denoised clip. Additionally, $\mathbf{DNS}(1,:)$ is a clean clip which is initially given as input for video prediction. Note that, the input of Video-GPT in this inference phase is a causal formulation. Hence, the clip-level mask is defined as the causal one for each iteration of autoregressive prediction, as shown in Fig. 3 (b). We use all previously generated clips as context, up to the maximum sequence length supported by model pre-training. To generate longer videos than fit within this context window, a standard sliding window approach is adopted.

3.4 Generalization on Downstream Video Tasks

As we noticed, GPT has shown its powerful generalization capacity on various downstream NLP tasks. Hence, we expect that Video-GPT can show the similar capacity on various downstream video tasks. For this reason, we adapt Video-GPT to address typical downstream video tasks in Fig. 4, covering both video generation and understanding. For video generation, we focus on class-to-video generation, text-to-video generation, and image animation. For video understanding, we focus on video classification, video retrieval, and video object segmentation.

Class-to-Video Generation and Text-to-Video Generation. As shown in Fig. 4, we use a similar processing mode for these two types of tasks, since both generation tasks of Video-GPT are conditioned on extra tokens (either from a category tag or a video description). Hence, we introduce a causal attention between extra tokens and clip tokens, i.e., a extra token only depends on other extra tokens, while a clip token depends on both extra and clip tokens.

Image Animation and Video Object Segmentation. Since both tasks are to predict video content conditioned on the given sequence, we discuss them together here. For image animation, we are given the first frame. The goal is to predict the rest video conditioned on the first frame. For video

Table 2: **Quantitative comparison of models evaluated on the Physics-IQ Benchmark.**

Model	*w/o.* Text	Spatial IoU↑	Spatio Temporal↑	Weighted Spatial IoU↑	MSE↓	Phys. IQ Score↑
Sora (I2V) OpenAI (2024)	✗	0.138	0.047	0.063	0.030	10.00
Pika 1.0 (I2V) PikaLabs (2024)	✗	0.140	0.041	0.078	0.014	13.00
SVD (I2V) Blattmann et al. (2023)	✓	0.132	0.076	0.073	0.021	14.80
LVM (V2V) Bai et al. (2023)	✓	0.100	0.147	0.077	0.021	18.02
Lumiere (I2V) Bar-Tal et al. (2024)	✗	0.113	0.173	0.061	0.016	19.00
Open-Sora-Plan v1.3.0 (I2V) Lin et al. (2024)	✗	0.142	0.139	0.074	0.021	19.42
Wan2.1 (I2V) Wang et al. (2025)	✗	0.153	0.100	0.112	0.023	20.89
HunyuanVideo Kong et al. (2024)	✗	-	-	-	-	22.36
Gen 3 (I2V) Runway (2024)	✗	0.201	0.115	0.116	0.015	22.80
Lumiere (V2V) Bar-Tal et al. (2024)	✗	0.170	0.155	0.093	0.013	23.00
Kling1.6 (I2V) Kuaishou (2024)	✗	0.197	0.086	0.144	0.025	23.64
LTX-Video-I2V HaCohen et al. (2024)	✗	-	-	-	-	26.80
Cogvideo-I2V-5B Hong et al. (2022)	✗	-	-	-	-	27.1
Seine (V2V) Chen et al. (2023b)	✗	0.163	0.208	0.131	0.012	29.13
VideoPoet (V2V) Kondratyuk et al. (2023)	✗	**0.204**	0.164	0.137	0.010	29.50
Video-GPT (V2V)	✓	0.198	**0.240**	**0.144**	**0.007**	**34.97**

Table 3: **Quantitative comparison of video generation models evaluated on the Kinetics-600.**

Model	Parameters	Architecture	Text Encoder	FVD(500)↓	FVD(5000)↓
Open-Sora-Plan v1.3.0 Lin et al. (2024)	2B	DiT	mT5-XXL Xue et al. (2020)	384.27	95.36
LVM Bai et al. (2023)	7B	Vanilla Transformer	-	356.48	94.61
Seine Chen et al. (2023b)	0.9B	U-Net	CLIP-ViT-L Radford et al. (2021)	332.80	91.08
Video-GPT	3.8B	Vanilla Transformer	-	**315.40**	**89.44**

Table 4: **Inference setting ablation.**

	Number of Frames in Each Video Clip							History Conditioned Classifier-Free Guidance					
Hyperparams.	1	2	4	8	16	**32**	54	1	1.5	2.0	2.5	**3.0**	3.5
Phy. IQ Score	0.00	6.21	18.44	29.90	32.86	**34.94**	32.18	25.09	31.96	32.22	32.65	**32.72**	31.88

Table 5: **Training setting ablation.**

	Pretraining Paradigm		Pretraining Frame Number			Add Noise to Clean Clip		Dataset Scale	
Hyperparams.	Next Token Prediction	**Next Clip Diffusion**	16	48	**80**	NO	**YES**	1M	**70M**
Phy. IQ Score	21.59	**34.94**	22.06	33.09	**34.94**	32.54	**33.09**	23.16	**33.09**

object segmentation, we are given a video and the first frame with the masked objects. The goal is to track the masked objects throughout the video. To address these tasks by fine-tuning, we leverage the input frame or video as the clean clip in our Video-GPT, and the video to be generated as the noisy clip. The mask strategy follows the settings in Sec. 3.2.

Video Classification and Video Retrieval. For video classification, we pool clean clip features from Video-GPT for linear probing. For video retrieval, we use the clean clip feature for contrastive learning with the text feature from a text encoder, e.g., CLIP-ViT-L-Patch14 Radford et al. (2021).

4 Experiments

4.1 Implementation

Dataset. In pretraining, we employ Panda-70M Chen et al. (2024b) as the training dataset. In order to vertify the impact of data scale on model performance, we also employ OpenVid-1M Nan et al. (2024) for ablation. For downstream tasks, we fine-tune the model with supervised datasets. Specifically, for class-to-video and video classification, we conduct experiments on UCF-101 Soomro et al. (2012). For text-to-video generation, we fine-tune the model with OpenVid-1M, and for the video retrieval task, we used videos and annotated texts from the Panda-70M. For video object segmentation, we randomly select 1000 cases from GetIn-1M Zhuang et al. (2025). Notably, for image animation—which is in high demand within the industry, there is no commonly recognized benchmark. Therefore, we colloct three datasets from the internet to evaluate the generalization capability of pretrained Video-GPT on image animation. These datasets correspond to the categories of *lie down*, *transform*, and *fly*, with each dataset containing fewer than one hundred videos.

Settings. In this paper, we use VAE from SDXL Podell et al. (2023), with Video-GPT inheriting the model architecture from Phi-3-mini Abdin et al. (2024). We additionally incorporated

Figure 6: **Qualitative results of Video-GPT on class-to-video generation on UCF-101.**

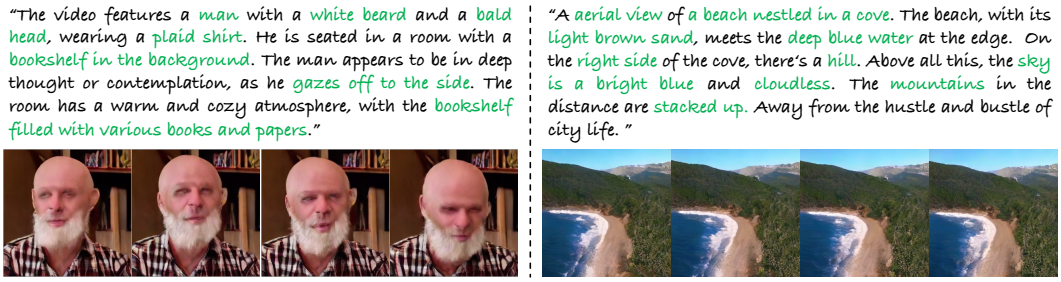


Figure 7: **Qualitative results of Video-GPT on text-to video-generation.**

`clean_input` and `noised_input` layers Xiao et al. (2024) to transform VAE-encoded image latents into tokens, as well as a `noised_output` layer to convert output tokens back into image latents, with these components having significantly fewer parameters than the main transformer. During the pretraining phase, we employ 320 H20 GPUs and implement a progressive training strategy as shown in Tab. 1. More pretraining and fine-tuning settings can be found in Sup. B.

4.2 Pretrained Model Evaluation

We evaluate pretrained Video-GPT performance on Physics-IQ Benchmark Motamed et al. (2025) and Kinetics-600 Carreira et al. (2018), which are used to evaluate the ability of models in predicting deterministic and uncertain videos respectively. We set the historical conditioned classifier-free guidance scale c=3. Unless specified all subsequent experiments keeps the same setting.

Physics-IQ Benchmark. Physics-IQ Benchmark presents a short video clip (3 seconds) depicting real-world physical motion and asks the model to predict future frames (5 seconds). Since macroscopic physical laws typically exhibit deterministic behavior, we used the Physics-IQ Benchmark to assess the model's ability to predict deterministic future events. As shown in Tab. 2, Video-GPT has a superior performance in modeling the physical world (more than **5 points** higher than the 2nd). In addition to Video-GPT, the results of LVM Bai et al. (2023), Open-Sora-Plan v1.3 Lin et al. (2024), and Seine Chen et al. (2023b) are also obtained by our own testing. This result shows that the model based on next clip diffusion pretraining paradigm can better learn world knowledge from video data.

