VideoPoet: A Large Language Model for Zero-Shot Video Generation

Dan 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 Minnen
David Ross Grant Schindler Mikhail Sirotenko Kihyuk Sohn
Krishna Somandepalli Huisheng Wang Jimmy Yan Ming-Hsuan Yang Xuan Yang
Bryan Seybold* Lu Jiang*

GOOGLE RESEARCH

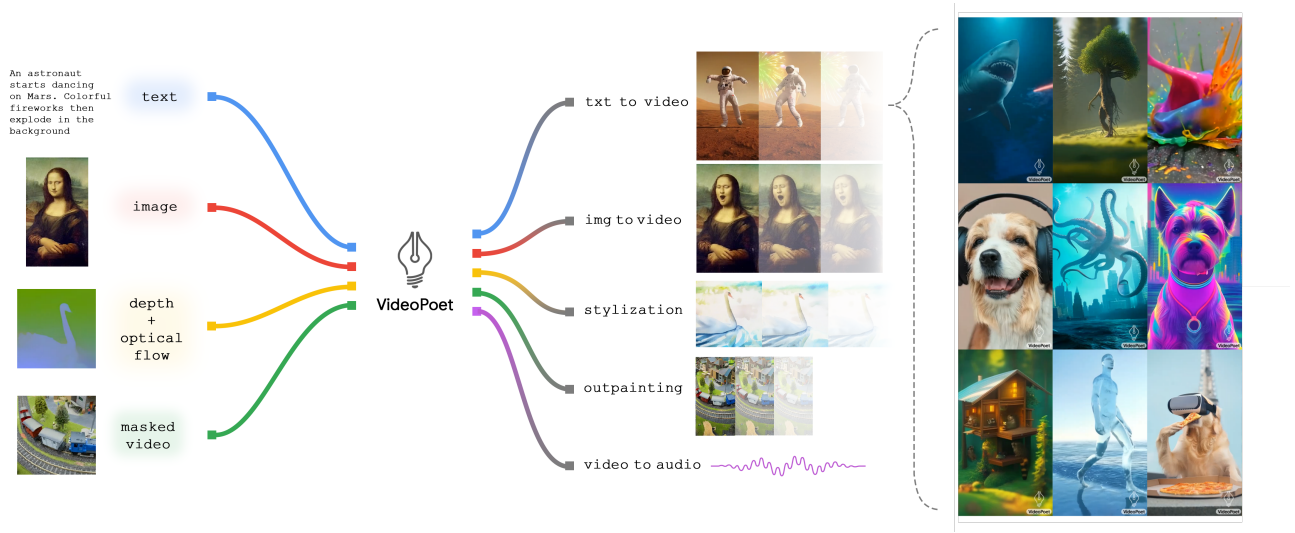


Figure 1. We propose **VideoPoet**, a versatile video generator that conditions on multiple types of inputs and performs a variety of video generation tasks. To see video and audio output from the model, please see http://sites.research.google/videopoet/.

Abstract

We present VideoPoet, a language model capable of synthesizing high-quality video, with matching audio, from a large variety of conditioning signals. VideoPoet employs a decoder-only transformer architecture that processes multimodal inputs – including images, videos, text, and audio. The training protocol follows that of Large Language Models (LLMs), consisting of two stages: pretraining and task-specific adaptation. During pretraining, VideoPoet incorporates a mixture of multimodal generative objectives within an autoregressive Transformer framework. Afterward, the pretrained LLM serves as a foundation that can be adapted for a range of video generation tasks. We present empirical results demonstrating the model's state-of-the-art capabilities in zero-shot video generation, specifically highlighting VideoPoet's ability to generate high-fidelity motions.

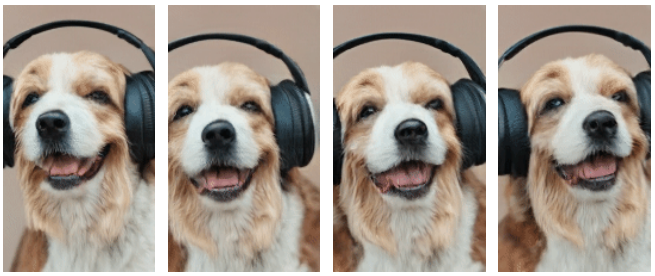
1. Introduction

In recent years there has been a surge of generative video models capable of a variety of video creation tasks. These include text-to-video [58, 67, 82], image-to-video [79], video-to-video stylization [11, 16, 68], and video editing [10, 27, 70] among other video applications. Most existing models employ diffusion-based methods [51] that are often considered the current top performers in video generation. These video models typically start with a pretrained image model, such as Stable Diffusion [46, 51], that pro-

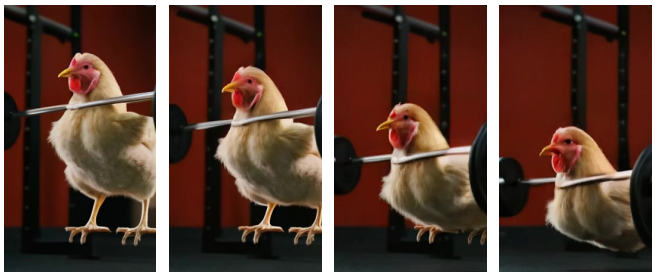
* Equal technical contribution.

Prompt: A photorealistic teddy bear is holding hands with another teddy bear, walking down 5th avenue when it is raining

Prompt: A lion with a mane made out of yellow dandelion petals roars



Prompt: A dog listening to music with headphones, highly detailed, 8k



Prompt: A chicken lifting weights

Prompt: A large blob of exploding splashing rainbow paint, with an apple emerging, 8k

Figure 2. Examples from our text-to-video generation results. Some prompts are edited for clarity.

duces high-fidelity images for individual frames, and then fine-tune the model to improve temporal consistency across video frames.

While Large Language Models (LLMs) are commonly used as foundational models across various modalities including language [3, 8], code [39, 44], audio [52], speech [1], and robotics [21, 87], the diffusion model remains the predominant approach for video generation. Although early research has demonstrated the effectiveness of LLMs in text-to-image generation (*e.g.*, DALL-E [49], Parti [76] among others [20]) and text-to-video (*e.g.*, [37]), language models have not reached a level of quality on par with video diffusion models in tasks like text-to-video generation as shown in previous studies [43, 67, 77]. In contrast to training exclusively for text-to-video tasks, the generative model of LLMs in the language domain emphasizes a large pretraining stage to learn a foundation [6, 61] by examining pretraining tasks that extend beyond text-to-video generation.

A notable advantage of employing LLMs in video generation lies in the ease of integrating existing LLM frameworks. This integration allows for reusing LLM infrastructure and leverages the optimizations our community has developed over many years for LLMs, including optimizations in learning recipes for model scaling [8, 17], training and inference infrastructure [19, 22], hardware, among other innovations. This couples with their flexibility in encoding many diverse tasks in the same model [8, 18, 36, 43, 47], which stands in contrast to most diffusion models where architectural changes and adapter modules are the dominant approach used to adapt the model to more diverse tasks [16, 24, 40, 83].

In this paper, we investigate the application of language models in video generation, following the canonical training protocols of LLMs in the language domain. We introduce *VideoPoet*, a language model for video generation. VideoPoet employs a decoder-only LLM architecture [3, 44, 63] that admits image, video, and audio modalities as discrete tokens, each produced by their respective tokenizer. The training of VideoPoet consists of two stages: (1) pretraining and (2) task-adaptation. During pretraining, VideoPoet incorporates a mixture of multimodal pretraining objectives within an autoregressive transformer framework. After pretraining, the model functions as a versatile multitask video generation model such as text-to-video, image-to-video, video editing and video-to-video stylization, as shown in Figure 1. Unlike [60], these capabilities are inherently integrated into a single LLM, rather than relying on a separate diffusion model controlled by text prompts. During subsequent task-adaptation, the pretrained model can be further fine-tuned either to enhance its generation quality on the training tasks or to perform new tasks.