Kinetics-600. Kinetics-600 contains 0.3M human motion videos, and human motion is often known for being unpredictable. Therefore, we measure the distance Unterthiner et al. (2018) between model-generated videos and videos from Kinetics-600 to evaluate the model's capability to predict highly uncertain future events, requiring the model to predict 13 future frames based on the 3 given frames. As shown in Tab. 3, Video-GPT achieves the best FVD among all models with a vanilla transformer architecture, rather than the more popular U-Net Ronneberger et al. (2015) or DiT Peebles & Xie (2022), which fully demonstrates the effectiveness of our pretraining.

Visualization. As shown in Fig. 5, Video-GPT accurately predicts the water filling and machine lifting physics, while other methods Bai et al. (2023); Chen et al. (2023b) show obvious errors.

4.3 Ablation Study

We ablate the properties of Video-GPT pretraining and inference in the Physics-IQ Benchmark. Unless specified, we perform ablation experiments based on the Video-GPT Stage 3 in Tab. 1.

Table 6: **Class to video quantitative comparison on the UCF-101.**

Model	Resolution	Architecture	VAE	FVD↓
CogVideo Hong et al. (2022)	160 × 160	Dual-channel Transformer	2D	626
Latte Ma et al. (2024)	256 × 256	DiT	2D	478
TATS Ge et al. (2022)	128 × 128	Time-Sensitive Transformer	-	332
OmniTokenizer Wang et al. (2024b)	256 × 256	Vanilla Transformer	3D	191
VideoFusion Luo et al. (2023)	128 × 128	U-Net	-	173
MAGVITv2-AR Yu et al. (2024)	-	Vanilla Transformer	3D	109
ACDIT Hu et al. (2024)	-	DiT	2D	90
Make-A-Video Singer et al. (2022)	256 × 256	U-Net	2D	81
MAGVITv2 Yu et al. (2024)	-	Vanilla Transformer	3D	58
LARP Wang et al. (2024a)	128 × 128	Vanilla Transformer	3D	57
FAR Gu et al. (2025)	128 × 128	DiT	2D	57
Video-GPT (from scratch)	240 × 320	Vanilla Transformer	2D	489
Video-GPT (finetune)	**240 × 320**	**Vanilla Transformer**	**2D**	**53**

Table 7: Video classification linear probe on UCF-101.

Method	Top-1
MemDPC Han et al. (2020)	54.1
VideoMAEv2 Wang et al. (2023a)	56.4
Video-GPT	**58.9**

Table 8: Video retrieval zero-shot on the MSR-VTT.

Method	R@1
VideoCLIP Xu et al. (2021)	10.4
SupportSet Patrick et al. (2020)	12.7
FiT Bain et al. (2021)	18.8
Video-GPT	**22.8**

Figure 8: **Qualitative results of Video-GPT on *lie down*, *transform* and *fly* image animation.** The frame in the blue box is the prefix frame we are given as history condition.

Figure 9: **Qualitative results of Video-GPT on video object segmentation.** The frames in blue box are condition frames. The frame in red box is the 1-st frame of the condition with object mask, and we circle the segmented object in red. Frames in green box are the segmentation result.

Number of Frames in Video Clip for Inference. As shown in Tab. 4, within a certain range, as the number of frames N_k processed in parallel for inference within each video clip increases, the quality of the videos generated by the model improves significantly. This thoroughly validates the superiority of our proposed next clip generation paradigm for video generation tasks.

History Conditioned Classifier-Free Guidance Scale for Inference. We conduct experiments based on N_k=16. As shown in Tab. 4, the Stage 4 model shows the strongest performance when we adjust the scale of history conditioned classifier-free guidance Ho (2022) c=3.0.

Number of Frames for PreTraining. As shown in Tab. 5, the results show that as the number of pretraining frames increases, the videos generated by Video-GPT are more and more in line with physical laws. This reveals that as the time window of model pretraining increases, the model's modeling of world knowledge is getting better and better.

Add Noise to Clean Clip for Pretraining. During inference, there is a deviation between **DNS** and **CL**. To fill this deviation, we add slight noise to clean frames during training as

$$\Phi(k,i) = (\beta + \gamma_{k,i})\Phi(k,i) + (1 - \beta - \gamma_{k,i})\epsilon_{k,i}, \tag{6}$$

where β and $\gamma_{k,i} \sim$ Uniform[0,1] refer to the basic retention degree and random retention degree of the clean frame respectively and $\epsilon_{k,i}$ is sampled by $\epsilon_{k,i} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. We set β=0.9. As shown in Tab. 5, the result shows that adding noise to the clean clip during training improves model performance.

Dataset Scale for Pretraining. We conduct ablation study on the impact of dataset scale based on OpenVid-1M and Panda-70M. As shown in Tab. 5, the results demonstrate that after expanding the scale of videos for pretraining, Video-GPT's modeling capability for the world significantly improved. This further indicates that Video-GPT has substantial room for improvement, as its pretraining can utilize almost all video data on the internet without annotations, similar to GPT.

4.4 DOWNSTREAM TASK EVALUATION

We fine-tuned Video-GPT on downstream tasks, it demonstrates that our model acquired excellent prior knowledge and modal representations through large-scale self-supervised pretraining with videos. Unless specified, we perform experiments based on the Stage 3 pretraining model in Tab. 1.

Class-to-Video Generation. As shown in Tab. 6, Video-GPT achieves the state-of-the-art performance on UCF-101 at high resolution. without the better 3D VAE. In addition, we test the Video-GPT trained from scratch on UCF-101, and the results show that our pretraining plays a very important role in the downstream task of class-to-video. Qualitative results are presented in Fig. 6.

Text-to-Video Generation. We employ the same text encoder as SDv2 Rombach et al. (2021). We filter OpenVid-1M based on motion and aesthetic scores Lin et al. (2024) and obtain 0.3M pairs for training, which is a relatively small dataset. As shown in Fig. 7, it demonstrates that our Video-GPT can achieve effective cross-modal generation with only a limited amount of text-video pairs.

Image Animation. In Fig. 8, we fine-tune Video-GPT 2K steps on three training sets with no more than 100 videos each, it shows amazing generalization ability on test cases outside the training set.

Video Classification. As shown in Tab. 7, Video-GPT linear probe on UCF-101 surpasses VideoMAEv2 Wang et al. (2023a), a pretrained model commonly used in the field of video understanding.

Video Retrieval. In Tab. 8, Video-GPT achieves impressive zero-shot performance on the MSR-VTT Xu et al. (2016) and surpasses VideoClip Xu et al. (2021) which was trained from scratch.

Video Object Segmentation. We finetune Video-GPT 5K steps on GetIn-1M 1K subset. In Fig. 9, it alse present good generalization performance when transferred to low level understanding task.

5 CONCLUSION

We introduce Video-GPT, a concise yet powerful large video foundation model that unites autoregressive modeling with diffusion through next clip diffusion. Treating each video clip as a word token allows the model to inherit the self-supervised property of GPT while retaining the sharp synthesis quality of diffusion. Pretraining on 70M unlabeled videos yields state-of-the-art accuracy on deterministic Physics-IQ Benchmark and uncertain Kinetics-600 forecasting, and fine-tuning transfers seamlessly to six challenging and diverse downstream tasks including generation and understanding. Future work will explore multi-modal pretraining, reinforcement-driven world interaction.

ACKNOWLEDGMENTS

This work was supported by Guangdong Science and Technology Program (Grant No. 2024TQ08X365).

ETHICS STATEMENT

Our Video-GPT is a concise yet powerful large video foundation model that unites autore- gressive modeling with diffusion through next clip diffusion. To ensure ethical compliance, our training data is carefully selected from public sources and take deliberate measures to minimize potential biases, fully aligning with universal ethical guidelines. We explicitly emphasize that this framework is not intended for misuse in achieving harmful purposes; downstream users are encouraged to adhere to ethical principles when applying the technology. Additionally, all authors declare no conflicts of interest related to this work.

REPEATABILITY STATEMENT

To ensure full reproducibility, we will publicly release all code and data necessary to replicate our experiments. Comprehensive implementation details, including model architecture, hyperparameters, and training methodology, are provided in this paper and its appendix. We are committed to open-sourcing all essential resources to ensure that our findings can be fully verified and built upon by the research community.

REFERENCES

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Hassan Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Singh Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Caio C'esar Teodoro Mendes, Weizhu Chen, Vishrav Chaudhary, Parul Chopra, Allison Del Giorno, Gustavo de Rosa, Matthew Dixon, Ronen Eldan, Dan Iter, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Jamie Huynh, Mojan Javaheripi, Xin Jin, Piero Kauffmann, Nikos Karampatziakis, Dongwoo Kim, Young Jin Kim, Mahoud Khademi, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Chen Liang, Weishung Liu, Eric Lin, Zeqi Lin, Piyush Madan, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Liliang Ren, Corby Rosset, Sambudha Roy, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Xianmin Song, Olatunji Ruwase, Praneetha Vaddamanu, Xin Wang, Rachel Ward, Guanhua Wang, Philipp Andre Witte, Michael Wyatt, Can Xu, Jiahang Xu, Sonali Yadav, Fan Yang, Ziyi Yang, Donghan Yu, Cheng-Yuan Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yunan Zhang, Xiren Zhou, and Yifan Yang. Phi-3 technical report: A highly capable language model locally on your phone. *ArXiv*, abs/2404.14219, 2024.

OpenAI Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haim ing Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Made laine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Benjamin Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Sim'on Posada Fishman, Juston Forte, Is abella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Raphael Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Jo hannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey,

Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Lukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Hendrik Kirchner, Jamie Ryan Kiros, Matthew Knight, Daniel Kokotajlo, Lukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Li, Rachel Lim, Molly Lin, Stephanie Lin, Ma teusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel P. Mossing, Tong Mu, Mira Murati, Oleg Murk, David M'ely, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Ouyang Long, Cullen O'Keefe, Jakub W. Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alexandre Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Pondé de Oliveira Pinto, Michael Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack W. Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario D. Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas A. Tezak, Madeleine Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cer'on Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll L. Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qim ing Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report. 2023.

Yutong Bai, Xinyang Geng, Karttikeya Mangalam, Amir Bar, Alan L. Yuille, Trevor Darrell, Jitendra Malik, and Alexei A. Efros. Sequential modeling enables scalable learning for large vision models. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22861–22872, 2023.

Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1708–1718, 2021.

Omer Bar-Tal, Hila Chefer, Omer Tov, Charles Herrmann, Roni Paiss, Shiran Zada, Ariel Ephrat, Junhwa Hur, Yuanzhen Li, Tomer Michaeli, Oliver Wang, Deqing Sun, Tali Dekel, and Inbar Mosseri. Lumiere: A space-time diffusion model for video generation. In *ACM SIGGRAPH Conference and Exhibition on Computer Graphics and Interactive Techniques in Asia*, 2024.

James Betker, Gabriel Goh, Li Jing, TimBrooks, Jianfeng Wang, Linjie Li, LongOuyang, JuntangZhuang, JoyceLee, YufeiGuo, WesamManassra, PrafullaDhariwal, CaseyChu, YunxinJiao, and Aditya Ramesh. Improving image generation with better captions. URL `https://api.semanticscholar.org/CorpusID:264403242`.

A. Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, and Dominik Lorenz. Stable video diffusion: Scaling latent video diffusion models to large datasets. *ArXiv*, abs/2311.15127, 2023.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Ma teusz Litwin,

Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *ArXiv*, abs/2005.14165, 2020.

Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Bechtle, Feryal M. P. Behbahani, Stephanie Chan, Nicolas Manfred Otto Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando de Freitas, Satinder Singh, and Tim Rocktaschel. Genie: Generative interactive environments. *ArXiv*, abs/2402.15391, 2024.

João Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics-600. *ArXiv*, abs/1808.01340, 2018.

Boyuan Chen, Diego Marti Monso, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *ArXiv*, abs/2407.01392, 2024a.

Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James T. Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *ArXiv*, abs/2310.00426, 2023a.

Tsai-Shien Chen, Aliaksandr Siarohin, Willi Menapace, Ekaterina Deyneka, Hsiang wei Chao, Byung Eun Jeon, Yuwei Fang, Hsin-Ying Lee, Jian Ren, Ming-Hsuan Yang, and S. Tulyakov. Panda-70m: Captioning 70m videos with multiple cross-modality teachers. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13320–13331, 2024b.

Xinyuan Chen, Yaohui Wang, Lingjun Zhang, Shaobin Zhuang, Xin Ma, Jiashuo Yu, Yali Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. Seine: Short-to-long video diffusion model for generative transition and prediction. *ArXiv*, abs/2310.20700, 2023b.

Aidan Clark, Jeff Donahue, and Karen Simonyan. Adversarial video generation on complex datasets. *arXiv: Computer Vision and Pattern Recognition*, 2019.

Google DeepMind. Gemini 2.5 flash, 2025. URL `https://gemini.google.com/`.

Chaorui Deng, Deyao Zhu, Kunchang Li, Shi Guang, and Haoqi Fan. Causal diffusion transformers for generative modeling. *ArXiv*, abs/2412.12095, 2024.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, et al. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. *ArXiv*, abs/2105.05233, 2021.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony S. Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aur'elien Rodriguez, Austen Gregerson, Ava Spataru, Bap tiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cris tian Cantón Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab A. AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriele Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guanglong Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Ju-Qing Jia, Kalyan Vasuden

Alwala, K. Upasani, Kate Plawiak, Keqian Li, Ken-591 neth Heafield, Kevin R. Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuen ley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melissa Hall Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Niko lay Bashlykov, Nikolay Bogoychev, Niladri S. Chatterji, Olivier Duchenne, Onur cCelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasić, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Ro main Sauvestre, Ron nie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sa hana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Chandra Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Vir ginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whit ney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yiqian Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zhengxu Yan, Zhengxing Chen, Zoe Papakipos, Aaditya K. Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adi Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Po-Yao (Bernie) Huang, Beth Loyd, Beto de Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Shang-Wen Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzm'an, Frank J. Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory G. Sizov, Guangyi Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Han Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kaixing(Kai) Wu, U KamHou, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, A Lavender, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pe dro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollár, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Ro-

han Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sung-Bae Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Andrei Poenaru, Vlad T. Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xia Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. The llama 3 herd of models. *ArXiv*, abs/2407.21783, 2024.

Patrick Esser, Sumith Kulal, A. Blattmann, Rahim Entezari, Jonas Muller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. *ArXiv*, abs/2403.03206, 2024.

Songwei Ge, Thomas Hayes, Harry Yang, Xiaoyue Yin, Guan Pang, David Jacobs, Jia-Bin Huang, and Devi Parikh. Long video generation with time-agnostic vqgan and time-sensitive transformer. In *European Conference on Computer Vision*, 2022.

Priya Goyal, Piotr Dollár, Ross B. Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *ArXiv*, abs/1706.02677, 2017.

Yuchao Gu, Weijia Mao, and Mike Zheng Shou. Long-context autoregressive video modeling with next-frame prediction. *ArXiv*, abs/2503.19325, 2025.

Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, et al. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024.

Tengda Han, Weidi Xie, and Andrew Zisserman. Memory-augmented dense predictive coding for video representation learning. In *European Conference on Computer Vision*, 2020.

Jonathan Ho. Classifier-free diffusion guidance. *ArXiv*, abs/2207.12598, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Conference and Workshop on Neural Information Processing Systems*, 2020.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. *ArXiv*, abs/2204.03458, 2022.

Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *ArXiv*, abs/2205.15868, 2022.

Jinyi Hu, Shengding Hu, Yuxuan Song, Yufei Huang, Mingxuan Wang, Hao Zhou, Zhiyuan Liu, Wei-Ying Ma, and Maosong Sun. Acdit: Interpolating autoregressive conditional modeling and diffusion transformer. *ArXiv*, abs/2412.07720, 2024.

Zhihao Hu and Dong Xu. Videocontrolnet: A motion-guided video-to-video translation framework by using diffusion model with controlnet. *ArXiv*, abs/2307.14073, 2023.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *ArXiv*, abs/2506.08009, 2025a. URL `https://api.semanticscholar.org/CorpusID:279251392`.

Zhipeng Huang, Shaobin Zhuang, Canmiao Fu, Binxin Yang, Ying Zhang, Chong Sun, Zhizheng Zhang, Yali Wang, Chen Li, and Zheng-Jun Zha. Wegen: A unified model for interactive multimodal generation as we chat. *ArXiv*, abs/2503.01115, 2025b.

Emmanuel Kahembwe and Subramanian Ramamoorthy. Lower dimensional kernels for video discriminators. *Neural networks : the official journal of the International Neural Network Society*, 132:506–520, 2019.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeff Wu, and Dario Amodei. Scaling laws for neural language models. *ArXiv*, abs/2001.08361, 2020.

Taehoon Kim, Gwangmo Song, Sihaeng Lee, Sangyun Kim, Yewon Seo, Soonyoung Lee, Seung Hwan Kim, Honglak Lee, and Kyunghoon Bae. L-verse: Bidirectional generation between image and text. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16505–16515, 2021.