Our experimental results demonstrate the LLM's state-of-the-art capabilities in generating videos with large and high-fidelity motions. In particular, we observe that through the powerful capabilities of the transformer architecture, the LLM can be straightforwardly trained on a multi-task, multimodal generative objective, allowing for generating consistent and realistic motion driven by text, as shown in Figure 2, or other prompts. We found that the LLM is capable of zero-shot video generation. We use the term "zero-shot video generation" because VideoPoet exhibits generalization capability in processing new text, image, or video inputs that diverge from the training data distribution. Furthermore, VideoPoet begins to show an ability to handle new tasks that were not included in its training. For example, VideoPoet demonstrates the ability to perform new editing tasks by sequentially chaining training tasks together, as elaborated in Section 7.

We provide the following contributions in this work:

- A simple method for training a Large Language Model (LLM) specifically for video generation tasks, utilizing tokenized video and audio data that seamlessly incorporates both text-paired and unpaired video data.
- An approach to super-resolution that increases video resolution within the latent token space using a bidirectional encoder with efficient windowed local attention.
- Evaluations and demonstrations that showcase the LLM's competitive performance, especially in producing realistic and interesting motion.

2. Related Work

Video Diffusion Models Most video generation works use diffusion-based methods for text-to-video [4, 5, 26, 31, 34, 58, 69, 71, 72, 81, 82, 82, 86] and video-to-video editing [16, 24, 25, 40]. Because video diffusion models are most often derived from text-to-image diffusion models [48, 53], additional tasks and modalities are added via inference tricks [42], architectural changes [23, 40] and adapter layers [29, 83]. Although these models are composable after training, they are not trained end-to-end in a unified model. Our multitask pretraining strategy in a single model improves the performance and provides powerful zero-shot capabilities.

Language Models for Video and Image Generation In contrast, video language models are derived from the general family of language models [47, 66] that easily combine multiple tasks in pretraining and demonstrate powerful zero-shot capabilities. Image generation language models can generate images autoregressively [76] or via masked prediction [12, 13]. Both families have been extended to text-to-video [37, 38, 67, 75] using paired data. Other text-to-video work with transformers only leverages video-text pairs for training, but we also leverage unpaired videos and the same video for different tasks. Because video language models can flexibly incorporate many tasks [77, 78], including video-to-video, we extend this family of work to text- and multimodal-conditioned tasks in this work with a synergistic pretraining strategy across many tasks.

Pretraining Task Design in LLMs Because language models so easily incorporate multiple training tasks, task selection is an important area of research. GPT-3 [8] and PaLM [18] show that training LLMs on diverse tasks leads to positive scaling effects on zero- and few-shot tasks. Other works show that masking approaches are a valuable learning target [36, 77, 78]. And as model sizes grow, training data must grow as well [36]. Our pretraining strategy enables using the same video for multiple training tasks even without paired text, so we can effectively increase the number of training examples without access to additional video-text pairs.

3. Model Overview

We are interested in researching an effective method for leveraging large language models for video generation. Our model consists of three main components: (1) modality-specific tokenizers, (2) a language model backbone (Figure 3), and (3) a super-resolution module (Figure 4).

The tokenizers map input data – *i.e.* image pixels, video frames, and audio waveforms – into discrete tokens in a *unified* vocabulary. The visual and audio tokens are flattened into a sequence of integers using raster scan ordering. The LLM accepts image, video and audio tokens as input along with text embeddings, and is responsible for generative multi-task and multimodal modeling. As illustrated in

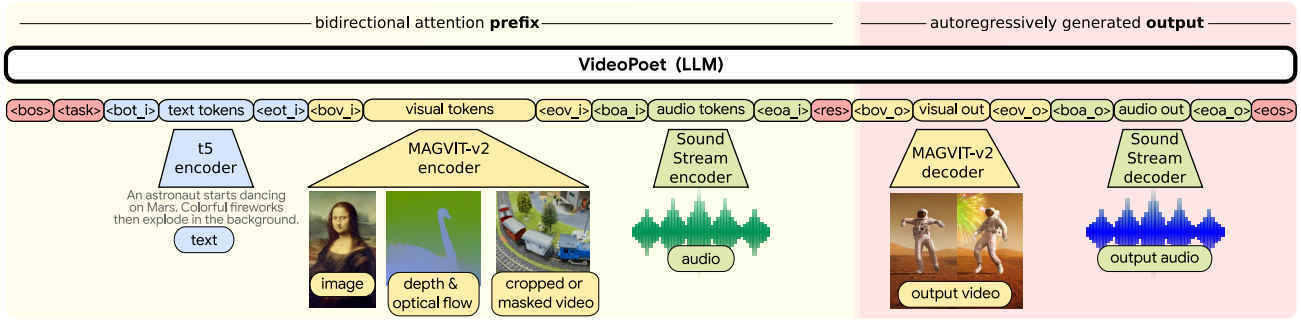


Figure 3. **Sequence layout for VideoPoet**. We encode all modalities into the discrete token space, so that we can directly use large language model architectures for video generation. We denote specital tokens in <>. The modality agnostic tokens are in darker red; the text related components are in blue; the vision related components are in yellow; the audio related components are in green. The left portion of the layout on light yellow represents the bidirectional prefix inputs. The right portion on darker red represents the autoregressively generated outputs with causal attention.

Figure 3, VideoPoet conditions on text embeddings, visual tokens, and audio tokens, and autoregressively predicts visual and audio tokens. Subsequently, the super-resolution module increases the resolution of the video outputs while refining visual details for higher quality. In the following, we discuss design specifics which allow our LLM to generate across video and audio modalities.

3.1. Tokenization

We employ the MAGVIT-v2 [78] tokenizer for joint image and video tokenization, and the SoundStream [80] tokenizer for audio. These visual and audio tokens are represented in a unified vocabulary. The unified vocabulary is constructed as follow: the initial 256 codes are reserved for special tokens and task prompts. Subsequently, the next 262,144 codes are allocated for image and video tokenization. This is followed by 4,096 audio codes. The text modality is represented by text embeddings for its better performance compared with training with text tokens from scratch.

Image and video tokenizer We represent images and videos as a sequence of discrete tokens in a unified vocabulary using the MAGVIT-v2 [78] tokenizer.

Specifically, a video clip is encoded and quantized into a sequence of integers, with a decoder mapping them back to the pixel space. As the bridge between the token and pixel spaces, the performance of this visual tokenizer sets the upper bound of the video generation quality. Meanwhile, the compression ratio determines the sequence length of the LLM for effective and efficient task setups.

MAGVIT-v2 tokenizes 17-frame 2.125-second 128×128 resolution videos sampled at 8 fps to produce a latent shape of $(5, 16, 16)$, which is then flattened into 1280 tokens, with a vocabulary size of 2^{18}. To facilitate the generation of short-form content for mobile, we also tokenize videos into a portrait aspect ratio at 128×224 resolution, producing a latent shape of $(5, 28, 16)$, or 2240 tokens. When the evaluation protocol is on 16 frames, we discard the generated last frame to make a 16-frame video.

The MAGVIT-v2 tokenizer enforces causal temporal dependency, where the frames are encoded without any information from future frames. This causal property simplifies the setup for frame prediction tasks and supports tokenization and generation of arbitrarily long videos. To jointly represent images and videos, we encode the first frame into (1,16,16) tokens, which can be used to represent a static image as well. And then, every 4-frame chunks are encoded into (1,16,16) tokens. These tokens are concatenated on the first (temporal) dimension. For masked objectives, we adopt the COMMIT [77] encoding as input to the tokenizer to optimally setup other tasks such as inpainting and outpainting. In simple terms, COMMIT encoding processes the input condition video and the target video differently to avoid information leakage during tokenization. The former involves tokenization of the conditional video with pixel masks applied, while the latter uses tokenization on the entire unmasked video.