D. Kondratyuk, Lijun Yu, Xiuye Gu, José Lezama, Jonathan Huang, Rachel Hornung, Hartwig Adam, Hassan Akbari, Yair Alon, Vighnesh Birodkar, Yong Cheng, Ming-Chang Chiu, Josh Dillon, Irfan Essa, Agrim Gupta, Meera Hahn, Anja Hauth, David Hendon, Alonso Martinez, David C. Minnen, David A. Ross, Grant Schindler, Mikhail Sirotenko, Kihyuk Sohn, Krishna Somandepalli, Huisheng Wang, Jimmy Yan, Ming Yang, Xuan Yang, Bryan Seybold, and Lu Jiang. Videopoet: A large language model for zero-shot video generation. *ArXiv*, abs/2312.14125, 2023.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jia-Liang Xiong, Xin Li, Bo Wu, Jianwei Zhang, Kathrina Wu, Qin Lin, Junkun Yuan, Yanxin Long, Aladdin Wang, Andong Wang, Changlin Li, Duojun Huang, Fan Yang, Hao Tan, Hongmei Wang, Jacob Song, Jiawang Bai, Jianbing Wu, Jinbao Xue, Joey Wang, Kai Wang, Mengyang Liu, Peng-Yu Li, Shuai Li, Weiyan Wang, Wenqing Yu, Xi Deng, Yang Li, Yi Chen, Yutao Cui, Yuanbo Peng, Zhen Yu, Zhiyu He, Zhiyong Xu, Zixiang Zhou, Xavier Xu, Yang-Dan Tao, Qinglin Lu, Songtao Liu, Daquan Zhou, Hongfa Wang, Yong Yang, Di Wang, Yuhong Liu, Jie Jiang, and Caesar Zhong. Hunyuanvideo: A systematic framework for large video generative models. *ArXiv*, abs/2412.03603, 2024.

Kuaishou. Kling ai, 2024. URL `https://klingai.kuaishou.com/`.

Black Forest Labs. Flux, 2024. URL `https://blackforestlabs.ai/`.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025.

Chengxuan Li, Di Huang, Zeyu Lu, Yang Xiao, Qingqi Pei, and Lei Bai. A survey on long video generation: Challenges, methods, and prospects. *ArXiv*, abs/2403.16407, 2024.

Chao Liao, Liyang Liu, Xun Wang, Zhengxiong Luo, Xinyu Zhang, Wenliang Zhao, Jie Wu, Liang Li, Zhi Tian, and Weilin Huang. Mogao: An omni foundation model for interleaved multi-modal generation. *ArXiv*, abs/2505.05472, 2025.

Bin Lin, Yunyang Ge, Xinhua Cheng, Zongjian Li, Bin Zhu, Shaodong Wang, Xianyi He, Yang Ye, Shenghai Yuan, Liuhan Chen, Tanghui Jia, Junwu Zhang, Zhenyu Tang, Yatian Pang, Bin She, Cen Yan, Zhiheng Hu, Xiao wen Dong, Lin Chen, Zhang Pan, Xing Zhou, Shaoling Dong, Yonghong Tian, and Li Yuan. Open-sora plan: Open-source large video generation model. *ArXiv*, abs/2412.00131, 2024.

Shanchuan Lin, Ceyuan Yang, Hao He, Jianwen Jiang, Yuxi Ren, Xin Xia, Yang Zhao, Xuefeng Xiao, and Lu Jiang. Autoregressive adversarial post-training for real-time interactive video generation. *ArXiv*, abs/2506.09350, 2025. URL `https://api.semanticscholar.org/CorpusID:279305478`.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *ArXiv*, abs/2210.02747, 2022.

Hao Liu, Wilson Yan, Matei Zaharia, and Pieter Abbeel. World model on million-length video and language with blockwise ringattention. *ArXiv*, abs/2402.08268, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2017.

Zhengxiong Luo, Dayou Chen, Yingya Zhang, Yan Huang, Liangsheng Wang, Yujun Shen, Deli Zhao, Jinren Zhou, and Tien-Ping Tan. Videofusion: Decomposed diffusion models for high-quality video generation. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10209–10218, 2023.

Xin Ma, Yaohui Wang, Gengyun Jia, Xinyuan Chen, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *ArXiv*, abs/2401.03048, 2024.

Willi Menapace, Aliaksandr Siarohin, Ivan Skorokhodov, Ekaterina Deyneka, Tsai-Shien Chen, Anil Kag, Yuwei Fang, Aleksei Stoliar, Elisa Ricci, Jian Ren, and S. Tulyakov. Snap video: Scaled spatiotemporal transformers for text-to-video synthesis. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7038–7048, 2024.

Saman Motamed, Laura Culp, Kevin Swersky, Priyank Jaini, and Robert Geirhos. Do generative video models understand physical principles? *ArXiv*, abs/2501.09038, 2025.

Kepan Nan, Rui Xie, Penghao Zhou, Tiehan Fan, Zhenheng Yang, Zhijie Chen, Xiang Li, Jian Yang, and Ying Tai. Openvid-1m: A large-scale high-quality dataset for text-to-video generation. *ArXiv*, abs/2407.02371, 2024.

OpenAI. Video generation models as world simulators, 2024. URL `https://openai.com/index/video-generation-models-as-world-simulators/`.

Yichen Ouyang, Jianhao Yuan, Hao Zhao, Gaoang Wang, and Bo Zhao. Flexifilm: Long video generation with flexible conditions. *ArXiv*, abs/2404.18620, 2024.

Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam M. Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *International Conference on Machine Learning*, 2018.

Mandela Patrick, Po-Yao (Bernie) Huang, Yuki M. Asano, Florian Metze, Alexander Hauptmann, João F. Henriques, and Andrea Vedaldi. Support-set bottlenecks for video-text representation learning. *ArXiv*, abs/2010.02824, 2020.

William S. Peebles and Saining Xie. Scalable diffusion models with transformers. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182, 2022.

PikaLabs. Pika 1.5, 2024. URL `https://pika.art/`.

Dustin Podell, Zion English, Kyle Lacey, A. Blattmann, Tim Dockhorn, Jonas Muller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *ArXiv*, abs/2307.01952, 2023.

Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, David Yan, Dhruv Choudhary, Dingkang Wang, Geet Sethi, Guan Pang, Haoyu Ma, Ishan Misra, Ji Hou, Jialiang Wang, Ki ran Jagadeesh, Kunpeng Li, Luxin Zhang, Mannat Singh, Mary Williamson, Matt Le, Matthew Yu, Mitesh Kumar Singh, Peizhao Zhang, Peter Vajda, Quentin Duval, Rohit Girdhar, Roshan Sumbaly, Sai Saketh Rambhatla, Sam S. Tsai, Samaneh Azadi, Samyak Datta, Sanyuan Chen, Sean Bell, Sharadh Ramaswamy, Shelly Sheynin, Siddharth Bhattacharya, Simran Motwani, Tao Xu, Tianhe Li, Tingbo Hou, Wei-Ning Hsu, Xi Yin, Xiaoliang Dai, Yaniv Taigman, Yaqiao Luo, Yen-Cheng Liu, Yi-Chiao Wu, Yue Zhao, Yuval Kirstain, Zecheng He, Zijian He, Albert Pumarola, Ali K. Thabet, Artsiom Sanakoyeu, Arun Mallya, Baishan Guo, Boris Araya, Breena Kerr, Carleigh Wood, Ce Liu, Cen Peng, Dimitry Vengertsev, Edgar Schonfeld, Elliot Blanchard, Felix Juefei-Xu, Fraylie Nord, Jeff Liang, John Hoffman, Jonas Kohler, Kaolin Fire, Karthik Sivakumar, Lawrence Chen, Licheng Yu, Luya Gao, Markos Georgopoulos, Rashel Moritz, Sara K. Sampson, Shikai Li, Simone Parmeggiani, Steve Fine, Tara Fowler, Vladan Petrovic, and Yuming Du. Movie gen: A cast of media foundation models. *ArXiv*, abs/2410.13720, 2024.

Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. 2018.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *ArXiv*, abs/2204.06125, 2022.

Zhongwei Ren, Yunchao Wei, Xun Guo, Yao Zhao, Bingyi Kang, Jiashi Feng, and Xiaojie Jin. Videoworld: Exploring knowledge learning from unlabeled videos. *ArXiv*, abs/2501.09781, 2025.

Robin Rombach, A. Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10674–10685, 2021.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *ArXiv*, abs/1505.04597, 2015.

Runway. Gen-3, 2024. URL `https://runwayml.com/`.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, Seyedeh Sara Mahdavi, Raphael Gontijo Lopes, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. *ArXiv*, abs/2205.11487, 2022.

Masaki Saito, Shunta Saito, Masanori Koyama, and Sosuke Kobayashi. Train sparsely, generate densely: Memory-efficient unsupervised training of high-resolution temporal gan. *International Journal of Computer Vision*, 128:2586 – 2606, 2018.

Eyal Segalis, Dani Valevski, Danny Lumen, Yossi Matias, and Yaniv Leviathan. A picture is worth a thousand words: Principled recaptioning improves image generation. *ArXiv*, abs/2310.16656, 2023.

Uriel Singer, Adam Polyak, Thomas Hayes, Xiaoyue Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-a-video: Text-to-video generation without text-video data. *ArXiv*, abs/2209.14792, 2022.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Machine Learning*, 2021.

Khurram Soomro, Amir Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *ArXiv*, abs/1212.0402, 2012.

Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *ArXiv*, abs/2406.06525, 2024.

Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *ArXiv*, abs/2405.09818, 2024.

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *ArXiv*, abs/1812.01717, 2018.

Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Xiaofeng Meng, Ningying Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Rui Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wen-Chao Zhou,

Wente Wang, Wen Shen, Wenyuan Yu, Xianzhong Shi, Xiaomin Huang, Xin Xu, Yan Kou, Yan-Mei Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhengbin Han, Zhigang Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *ArXiv*, abs/2503.20314, 2025.

Hanyu Wang, Saksham Suri, Yixuan Ren, Hao Chen, and Abhinav Shrivastava. Larp: Tokenizing videos with a learned autoregressive generative prior. *ArXiv*, abs/2410.21264, 2024a.

Junke Wang, Yi Jiang, Zehuan Yuan, Binyue Peng, Zuxuan Wu, and Yu-Gang Jiang. Omnitokenizer: A joint image-video tokenizer for visual generation. *ArXiv*, abs/2406.09399, 2024b.

Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. Videomae v2: Scaling video masked autoencoders with dual masking. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14549–14560, 2023a.

Qiuheng Wang, Yukai Shi, Jiarong Ou, Rui Chen, Ke Lin, Jiahao Wang, Boyuan Jiang, Haotian Yang, Mingwu Zheng, Xin Tao, Fei Yang, Pengfei Wan, and Di Zhang. Koala-36m: A large-scale video dataset improving consistency between fine-grained conditions and video content. *ArXiv*, abs/2410.08260, 2024c.

Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, Yingli Zhao, Yulong Ao, Xuebin Min, Tao Li, Boya Wu, Bo Zhao, Bowen Zhang, Lian zi Wang, Guang Liu, Zheqi He, Xi Yang, Jingjing Liu, Yonghua Lin, Tiejun Huang, and Zhongyuan Wang. Emu3: Next-token prediction is all you need. *ArXiv*, abs/2409.18869, 2024d.

Yaohui Wang, Xinyuan Chen, Xin Ma, Shangchen Zhou, Ziqi Huang, Yi Wang, Ceyuan Yang, Yinan He, Jiashuo Yu, Pe der Yang, Yuwei Guo, Tianxing Wu, Chenyang Si, Yuming Jiang, Cunjian Chen, Chen Change Loy, Bo Dai, Dahua Lin, Y. Qiao, and Ziwei Liu. Lavie: High-quality video generation with cascaded latent diffusion models. *ArXiv*, abs/2309.15103, 2023b.

Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Jian Ma, Xinyuan Chen, Yaohui Wang, Ping Luo, Ziwei Liu, Yali Wang, Limin Wang, and Y. Qiao. Internvid: A large-scale video-text dataset for multimodal understanding and generation. *ArXiv*, abs/2307.06942, 2023c.

Shitao Xiao, Yueze Wang, Junjie Zhou, Huaying Yuan, Xingrun Xing, Ruiran Yan, Shuting Wang, Tiejun Huang, and Zheng Liu. Omnigen: Unified image generation. *ArXiv*, abs/2409.11340, 2024.

Hu Xu, Gargi Ghosh, Po-Yao (Bernie) Huang, Dmytro Okhonko, Armen Aghajanyan, and Florian Metze Luke Zettlemoyer Christoph Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding. In *Conference on Empirical Methods in Natural Language Processing*, 2021.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5288–5296, 2016.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mt5: A massively multilingual pre-trained text-to-text transformer. In *North American Chapter of the Association for Computational Linguistics*, 2020.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Ke-Yang Chen, Kexin Yang, Mei Li, Min Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yunyang Wan, Yunfei Chu, Zeyu Cui, Zhenru Zhang, and Zhi-Wei Fan. Qwen2 technical report. *ArXiv*, abs/2407.10671, 2024a.

Sherry Yang, Jacob Walker, Jack Parker-Holder, Yilun Du, Jake Bruce, Andre Barreto, Pieter Abbeel, and Dale Schuurmans. Video as the new language for real-world decision making. *ArXiv*, abs/2402.17139, 2024b.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Xiaotao Gu, Yuxuan Zhang, Weihan Wang, Yean Cheng, Ting Liu, Bin Xu, Yuxiao Dong, and Jie Tang. Cogvideox: Text-to-video diffusion models with an expert transformer. *ArXiv*, abs/2408.06072, 2024c.

Hongwei Yi, Shitong Shao, Tian Ye, Jiantong Zhao, Qingyu Yin, Michael Lingelbach, Li Yuan, Yonghong Tian, Enze Xie, and Daquan Zhou. Magic 1-for-1: Generating one minute video clips within one minute. *ArXiv*, abs/2502.07701, 2025.

Tianwei Yin, Qiang Zhang, Richard Zhang, William T. Freeman, Frédo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast causal video generators. *ArXiv*, abs/2412.07772, 2024.

Lijun Yu, José Lezama, Nitesh Bharadwaj Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Agrim Gupta, Xiuye Gu, Alexander G. Hauptmann, Boqing Gong, Ming-Hsuan Yang, Irfan Essa, David A. Ross, and Lu Jiang. Language model beats diffusion - tokenizer is key to visual generation. In *International Conference on Learning Representations*, 2024.

Zhengrong Yue, Shaobin Zhuang, Kunchang Li, Yanbo Ding, and Yali Wang. V-stylist: Video stylization via collaboration and reflection of mllm agents. *ArXiv*, abs/2503.12077, 2025.

Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model. *ArXiv*, abs/2408.11039, 2024.

Shaobin Zhuang, Kunchang Li, Xinyuan Chen, Yaohui Wang, Ziwei Liu, Yu Qiao, and Yali Wang. Vlogger: Make your dream a vlog. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8806–8817, 2024.

Shaobin Zhuang, Zhipeng Huang, Binxin Yang, Ying Zhang, Fangyikang Wang, Canmiao Fu, Chong Sun, Zheng-Jun Zha, Chen Li, and Yali Wang. Get in video: Add anything you want to the video. *ArXiv*, abs/2503.06268, 2025.

Algorithm 1 Video-GPT Pre-training

Input:
V: The set of all training videos.
M_θ: Video-GPT model with parameters θ.
VAE_E: The pre-trained VAE Encoder.
T: Total number of training steps.

Main Training Loop:
for $step = 1 \rightarrow T$ **do**
  // 1. Input Sequence Construction (Sec. 3.1)
  $v \leftarrow$ sample_video(V) ▷ Sample a video from the training set
  all_frames $\leftarrow$ sample_n_frames(v) ▷ Sample n frames from the video
  clips $\leftarrow$ partition_into_k_clips(all_frames) ▷ K is randomly sampled

  $S_{\text{input}} \leftarrow []$ ▷ Initialize full input sequence for the model
  $S_{\text{target}} \leftarrow []$ ▷ Initialize ground truth clean latents for loss

  for $k = 1 \rightarrow$ length(clips) **do**
    $C_k \leftarrow$ clips$[k]$
    $\Phi_k \leftarrow \text{VAE}_E(C_k)$ ▷ Encode clean clip to latent space Φ_k

  // (Optional) Add slight noise for training robustness (Sec. 4.3, Eq. 6)
    $\Phi_{k,\text{noisy}} \leftarrow$ add_slight_noise(Φ_k)

  // Create a noisy version via forward diffusion (flow matching)
    $\alpha_k \leftarrow$ sample_from_uniform$(0, 1)$ ▷ Sample a single noise level for the clip
    $\varepsilon_k \sim \mathcal{N}(0, I)$ with shape of Φ_k ▷ Sample from a standard Gaussian
    $\Psi_k \leftarrow \alpha_k \Phi_k + (1 - \alpha_k)\varepsilon_k$ ▷ Noisy latent features

  // Format clips into token sequences
    $\text{NS}_k \leftarrow$ format_noisy_sequence(Ψ_k, α_k) ▷ e.g., $[< \text{diff} >, \alpha_k, \Psi_{k,1}, \ldots]$
    $\text{CL}_k \leftarrow$ format_clean_sequence$(\Phi_{k,\text{noisy}})$ ▷ e.g., $[< \text{img} >, \Phi_{k,1}, < \backslash\text{img} >, \ldots]$

    Append NS_k and CL_k to S_{input}
    Append Φ_k to S_{target} ▷ Target for loss is the original clean latent
  end for

  // 2. Pre-training with Next Clip Diffusion (Sec. 3.2)
  // Hierarchical attention mask is applied internally by M_θ
  predicted_$\Phi \leftarrow M_\theta(S_{\text{input}})$ ▷ Model forward pass

  // 3. Loss Calculation and Optimization
  // L2 loss is calculated only on outputs corresponding to noisy inputs
  $\mathcal{L} \leftarrow$ l2_loss(predicted_Φ, S_{target})
  $\mathcal{L}$.backward()
  optimizer.step()
  optimizer.zero_grad()
end for

A PSEUDOCODE OF VIDEO-GPT

A.1 TRAINING PSEUDOCODE

As shown in Algorithm 1.

Algorithm 2 Video-GPT Autoregressive Inference

Input:
M_θ: Pre-trained Video-GPT model.
$\text{VAE}_E, \text{VAE}_D$: VAE Encoder and Decoder.
C_1: Initial conditioning video clip (the first clip).
N: Total number of clips to generate.
c: Classifier-Free Guidance scale (Sec. 4.3).
T: Total number of denoising steps.