Since the first frame is tokenized separately, MAGVIT-v2 allows images to be represented in the same vocabulary as video. In addition to being more compact, images provide many learnable characteristics that are not typically represented in videos, such as strong visual styles (*e.g.*, art paintings), objects which are infrequently seen in video, rich captions, and significantly more text-image paired training data. When training on images, we resize the images to 128×128 which are then tokenized to a latent shape of $(1, 16, 16)$, or 256 tokens.

We scale the MAGVIT-v2 model's size and train it on the datasets discussed in Section 5.1. The training follows two steps: image training, inflation [78] and video training.

Audio tokenizer We tokenize audio clips with the pretrained SoundStream [80] tokenizer. We embed 2.125 seconds of audio to produce 106 latent frames at a residual vector quantizer (RVQ) of four levels. To improve audio generation performance, we transpose the clip before flattening so that the model predicts the full audio clip at each RVQ granularity level before moving on to the finer grained levels. Finally, each RVQ level has a disjoint vocabulary with each level containing 1,024 codes. This results in a combined audio vocabulary size of 4,096 codes.

Text tokenizer and embedding as input We find that a strong text encoding is important for accurate and high quality text-to-video generation. Pretrained text representations in general outperformed training our model with text tokens from scratch. Due to computational constraints, we found it more efficient to leverage off-the-shelf pretrained language embeddings at our model scale so the model can allocate more capacity to generating and understanding vision and audio modalities.

Therefore, instead of inputting text tokens into the model directly, we first input the tokens into a frozen pretrained T5 XL [47] encoder to produce a sequence of text embeddings. For tasks with text guidance, such as text-to-video, T5 XL embeddings are projected into the transformer's embedding space with a linear layer. We use up to a maximum of 64 text tokens for all of our experiments.

3.2. Language Model Backbone

Now that we have tokenized all modalities into discrete tokens, we can directly reuse a language model to generate videos and audios in the token space. We use a prefix language model with a decoder-only architecture [61] as the backbone. By constructing different patterns of input tokens to output tokens during training, we can control the types of tasks the model is able to perform as explained in Section 4. As explained in Section 3.1, we use a shared multimodal vocabulary to represent the generation of all modalities as a language modeling problem. This produces a total vocabulary size of approximately 300,000.

3.3. Super-Resolution

Generating high-resolution (HR) videos with an autoregressive transformer incurs heavy computational cost due to the increase in sequence length. To illustrate this with an example, the video tokenizer of Section 3.1 operating on a $17 \times 896 \times 512$ video produces a sequence of $35,840$ tokens, making autoregressive sampling highly impractical.

Aiming at efficient and high-quality generative video upsampling, we develop a custom spatial super-resolution (SR) non-autoregressive video transformer [77] to operate in token space on top of the language model output. To mitigate the computational requirements of the very long sequences involved, and in particular the quadratic memory of the self-attention layers, our design incorporates windowed local attention [30]. More precisely, our SR transformer is composed of blocks of three transformer layers, each of which performs self-attention in a local window aligned with one of three axis [64]: *spatial vertical*, *spatial horizontal* and *temporal*. The cross-attention layers attend to the low-resolution (LR) token sequence and are also divided into local windows, isomorphic to those of the self-attention layers. All blocks also include cross-attention to text embeddings from a frozen T5 XL encoder. See Figure 4 for a schematic representation of the custom transformer architecture.

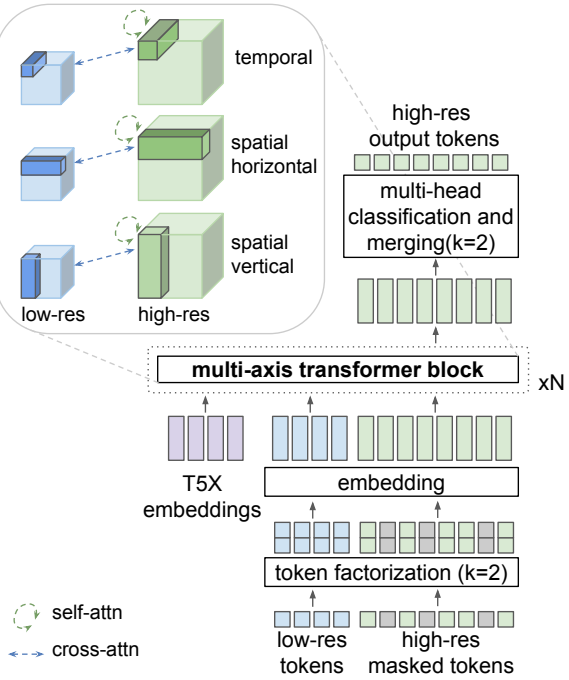


Figure 4. **Architecture for video super-resolution.** We adopt multi-axis attention [30, 64] and masked video modeling [77, 78], conditioned on low-resolution tokens and text embeddings.

We train the SR transformer with the MAGVIT [77] objective, and use token factorization [78] to account for the large vocabulary size. For training, the LR token sequences are obtained by tokenizing bicubic-downsampled versions of the ground truth videos and applying noise augmentation [34] in the discrete latent space. Specifically, we randomly resample the value of a random subset of the LR tokens and independently drop the LR condition and text embeddings for 10% of the training samples. During inference, we use non-autoregressive sampling [12, 77] with classifier-free guidance [7] independently on both the LR condition and the text embeddings. We use a cascade of two $2\times$ stages to generate videos of 896×512 resolution

from the 224×128 samples of VideoPoet. We refer to the appendix for further details on the implementation.

4. LLM Pretraining for Generation

VideoPoet demonstrates general-purpose generative video modeling by training with a large mixture of multimodal objectives. The objectives work together so that individual tasks can be *chained* (see Section 7.1), demonstrating a zero-shot capability that goes beyond any individual task.

4.1. Task Prompt Design

We design a mixture of tasks used in pretraining to produce a foundation model capable of general purpose video generation. For each task we define a prefix input and output such that the model conditions on the prefix, and we only apply the loss on the output.

We define the tasks as follows:

1. **Unconditioned video generation**: generate video frames without conditioning on an input.
2. **Text-to-video**: generate video frames from a text prompt.
3. **Video future prediction**: given an input video of variable length, predict future frames.
4. **Image-to-video**: given the first frame of a video as an input image, predict the future video frames.
5. **Video inpainting and outpainting**: given a masked video, predict the video with the masked contents filled in.
6. **Video stylization**: given a text prompt, optical flow, depth, and optionally the first frame from a video, predict the video frames (see Section 4.1).
7. **Audio-to-video**: given an input audio waveform, predict the corresponding video.
8. **Video-to-audio**: given an input video, predict the corresponding audio waveform.

Below we discuss design decisions within the task prompt design.

Representing an image as a video Producing a high quality initial frame is crucial for the model to generate good video examples, as motion is anchored on the appearance of the first frame. The causal dependencies of frames in the MAGVIT V2 tokenizer allow us to represent any image as if it were the first frame of a video using the same vocabulary. This design enables us to leverage text-image and text-video paired data in joint training, where the image-text data is orders of magnitude larger.

In the input sequence, we leave out the end-of-sequence token (`<eos>`) in the text-to-image pretraining task so the model keeps generating more video tokens at inference time. This setting causes the model not to distinguish between video or image data, further facilitating information sharing between the two modalities. This results in higher quality initial frames being predicted, which significantly reduces the errors and artifacts in future frames.