// 1. Initialization GenerateVideo(C_1, N, c, T)
$DNS_1 \leftarrow \text{VAE}_E(C_1)$ ▷ Encode the initial clip to a Denoised Sequence.
$Generated_Latents \leftarrow [DNS_1]$ ▷ Initialize list of generated latent clips.

// 2. Autoregressive Generation Loop (Sec. 3.3)
for $k = 1$ **to** $N - 1$ **do**
    $NS_{\text{next}} \leftarrow$ sample_from_gaussian(shape(DNS_1)) ▷ Prepare noisy input for clip $k + 1$.

    // 3. T steps denoising
    for $t = 1$ **to** T **do**
        $Context \leftarrow Generated_Latents$ ▷ Clean latents as context.

        // 4. Denoise using History Conditioned Classifier-Free Guidance
        $input_{\text{cond}} \leftarrow$ create_input_sequence($Context, NS_{\text{next}}$)
        $pred_{\text{cond}} \leftarrow M_\theta(input_{\text{cond}})$ ▷ Conditional prediction (with history).
        $input_{\text{uncond}} \leftarrow$ create_input_sequence($\emptyset, NS_{\text{next}}$)
        $pred_{\text{uncond}} \leftarrow M_\theta(input_{\text{uncond}})$ ▷ Unconditional prediction (null context).
        $cfg_{\text{cond}} \leftarrow pred_{\text{uncond}} + c \cdot (pred_{\text{cond}} - pred_{\text{uncond}})$ ▷ Predictions with CFG.
        $NS_{\text{next}} \leftarrow$ Flow_Matching(cfg_{cond}, t) ▷ Update noisy latent via flow matching.
    end for
    APPEND($Generated_Latents, NS_{\text{next}}$) ▷ Append the newly generated clean latent clip.
end for

// 5. Decode the final video
$Final_Video_Clips \leftarrow []$
for each $latent_clip$ **in** $Generated_Latents$ **do**
    $video_clip \leftarrow \text{VAE}_D(latent_clip)$
    APPEND($Final_Video_Clips, video_clip$)
end for
$Final_Video \leftarrow$ concatenate($Final_Video_Clips$)
return $Final_Video$

A.2 Inference Pseudocode

As shown in Algorithm 2.

A.3 Uniform Sampling Pseudocode

As shown in Algorithm 3.

B More Implementation Details

As shown in Tab. 9, 10, 11, 12, 13, 14, 15, 16, 17 and 18.

C More Visualization Results

The full mask of the input in Fig. 2 (a) is shown in Fig. 10.

Table 9: **Pretraining Stage 1 setting.**

config	Panda-70M
optimizer	AdamW Loshchilov & Hutter (2017)
optimizer momentum	β_1, β_2=0.9, 0.95
weight decay	0.1
learning rate schedule	consistent
learning rate	1e-4
warmup steps Goyal et al. (2017)	1000
total steps	300000
input frame	16
resulution	flexible
longest side	320
frame interval	4
video clip	16
computing unit	320

Table 10: **Pretraining Stage 2 setting.**

config	Panda-70M
optimizer	AdamW Loshchilov & Hutter (2017)
optimizer momentum	β_1, β_2=0.9, 0.95
weight decay	0.1
learning rate schedule	consistent
learning rate	1e-4
warmup steps Goyal et al. (2017)	1000
total steps	25000
input frame	48
resulution	flexible
longest side	320
frame interval	4
video clip	$\sim$ Uniform$\{2, 3, \cdots, 48\}$
computing unit	320

Table 11: **Pretraining Stage 3 setting.**

config	Panda-70M
optimizer	AdamW Loshchilov & Hutter (2017)
optimizer momentum	β_1, β_2=0.9, 0.95
weight decay	0.1
learning rate schedule	consistent
learning rate	1e-4
warmup steps Goyal et al. (2017)	1000
total steps	40000
input frame	48
resulution	flexible
longest side	320
frame interval	$\sim$ Uniform$\{4, 5, \cdots, 12\}$
video clip	$\sim$ Uniform$\{2, 3, \cdots, 48\}$
computing unit	320

Table 12: **Pretraining Stage 4 setting.**

config	Panda-70M
optimizer	AdamW Loshchilov & Hutter (2017)
optimizer momentum	β_1, β_2=0.9, 0.95
weight decay	0.1
learning rate schedule	consistent
learning rate	2e-5
warmup steps Goyal et al. (2017)	1000
total steps	20000
input frame	80
resulution	flexible
longest side	320
frame interval	$\sim$ Uniform$\{4, 5, \cdots, 12\}$
video clip	$\sim$ Uniform$\{2, 3, \cdots, 80\}$
computing unit	320

Table 13: **Class to video fine-tuning on UCF-101 setting.**

config	UCF-101
optimizer	AdamW Loshchilov & Hutter (2017)
optimizer momentum	β_1, β_2=0.9, 0.95
weight decay	0.1
learning rate schedule	consistent
learning rate	1e-4
warmup steps Goyal et al. (2017)	1000
total steps	165000
input frame	16
resulution	240×320
longest side	320
frame interval	1
video clip	1
computing unit	64

Table 14: **Text to video fine-tuning on OpenVid-0.3M setting.**

config	OpenVid-0.3M
optimizer	AdamW Loshchilov & Hutter (2017)
optimizer momentum	β_1, β_2=0.9, 0.95
weight decay	0.1
learning rate schedule	consistent
learning rate	1e-4
warmup steps Goyal et al. (2017)	1000
total steps	70000
input frame	24
resulution	flexible
longest side	320
frame interval	4
video clip	1
computing unit	128

Table 15: **Image animation fine-tuning on animation dataset setting.**

config	Anim. Dataset
optimizer	AdamW Loshchilov & Hutter (2017)
optimizer momentum	β_1, β_2=0.9, 0.95
weight decay	0.1
learning rate schedule	consistent
learning rate	1e-4
warmup steps Goyal et al. (2017)	1000
total steps	2000
input frame	48
resulution	flexible
longest side	320
frame interval	flexible
video clip	2
computing unit	8

Table 16: **Video classification fine-tuning on UCF-101 setting.**

config	UCF-101
optimizer	AdamW Loshchilov & Hutter (2017)
optimizer momentum	β_1, β_2=0.9, 0.95
weight decay	0.1
learning rate schedule	consistent
learning rate	1e-4
warmup steps Goyal et al. (2017)	1000
total steps	10000
input frame	16
resulution	224×224
longest side	224
frame interval	1
video clip	1
computing unit	8

Table 17: **Video retrieval fine-tuning on Panda-70M.**

config	Panda-70M
optimizer	AdamW Loshchilov & Hutter (2017)
optimizer momentum	β_1, β_2=0.9, 0.95
weight decay	0.1
learning rate schedule	consistent
learning rate	1e-4
warmup steps Goyal et al. (2017)	1000
total steps	200000
input frame	8
resulution	224×224
longest side	224
frame interval	flexible
video clip	1
computing unit	64

Table 18: **Video object segmenration fine-tuning on GetIn-1K setting.**

config	GetIn-1K
optimizer	AdamW Loshchilov & Hutter (2017)
optimizer momentum	β_1, β_2=0.9, 0.95
weight decay	0.1
learning rate schedule	consistent
learning rate	1e-4
warmup steps Goyal et al. (2017)	1000
total steps	5000
input frame	48
resulution	flexible
longest side	224
frame interval	flexible
video clip	2
computing unit	8

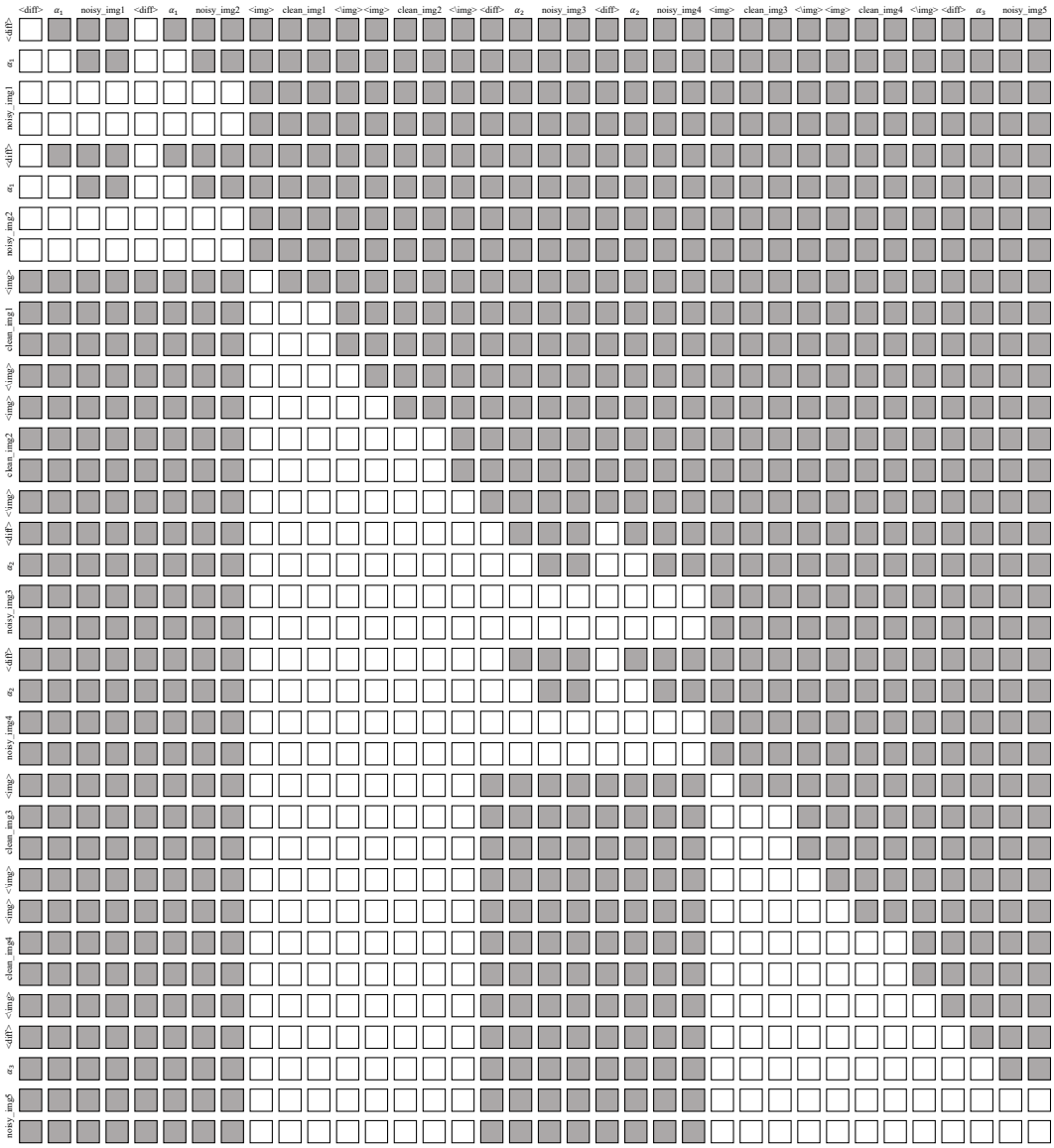


Figure 10: **The full attention mask of Fig. 2 (a) training input.**

Algorithm 3 GenerateRandomList

Input:
num_frames: Total number of frames (integer, ≥ 1).
Output:
$parts$: A list of positive integers that sum to num_frames.

// 1. Initialization
$parts \leftarrow []$ ▷ Initialize the output list.
$previous \leftarrow 0$ ▷ Track the last cut position (start at 0).

// 2. Decide the number of segments internally (no extra input)
$k \leftarrow$ random_integer$(1, num_frames)$ ▷ Number of parts; chosen uniformly to mimic sampling logic.
$m \leftarrow k - 1$ ▷ Number of cut points to sample; $m = 0$ yields no cuts.

// 3. Sample and sort cut positions from the valid range
$cuts \leftarrow$ sorted(random_sample(range$(1, num_frames), m)$) ▷ Distinct cuts in $\{1, \ldots, num_frames - 1\}$; sorted ascending.

// 4. Build parts by consecutive differences
for each cut **in** $cuts$ **do**
  APPEND$(parts, cut - previous)$ ▷ Length of the segment between $previous$ and cut.
  $previous \leftarrow cut$ ▷ Advance the cursor to the current cut.
end for
APPEND$(parts, num_frames - previous)$ ▷ Append the final segment from the last cut to num_frames.

return $parts$

We show the results of Video-GPT for zero-shot long video prediction in Fig. 11 and 12.

In order to demonstrate the capabilities of Video-GPT more comprehensively, we test Video-GPT's prediction ability for videos with partially static clips (minimal motion) and partially with abrupt motion. As can be seen from Fig. 13, Video-GPT is able to handle such extreme video cases. Although there is still much room for improvement, we believe that Video-GPT has indeed learned highly generalizable world knowledge from massive amounts of video data.

As can be seen from Fig. 14, we employ Video-GPT prediction to generate a 1-minute long video. Experiments have shown that Video-GPT can generate videos up to 1-minute long while maintaining temporal coherence and content consistency. As shown in Fig. 15, we generate 1-minute videos using Open-Sora-Plan Lin et al. (2024). From the comparison between Fig. 14 and 15, we can find that the final images of the video generated by Open-Sora-Plan has been significantly collapsed, while the image generated by Video-GPT is still relatively clear. This comparison once again verifies the strong temporal robustness of our Video-GPT.

D LIMITATIONS

Our model exhibits remarkable performance and generalization across diverse tasks, affirming the strength of the next clip diffusion paradigm. It is important to note, however, that our experiments are conducted with a moderately scaled model due to available computational resources. This choice, while pragmatic, has not detracted from the quality of our good results. Looking forward, further scaling may reveal additional performance gains. Nonetheless, the current configuration already sets a high benchmark, underscoring both the efficacy and robustness of our approach in video generative modeling.

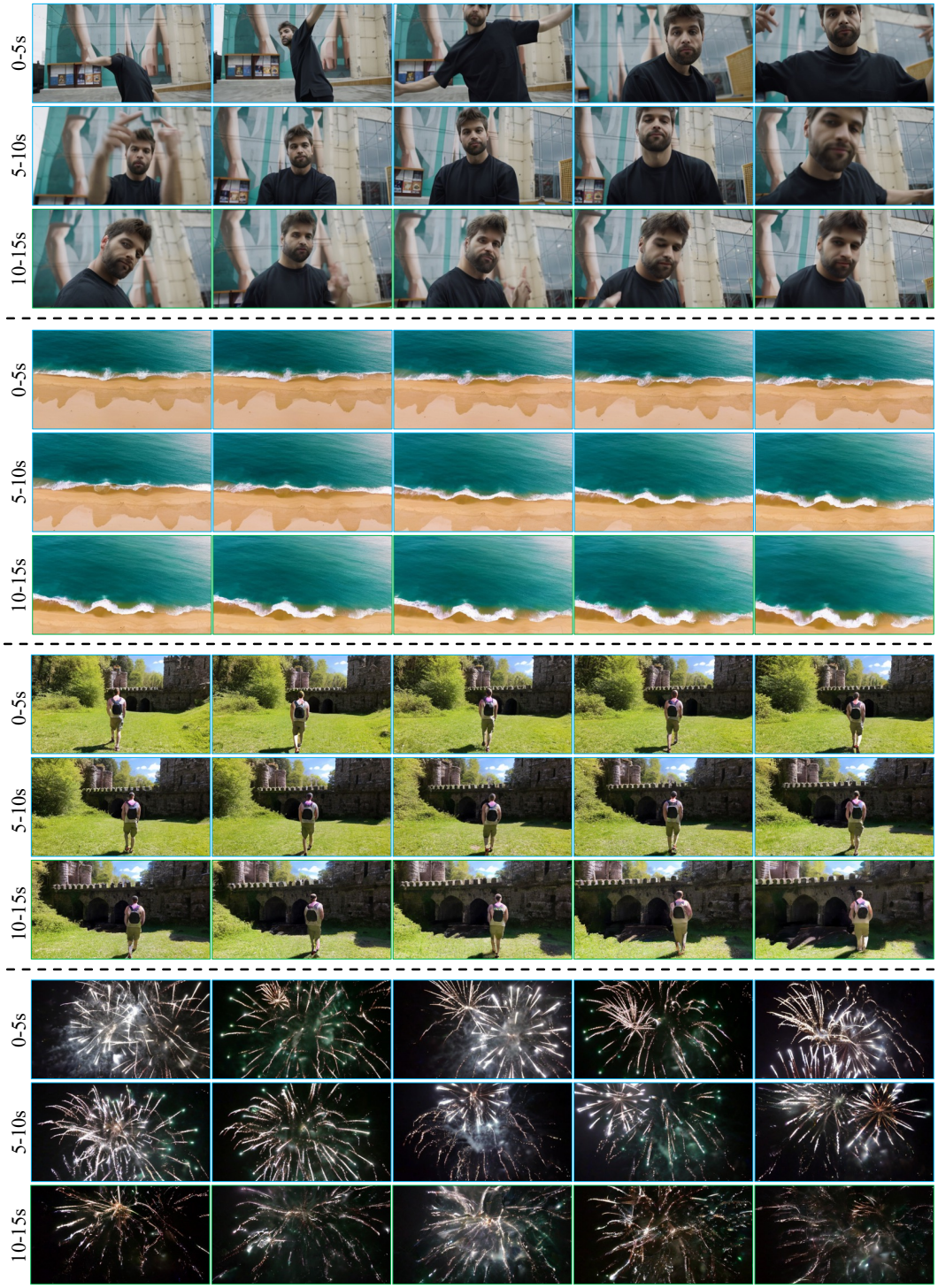


Figure 11: **The long video zero-shot prediction result generated by Video-GPT.** The frames in blue box are condition frames. Frames in green box are the generated prediction result.