Video token format For all examples, we apply several variants. We have two resolutions: 128×128 and 128×224. We also generate on two video lengths: 17 frames (2.125 seconds) and 41 frames (5.125 seconds), both at 8 frames per second. We combine the two resolutions and two video lengths, which leads to a total of 4 combinations. Images are a special case of a 1-frame video, which we tokenize at 128×128 resolution. To be able to switch between different resolutions and duration, we use special conditioning tokens that indicate what format of video should be generated.

Video Stylization To perform video stylization, we follow an approach inspired by [15, 24, 83] to predict videos from the combination of text, optical flow, and depth signals. On a subset of steps, we also condition on the first video frame. As described in [24], the text will generally define the "content" or appearance of the output and the optical flow and depth control the "structure." In contrast to the diffusion-based approaches that usually use external cross-attention networks [83] or latent blending [42] for stylization, our approach is more closely related to machine translation using large language models in that we only need to provide the structure and text as a prefix to a language model.

To perform the task, we estimate optical flow from RAFT [62] and produce monocular depth maps from MIDAS [50], and then normalize and concatenate on the channel dimension. This conveniently produces the same number of channels as the RGB ground truth and so can be tokenized in the same fashion as RGB videos with the MAGVIT-v2 tokenizer without retraining the tokenizer. The task of stylization is to reconstruct the ground truth video from the given optical flow, depth, and text information. During inference, we apply optical flow and depth estimation on an input video but then vary the text prompt to generate a new style, *e.g.* "cartoon".

Task layout In Figure 3 we illustrate a typical input-output sequence layout. For each task, the input sequence may include three types of input tokens:

- **text tokens (embeddings)**: the pre-extracted T5 embeddings for any text.
- **visual tokens**: the MAGVIT-v2 tokens representing the images, video subsection, or COMMIT encoded video-to-video task.
- **audio tokens**: the SoundStream tokens representing audio.

Likewise, the model outputs two types of tokens: visual tokens and audio tokens. In addition to video and audio to-

kens along with text embeddings, we incorporate additional *special tokens* enumerated as shown in Table 1.

Special Token	Usage
<bos>	Beginning of sequence
<task_tokens>	Task to perform for this sequence
<bot_i>	Beginning of the text input.
<eot_i>	End of the text input.
<bov_i>	Beginning of the visual input.
<eov_i>	End of the video input.
<boa_i>	Beginning of the audio input.
<eoa_i>	End of the audio input.
<res>	Output resolution for the video.
<bov_o>	Beginning of the video output.
<eov_o>	End of the video output.
<boa_o>	Beginning of the audio output.
<eoa_o>	End of the audio output.
<eos>	End of the entire sequence.

Table 1. **List of representative special tokens used in training and inference**.

When a modality is not included in a task, such as text and audio for unconditioned video generation, then the corresponding input or output tokens together with the beginning and end special tokens are omitted from the sequence to reduce the sequence length. To indicate the type of task, we condition on the <task_tokens>, which has a unique token for each unique output. For the <res> token, the resolution is only specified for 128×224 output, 128×128 resolution is assumed otherwise. We note that changes in the input types do not always require a new <task_tokens>, as the model can learn how to incorporate a mixture of context signals for the same output type. For example, text-to-video, image-to-video, and unconditioned video generation all use the same <task_tokens>.

The video-to-video tasks use the COMMIT encoding [77] to obtain the tokens for the tasks such as inpainting and outpainting. Text is encoded as T5 XL embeddings are inserted into reserved sequence positions just after the BOS token.

4.2. Training Strategy

We train on image-text pairs and video with or without text or audio. Both text and sound are noisy and may not match the visual content. The model is trained on approximately 2 trillion tokens across all modalities.

For multi-task training, we employ accelerated Alternating Gradient Descent (AGD) as formulated in [2] to efficiently train on variable sequence lengths. While packing [47] is an alternative, AGD results in a near 0% padding ratio, providing optimal per-token loss efficiency [2]. We accomplish this by grouping each task by sequence length, and alternately sample one group at each iteration. Because sequence lengths are fixed per task, we can optimally train without any padding. Due to images requiring fewer tokens, we can include roughly 5x more images per batch than videos, *i.e*. 256 image tokens vs. 1280 video tokens.

We find that sampling from image and video datasets uniformly across time can lead to suboptimal results, as training on images can enhance the model's understanding of objects, but does not capture any motions that are represented in video data. As a result, we devise a two-stage pretraining strategy, where we augment our sampling weights to sample from the image data 90% of the time and 10% video for the first 25% iterations of training. We then switch to training on video 90% and image 10% for the rest of training iterations.

After pretraining, we can fine-tune the pretrained model to either improve its performance on specific tasks or to enable it to undertake new tasks. For example, we finetune the model on both text-to-video and image-to-video tasks using a high-quality data subset. We observe improved generation quality which aligns with findings from [85]. Furthermore, we note that the fine-tuned model mitigates the issue of decoding collapse, characterized by the degradation of predictions into repetitive tokens. This improvement not only enhances the model's output diversity but also enables increasing the Classifier-Free Guidance [33] scale, leading to an overall enhancement in quality. In addition, we also finetune the pretrained model to perform video-to-audio generation.

5. Experiments

5.1. Experimental Setup

Training tasks As discussed in Section 4, we train the model on a mixture of text-to-image, text-to-video, image-to-video, and video-to-video tasks—including outpainting, inpainting, stylization, and future frame prediction—as well as video-to-audio, audio-to-video, unconditioned image, and unconditioned video generation. We finetune a model on two tasks—text-to-image and text-to-video—for text-to-video evaluations, and video-to-audio for some examples. Unless explicitly stated, we do not finetune on specific tasks before evaluating.

Datasets We train on a total of 1B image-text pairs and ∼270M videos (over 100M of which include paired text) from the public internet and other sources. The data has been filtered to remove egregious content and sampled to improve contextual and demographic diversity.

Evaluation protocol This paper employs a zero-shot generation evaluation protocol, since the model has not been trained on the data distributions of target benchmarks.

Method	Pretraining Tasks						Zero-shot Evaluation Benchmark				
							T2V		FP	Inpainting	Outpainting
	T2I	T2V	Uncond	FP	Painting	AVCont	MSR-VTT CLIPSIM ↑	UCF101 FVD ↓	K600 FVD ↓	SSV2 FVD ↓	SSV2 FVD ↓
T2V		✓					0.244	822	759	2,333	2,310
T2V+I	✓	✓					0.247	1,025	794	2,118	1,916
SSL			✓	✓	✓	✓	0.226	1,742	700	1,093	1,500
NO T2I		✓	✓	✓	✓	✓	0.235	1,008	755	95	389
ALL	✓	✓	✓	✓	✓	✓	0.240	1,085	729	127	636
Ours (8B)	✓	✓	✓	✓	✓	✓	0.305	355	687	4.7	13.76

Table 2. **Pretraining task analysis on 300M models.** We observe including all pretraining tasks leads to the best overall performance, and scaling up the model to 8B yields significantly better performance. T2I is text-to-image, T2V is text-to-video, FP for frame prediction, Painting denotes both inpainting and outpainting. Uncond for unconditional generation. AVCont is audio-video continuation.

Specifically, the evaluation benchmark includes two text-to-video generation tasks on MSR-VTT [74] and UCF-101 [59], as well as the frame prediction task on Kinetics 600 (k600) [9], in which the first 5 frames are provided as condition to predict the next 11 frames. It also includes inpainting and outpainting tasks [77] on Something-Something V2 (SSV2) [28]. Additionally, we assess stylization tasks using a subset of the DAVIS dataset[1] [45], as further detailed below.

We employ commonly used metrics such as FVD [65], CLIP similarity score [73], and Inception Score (IS) [55] for evaluation. It is important to note that the specific metrics and evaluation methods vary across different datasets. Detailed information on these variations can be found in Appendix A.1.