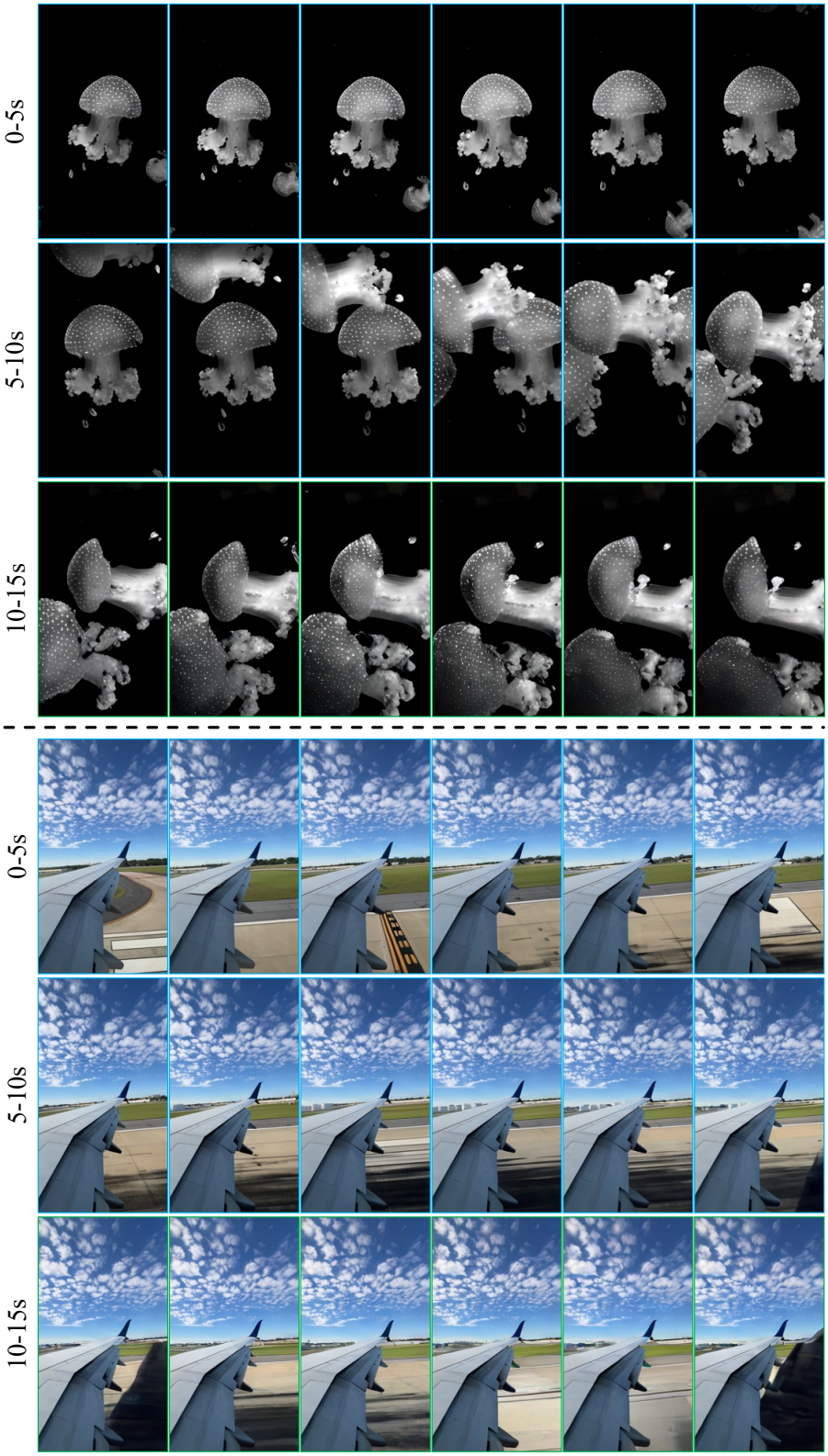


Figure 12: **The long video zero-shot prediction result generated by Video-GPT.** The frames in blue box are condition frames. Frames in green box are the generated prediction result.

Figure 13: **The partly stationary clips (minimal motion) and partly abrupt motion video zero-shot prediction result generated by Video-GPT.**
To test the model's ability to handle extreme cases with partly stationary clips (minimal motion) and partly abrupt motion, we copied an image 11 times to simulate a static video, and then edited the image using FLUX-Kontext Labs et al. (2025). The edited image was then used as the 12th frame of the video to simulate a abrupt, large motion, and Video-GPT (stage4) predicted the next 48 frames. The frames in the orange boxes are given stationary video frames, the red box is abrupt motion frame, and the green boxes are video frames generated by Video-GPT prediction.

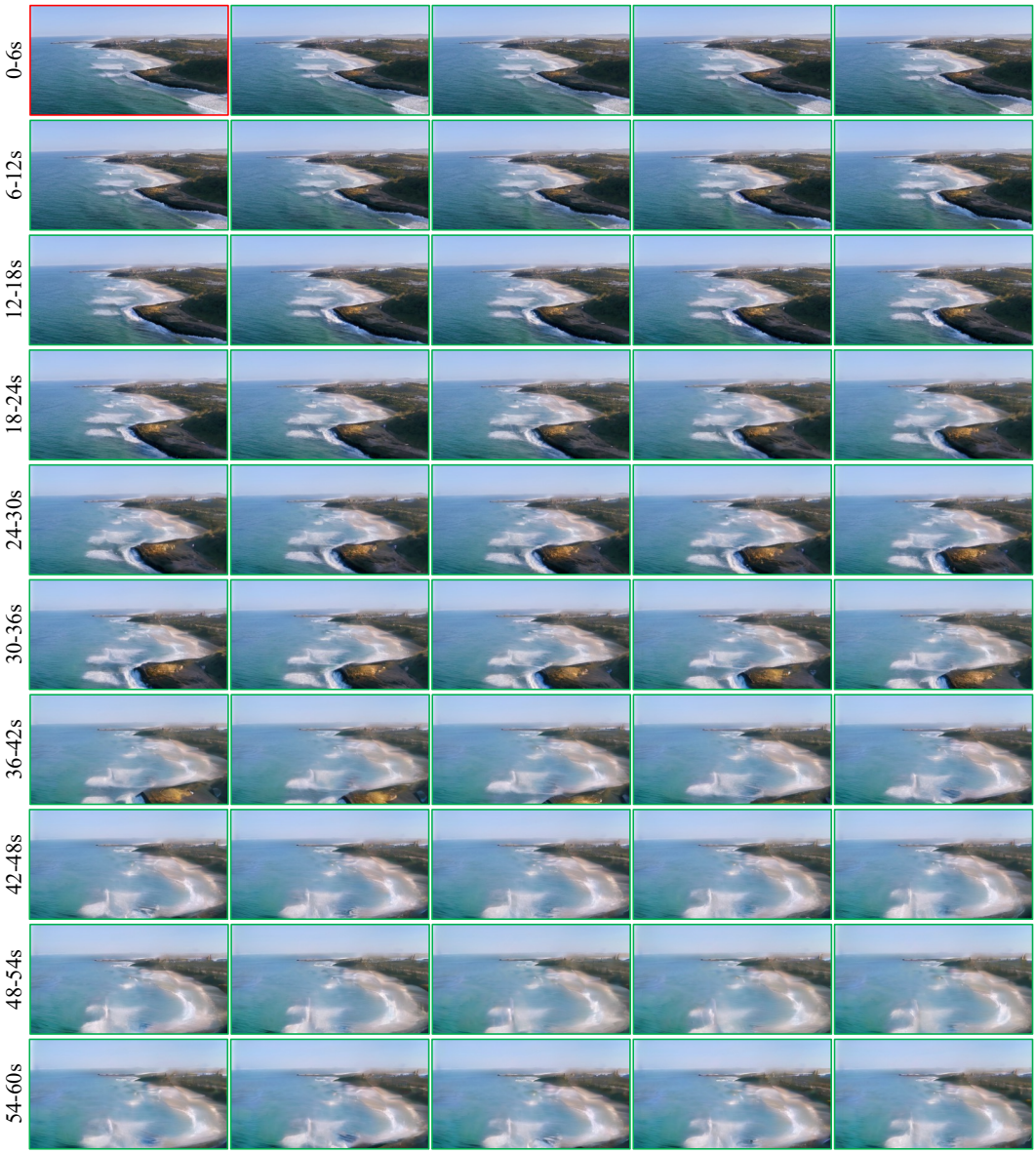


Figure 14: **1-minute video generated by Video-GPT.**
The frame in the red boxes is conditional video frame, the green boxes are video frames generated by Video-GPT prediction.

Figure 15: **1-minute video generated by Open-Sora-Plan Lin et al. (2024).**
The frame in the red box is conditional video frame, the green boxes are video frames predicted by Open-Sora-Plan.

E DECLARATION OF USE OF LARGE LANGUAGE MODELS (LLM)

We affirm that this paper was primarily written by the authors. Large Language Models (LLMs) were utilized solely as general-purpose assistive tools for language refinement, grammar correction, and stylistic improvements during the writing process. Specifically, Gemini 2.5 Flash (DeepMind, 2025) was employed for minor text polishing and rephrasing to enhance clarity and readability. No LLM was used for conceptual ideation, experimental design, data analysis, or generating any substantive content of the research.