5.2. Pretraining task analysis

We investigate the learning capabilities of different combinations of pretraining tasks using a model with 300 million parameters. All task combinations are trained using a learning rate of $1e^{-3}$ for the same number of steps (300K) with a batch size of 1024.

For the pretraining tasks, we consider text-to-video (T2V), text-to-image (T2I), and four self-supervised learning (SSL) tasks: frame prediction (FP), central inpainting and central outpainting (Painting) [77] and audio-video continuation (AVCont) where the model is provided with the first frame and its corresponding audio to predict the subsequent 15 frames and their matching audio. We use uniform sampling among the selected tasks, so when fewer tasks are selected, they are trained more extensively. For each video task, we randomly select 20% from a training subset of 50 million videos. Regarding the text-to-image task, we randomly sample 50 million text-image pairs from our training dataset. For tasks involving audios, our sampling is exclusive to videos that contain an audio track.

[1] https://davischallenge.org/

The comparison results are presented in Table 2. We evaluate a model across the four tasks within the zero-shot evaluation benchmark: the text-to-video (T2V) task on MSR-VTT [74] and UCF 101 [59], frame prediction (FP) on Kinetics 600 (K600) [9], as well as inpainting and outpainting on Something-Something V2 (SSV2) [28]. The model is not trained on the data distributions of these evaluation datasets, and thus it's zero-shot evaluation.

The top rows of Table 2 depict each pretraining task configuration of the 300 million parameter model, which are comparable in their setup. We observe that incorporating all pretraining tasks results in the best overall performance, on average, across all evaluated tasks. The last row (Full) represents a model with 8 billion parameters, trained on the pretraining tasks as discussed in Section 3, and utilizing significantly more compute resources. All metrics are significantly improved with this scaling up experiment.

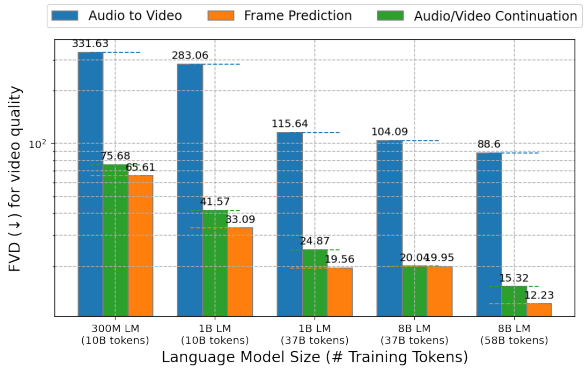
Note that here we do not adapt our pretrained model to finetune on any downstream datasets. This means that for evaluations on individual tasks, our single pretrained model is generalized across all tasks simultaneously in the zero-shot, out-of-domain setting.

5.3. Model scaling

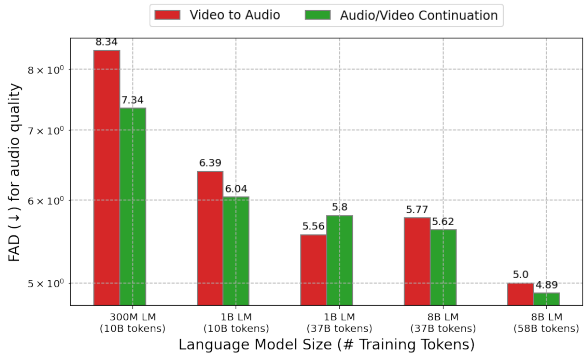
To study model scaling, this experiment uses a subset of the training set without text-paired data and slightly different task prompt design. We evaluate the video generation quality using Fréchet Video Distance (FVD) [65] and audio generation quality using the Fréchet Audio Distance (FAD), which employs the VGGish model as the embedding function [32]. Both FVD and FAD metrics in these model scaling experiments are computed on a subset of our videos with audio, which were held out from training.

Figure 5 shows that as the model size grows and the amount of training data increases, performance improves across visual and audiovisual tasks.

After obtaining the above results, we retrain our 1B



(a) Video generation quality in FVD (↓).



(b) Audio generation quality in FAD (↓).

Figure 5. **Impact of model and data scale on video and audio generation quality.** The performance improves significantly when we scale up the model and training data. Language models with 300 million, 1 billion, and 8 billion parameters are trained on datasets comprising 10, 37, and 58 billion visual and audio tokens, respectively.

and 8B models using the task design and text-paired training data discussed in Section 3. We include a qualitative comparison of our 1B and 8B pretrained models in Appendix A.3. Increasing the model size improved temporal consistency, prompt fidelity, and motion dynamics while adding capabilities for limited text rendering, spatial understanding, and counting.

5.4. Comparison to State of the art

5.4.1 Text-to-video

In Table 3, we conduct zero-shot text-to-video evaluation on the widely used MSR-VTT [74] and UCF-101 [59] datasets. We measure CLIP similarity scores [73] following [67], FVD [65] following [77] for UCF101 and following [82] for MSR-VTT, and Inception Score (IS) [55]. Our model shows very competitive CLIP similarity score and FVD performance on MSR-VTT and UCF-101. We found that our pretrained foundation model already achieves great performance on all metrics. After finetuned on high-quality subset of text-video pairs, VideoPoet achieves better CLIPSIM on MSR-VTT. For more details on the evaluation settings, see Appendix A.1.

Model	MSR-VTT		UCF-101	
	CLIPSIM	FVD	FVD	IS
CogVideo (EN) [37]	0.2631	1294	702	25.27
MagicVideo [86]	-	998	655	-
Video LDM [5]	0.2929	-	551	33.45
ModelScopeT2V [69]	0.2930	550	-	-
InternVid [72]	0.2951	-	617	21.04
VideoFactory [71]	0.3005	-	410	-
Make-A-Video [58]	0.3049	-	367	33.00
Show-1 [82]	0.3072	538	394	35.42
VideoPoet (Pretrain)	0.3049	**213**	**355**	**38.44**
VideoPoet (Task adapt)	**0.3123**	-	-	-

Table 3. **Comparison on zero-shot text-to-video benchmarks.** VideoPoet achieves SOTA performance on MSR-VTT and UCF-101. Different papers use different evaluation protocols, so not all numbers are strictly comparable. See Appendix A.1 for details.

5.4.2 Human evaluations with text-to-video

We analyze VideoPoet using human raters on two tasks: text-to-video and video stylization. For text-to-video we compare VideoPoet with other recently published models, specifically: Show-1 [82], VideoCrafter [14] and Phenaki [67]. Show-1 and VideoCrafter are state-of-the-art publicly available video diffusion models while Phenaki is a token-based approach using the masked token modeling [12].

We first developed a unified evaluation prompt bank consisting of >200 selected prompts covering a variety of categories and styles. A large subset of our prompts are sourced from published prompt sets (including, *e.g.*, Show-1, Video LDM [5]) and the remaining have been manually created for this project. We selected the prompts prior to generating videos and fixed these choices after initial selection. We also selected preferentially for prompts that contain an explicit mention of motion so that the evaluation would not be biased for models that generate high quality videos that are almost still (e.g., "person jumping off of a chair" over "person standing on a chair"). The finetuned model discussed in Section 4.2 was used for the user study.

We then compare each model against VideoPoet in a side-by-side fashion, for each prompt, showing raters videos generated by two models at a time (in randomized order so as to not bias raters). Not all methods generate videos at the same size or aspect ratio, so we resize each video to a fixed area while maintaining its original aspect ratio. Raters are then asked to compare the videos along 5 dimensions and report whether the two videos are similar or one is better than the other. Specifically, we ask raters

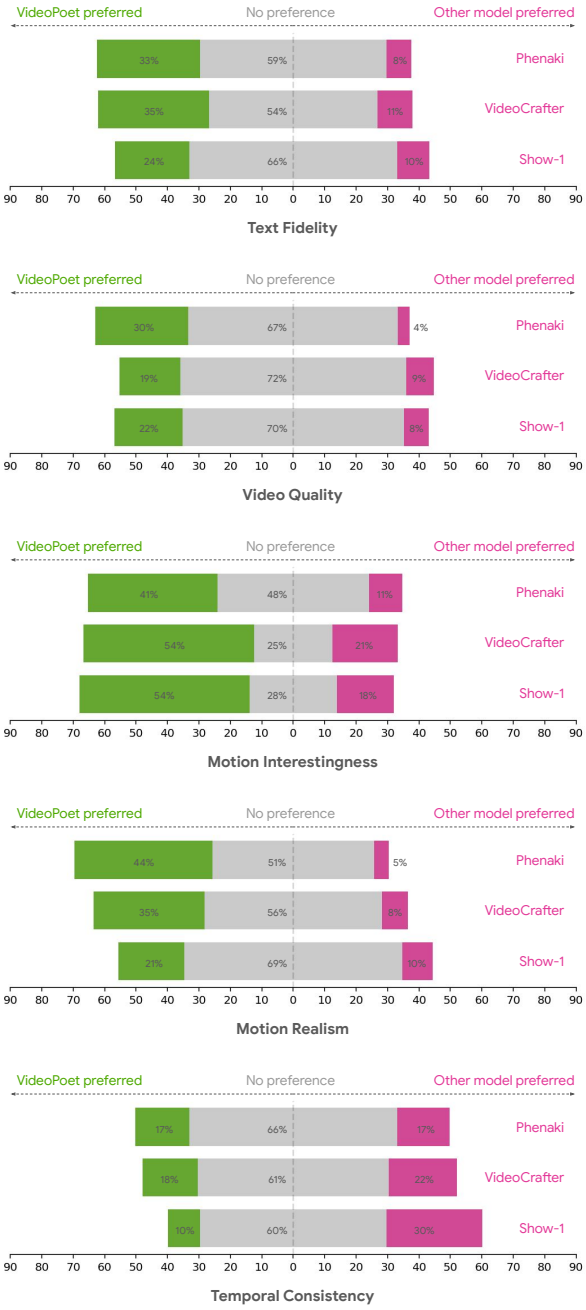


Figure 6. **Human side-by-side evaluations comparing VideoPoet with recent leading text-to-video generative models.** Green, gray, and pink bars represent the proportion of trials where VideoPoet was *preferred over* an alternative, *similar to*, or *less preferred* to an alternative, respectively. The comparison models are Phenaki [67], VideoCrafter [14], and Show-1 [82], respectively. We note that the 'temporal consistency' metric should be interpreted alongside other motion metrics, as it tends to be biased towards static scenes, which inherently display perfect temporal consistency.

to consider (1) text fidelity (which video follows the text prompt most faithfully), (2) video quality, (3) motion "interestingness", (4) motion realism and (5) temporal consistency. Raters are required to undergo a training consisting of going over a collection of "training examples" for each of these five dimensions.

Our findings are summarized in Figure 6, where green, gray, and pink bars represent the proportion of trials where VideoPoet was *preferred over* an alternative, *similar to*, or *less preferred* to an alternative, respectively. Experiments where the green segment is larger than the pink segment mean that VideoPoet is preferred over the alternative on average. Our results show that VideoPoet in general outperforms all baseline models along almost all of the dimensions (text fidelity, quality, motion interestingness and realism) and achieves its most significant wins along the motion categories.

On temporal consistency, VideoPoet shows performance on-par with Phenaki and VideoCrafter but slightly underperforms the Show-1 model. We believe this is due to an inherent trade-off with motion interestingness, *i.e.*, a static scene is more temporally consistent but is less interesting. More interesting larger motions necessitate more possibilities of producing noticable artifacts vs. safer small motions.

5.4.3 Video Stylization

Model	CLIPSIM
Control-A-Video [16][depth]	32.5
VideoPoet (Ours)	34.2

Table 4. **Comparison on video stylization.** VideoPoet outperforms Control-A-Video by a large margin.

To evaluate stylization capabilities, we choose 20 videos from the public DAVIS 2016[2] [45] dataset and provide 2 style prompts for each video. For more details, please refer to Appendix A.4. Following [23], we evaluated the CLIP-embedding consistency between each frame and the text prompt to determine if the stylization results matches the text. As shown in Table 4, VideoPoet outperforms Control-A-Video conditioned on depth by a large margin. We also conduct human evaluations as discussed above comparing with Control-A-Video [16]. Human raters consistently preferr our text fidelity and video quality as shown in Figure 7.

6. Responsible AI and Fairness Analysis

We evaluate whether the generated outputs of our model are fair regarding protected attributes such as (1) Perceived Age

[2]DAVIS license: https://creativecommons.org/licenses/by-nc/4.0/deed.en

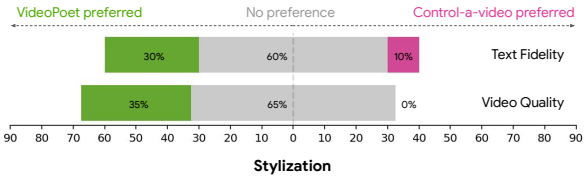


Figure 7. **Human side-by-side evaluations comparing VideoPoet with the video stylization model Control-a-video [16].** Raters prefer VideoPoet on both text fidelity and video quality Green, gray, and pink bars represent the proportion of trials where VideoPoet was preferred over an alternative, similar to, or preferred less than an alternative, respectively.

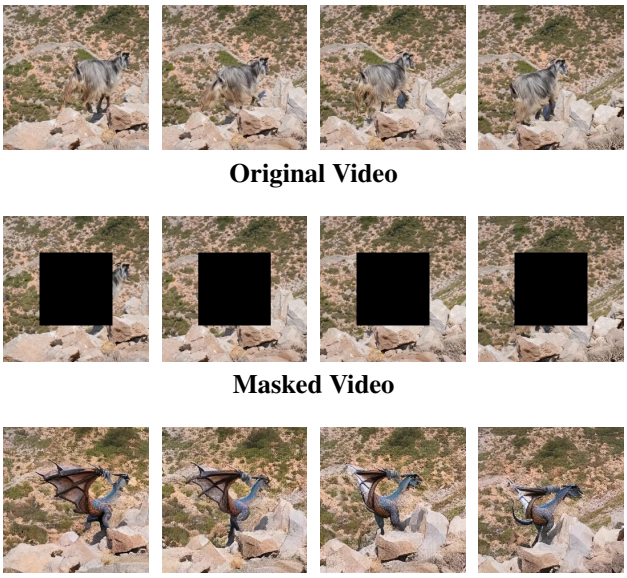
(2) Perceived Gender Expression (3) Perceived Skin Tone. We construct 306 prompts with template — "a {profession or people descriptor} looking {adverb} at the camera" with "profession" being crawled from the US Bureau of Labor and Statistics and "people descriptors" including emotion state, socioeconomic class, *etc*. The "adverb" is used to generate semantically unchanged prompt templates such as "straightly" or "directly". We generate 8 videos for each prompt and for each generated video we infer an approximation of the expressed attribute regarding the 3 protected attributes. Across 10 prompts that have the same semantic meaning but different "adverbs", we observe our outputs generally introduced a stronger distribution shift toward "Young Adults" (age 18-35), "Male" and "Light Skin Tone". However, we observe changing the "adverb" in the prompt template can significantly alter the output distributions. Therefore, our model can be prompted to produce outputs with non-uniform distributions across these groups, but also possess the ability of being prompted to enhance uniformity, though prompts are semantically unchanged. While research has been conducted in the image generation and recognition domain [56, 57, 84], this finding highlights the importance of continued research to develop strategies to mitigate issues and improve fairness for video generation.

7. LLM's Capabilities in Video Generation

In this section we highlight several notable capabilities we discover from the pretrained VideoPoet, shedding light on the Large Language Models (LLMs)'s promising potential in video generation.

7.1. Zero-Shot Video Editing and Task Chaining

A simple example of zero-shot editing is inpainting with text control as in Figure 8, but our model can do even more by chaining multiple capabilties. Because of our multi-task pretraining strategy, our model exhibits task generalization that can be chained together to perform novel tasks. We show an example in Figure 9 that we can apply image-to-video to animate images, and then stylize those images with video-to-video effects. We also show applying video-



Original Video

Masked Video

Inpainted Video from Text Prompt
Prompt: A blue dragon walking along a ridge

Figure 8. Example of zero-shot video editing via task chaining (inpainting and text-to-video) – the original video is first inpainted and then edited via a text prompt.

Animated from Still Image

Stylized Video
Prompt: An oil painting of a snowman with a red hat opening their mouth to yawn

Figure 9. Example of zero-shot video editing via task chaining (text conditioned image-to-video and stylization) – the original painting is first animated via a text prompt and then stylized via another text prompt.

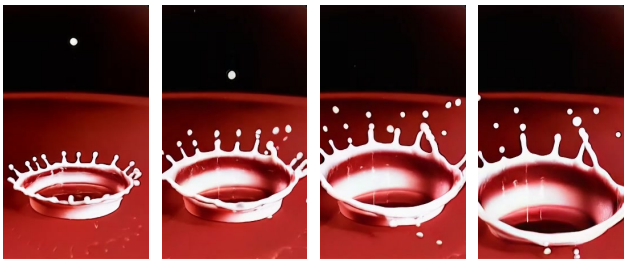
Original Video

Outpainted Video

Stylized Video

Prompt: A gingerbread and candy train on a track

Figure 10. Example of zero-shot video editing via task chaining (outpainting and stylization) – the original video is first outpainted and then stylized via a text prompt.



Animated from Photograph

Animated from Historical Photo

Animated from Painting

Figure 11. Examples of videos animated from a variety of still images plus text prompts tailored to each initial image.

to-video outpainting followed by video-to-video stylization in Figure 10. On our project website[3], we also show text-to-audiovisual-output by generating video from text followed by video-to-audio tasks. At each stage, the quality of the output seems to be sufficient to remain in-distribution (*i.e.* teacher forcing) for the next stage without noticeable artifacts.

We hypothesize that these capabilities are partly attributable to our multimodal task design within a LLM transformer framework that allows for modeling multimodal content using a single transformer architecture over a unified vocabulary. Our approach contrasts with others, such as diffusion models, which typically solve these tasks by adopting multiple individually tuned adapter models to control the diffusion process [24, 29, 40].

7.2. Coherent Long Video Generation and Image-to-Video

A benefit of an decoder-based language model is that it pairs well with autoregressively extending generation in time. We present two different variants of this capability: generating longer videos, and converting images to videos.

Because the MAGVIT-v2 tokenizer that we use encodes the first frame independently of the subsequent frames, we can encode an image without any padding as the first frame of a video. We can then predict the remaining tokens for subsequent frames to produce a video from any image as shown in Figure 11.[4]

We observe temporally conherent generations of objects in a video scene with dynamic, and meaningful motion (see Figure 12). To predict the future frames, despite the model only being able to view up to a short temporal context, such as the first frame or the first second of video, the model is able to keep the motion, style, and identity of objects consistent across more than a total of 8 seconds of video output.

[3] http://sites.research.google/videopoet/

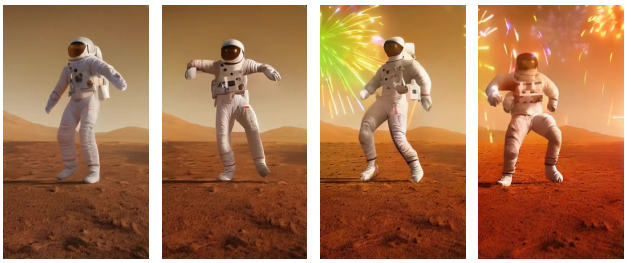


Figure 12. Example of coherent long video generation.

Camera Motion: Arc shot

Camera Motion: FPV drone shot

Figure 13. Examples of directed camera movement from the same initial frame.

7.3. 3D Structure, Camera Motion, Visual Styles

Because our training spans videos, images, and text, we can prompt our model to demonstrate many aspects of understanding about the world including 3D structures, camera motions, and visual styles learned from these different sources. Even though we do not specifically add training data or losses to encourage 3D consistency, our model can rotate around objects and predict reasonable visualizations of the backside of objects. Additionally, with only a small proportion of input videos with text describing camera motion, our model can use short text prompts to apply a range of camera motions to image-to-video and text-to-video generations (see Figure 13), which has been noted to be difficult for many state-of-the-art video generation models [41].

In addition, these controls can be added on top of a wide range of styles, such as watercolor or oil paintings. These stylization training sources are primarily observed in the text-image training data. The ability to generalize across and combine these different types of styles to produce large motions following text prompts underscores the strength of our model's understanding of objects in a temporal context.

[4]For image-to-video examples we source images from Wikimedia Commons: https://commons.wikimedia.org/wiki/Main_Page

8. Conclusion

VideoPoet demonstrates that large language models, trained on discrete tokens, can generate compelling videos and that these models offer a unique flexibility compared to video diffusion models. Our large language model formulation benefits from training on many multimodal tasks with a unified architecture and vocabulary. Consequently, the pre-trained model can serve as a foundation for a diverse variety of video related capabilities, including multiple forms of editing. Moreover, a particular strength of our model lies in its ability to generate high-fidelity, large, and complex motions.

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 preprint arXiv:2301.11325*, 2023. 2

[2] Hassan Akbari, Dan Kondratyuk, Yin Cui, Rachel Hornung, Huisheng Wang, and Hartwig Adam. Alternating gradient descent and mixture-of-experts for integrated multimodal perception. *arXiv preprint arXiv:2305.06324*, 2023. 7

[3] Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023. 2, 3

[4] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 3

[5] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 22563–22575, 2023. 3, 9, 17

[6] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021. 2

[7] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 18392–18402, 2023. 5, 17

[8] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakan-

tan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. 33:1877–1901, 2020. 2, 3

[9] Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics-600. *arXiv preprint arXiv:1808.01340*, 2018. 8

[10] Duygu Ceylan, Chun-Hao P Huang, and Niloy J Mitra. Pix2video: Video editing using image diffusion. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 23206–23217, 2023. 1

[11] Wenhao Chai, Xun Guo, Gaoang Wang, and Yan Lu. Stablevideo: Text-driven consistency-aware diffusion video editing. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 23040–23050, 2023. 1

[12] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 11315–11325, 2022. 3, 5, 9

[13] Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy, William T Freeman, Michael Rubinstein, et al. Muse: Text-to-image generation via masked generative transformers. *arXiv preprint arXiv:2301.00704*, 2023. 3

[14] Haoxin Chen, Menghan Xia, Yingqing He, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Jinbo Xing, Yaofang Liu, Qifeng Chen, Xintao Wang, et al. Videocrafter1: Open diffusion models for high-quality video generation. *arXiv preprint arXiv:2310.19512*, 2023. 9, 10

[15] Weifeng Chen, Jie Wu, Pan Xie, Hefeng Wu, Jiashi Li, Xin Xia, Xuefeng Xiao, and Liang Lin. Control-a-video: Controllable text-to-video generation with diffusion models. *arXiv preprint arXiv:2305.13840*, 2023. 6

[16] Weifeng Chen, Jie Wu, Pan Xie, Hefeng Wu, Jiashi Li, Xin Xia, Xuefeng Xiao, and Liang Lin. Control-a-video: Controllable text-to-video generation with diffusion models. *arXiv preprint arXiv:2305.13840*, 2023. 1, 2, 3, 10, 11, 18

[17] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. PaLM: Scaling language modeling with pathways. *arXiv:2204.02311*, 2022. 2

[18] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240): 1–113, 2023. 2, 3

[19] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. In *Adv. Neural Inform. Process. Syst.*, 2022. 2

[20] 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. *Advances in Neural Information Processing Systems*, 34:19822–19835, 2021. 2

[21] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023. 2

[22] Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. GLaMs: Efficient scaling of language models with mixture-of-experts. 2022. 2

[23] Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7346–7356, 2023. 3, 10

[24] Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7346–7356, 2023. 2, 3, 6, 12, 18

[25] Ruoyu Feng, Wenming Weng, Yanhui Wang, Yuhui Yuan, Jianmin Bao, Chong Luo, Zhibo Chen, and Baining Guo. Ccedit: Creative and controllable video editing via diffusion models. *arXiv preprint arXiv:2309.16496*, 2023. 3

[26] 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. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 22930–22941, 2023. 3, 17

[27] Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Tokenflow: Consistent diffusion features for consistent video editing. *arXiv preprint arXiv:2307.10373*, 2023. 1

[28] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The "something something" video database for learning and evaluating visual common sense. In *Int. Conf. Comput. Vis.*, 2017. 8

[29] Yuwei Guo, Ceyuan Yang, Anyi Rao, Yaohui Wang, Yu Qiao, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725*, 2023. 3, 12

[30] Agrim Gupta, Stephen Tian, Yunzhi Zhang, Jiajun Wu, Roberto Martín-Martín, and Li Fei-Fei. Maskvit: Masked visual pre-training for video prediction. *arXiv preprint arXiv:2206.11894*, 2022. 5

[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*, 2 (3):4, 2023. 3

[32] Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron Weiss, and Kevin Wilson. Cnn architectures for large-scale audio classification. In *ICASSP*, 2017. 8

[33] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 7, 17

[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 preprint arXiv:2210.02303*, 2022. 3, 5

[35] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv:2204.03458*, 2022. 17

[36] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022. 2, 3

[37] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv preprint arXiv:2205.15868*, 2022. 2, 3, 9

[38] Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. Gaia-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023. 3

[39] Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, et al. StarCoder: may the source be with you! *arXiv:2305.06161*, 2023. 2

[40] Jun Hao Liew, Hanshu Yan, Jianfeng Zhang, Zhongcong Xu, and Jiashi Feng. Magicedit: High-fidelity and temporally coherent video editing. *arXiv preprint arXiv:2308.14749*, 2023. 2, 3, 12, 18

[41] Yaofang Liu, Xiaodong Cun, Xuebo Liu, Xintao Wang, Yong Zhang, Haoxin Chen, Yang Liu, Tieyong Zeng, Raymond Chan, and Ying Shan. Evalcrafter: Benchmarking and evaluating large video generation models. *arXiv preprint arXiv:2310.11440*, 2023. 13

[42] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021. 3, 6

[43] Charlie Nash, Joao Carreira, Jacob Walker, Iain Barr, Andrew Jaegle, Mateusz Malinowski, and Peter Battaglia. Transframer: Arbitrary frame prediction with generative models. *arXiv preprint arXiv:2203.09494*, 2022. 2

[44] OpenAI. GPT-4 technical report. *arXiv:2303.08774*, 2023. 2, 3

[45] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016. 8, 10, 17

[46] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 1

[47] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(1):5485–5551, 2020. 2, 3, 5, 7

[48] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *arXiv preprint arXiv:2102.12092*, 2021. 3

[49] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1 (2):3, 2022. 2

[50] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44 (3):1623–1637, 2020. 6

[51] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10684–10695, 2022. 1

[52] Paul K Rubenstein, Chulayuth Asawaroengchai, Duc Dung Nguyen, Ankur Bapna, Zalán Borsos, Félix de Chaumont Quitry, Peter Chen, Dalia El Badawy, Wei Han, Eugene Kharitonov, et al. Audiopalm: A large language model that can speak and listen. *arXiv preprint arXiv:2306.12925*, 2023. 2

[53] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. 35:36479–36494, 2022. 3

[54] Masaki Saito, Shunta Saito, Masanori Koyama, and Sosuke Kobayashi. Train sparsely, generate densely: Memory-efficient unsupervised training of high-resolution temporal gan, 2020. 17

[55] Masaki Saito, Shunta Saito, Masanori Koyama, and Sosuke Kobayashi. Train sparsely, generate densely: Memory-efficient unsupervised training of high-resolution temporal gan. *Int. J. Comput. Vis.*, 128(10):2586–2606, 2020. 8, 9

[56] Candice Schumann, Susanna Ricco, Utsav Prabhu, Vittorio Ferrari, and Caroline Pantofaru. A step toward more inclusive people annotations for fairness. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 916–925, 2021. 11

[57] Candice Schumann, Gbolahan Oluwafemi Olanubi, Auriel Wright, Ellis Monk, Courtney Heldreth, and Susanna Ricco. Consensus and subjectivity of skin tone annotation for ML fairness. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. 11

[58] 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 preprint arXiv:2209.14792*, 2022. 1, 3, 9

[59] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 8, 9, 17

[60] Zineng Tang, Ziyi Yang, Chenguang Zhu, Michael Zeng, and Mohit Bansal. Any-to-any generation via composable diffusion. *arXiv preprint arXiv:2305.11846*, 2023. 3

[61] Yi Tay, Mostafa Dehghani, Vinh Q Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Dara Bahri, Tal

Schuster, Steven Zheng, et al. Ul2: Unifying language learning paradigms. In *Int. Conf. Learn. Represent.*, 2022. 2, 5

[62] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Eur. Conf. Comput. Vis.*, pages 402–419, 2020. 6

[63] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 3

[64] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. In *Eur. Conf. Comput. Vis.*, pages 459–479, 2022. 5

[65] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. Fvd: A new metric for video generation. 2019. 8, 9

[66] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. 30, 2017. 3

[67] 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 preprint arXiv:2210.02399*, 2022. 1, 2, 3, 9, 10, 17

[68] Vikram Voleti, Alexia Jolicoeur-Martineau, and Chris Pal. Mcvd-masked conditional video diffusion for prediction, generation, and interpolation. 35:23371–23385, 2022. 1

[69] Jiuniu Wang, Hangjie Yuan, Dayou Chen, Yingya Zhang, Xiang Wang, and Shiwei Zhang. Modelscope text-to-video technical report. *arXiv preprint arXiv:2308.06571*, 2023. 3, 9

[70] Wen Wang, Kangyang Xie, Zide Liu, Hao Chen, Yue Cao, Xinlong Wang, and Chunhua Shen. Zero-shot video editing using off-the-shelf image diffusion models. *arXiv preprint arXiv:2303.17599*, 2023. 1

[71] Wenjing Wang, Huan Yang, Zixi Tuo, Huiguo He, Junchen Zhu, Jianlong Fu, and Jiaying Liu. Videofactory: Swap attention in spatiotemporal diffusions for text-to-video generation. *arXiv preprint arXiv:2305.10874*, 2023. 3, 9

[72] Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinyuan Chen, Yaohui Wang, Ping Luo, Ziwei Liu, et al. Internvid: A large-scale video-text dataset for multimodal understanding and generation. *arXiv preprint arXiv:2307.06942*, 2023. 3, 9

[73] 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. 8, 9

[74] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296, 2016. 8, 9, 17

[75] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157*, 2021. 3

[76] 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 preprint arXiv:2206.10789*, 2022. 2, 3, 17

[77] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10459–10469, 2023. 2, 3, 4, 5, 7, 8, 9, 17

[78] 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, 17

[79] Sihyun Yu, Kihyuk Sohn, Subin Kim, and Jinwoo Shin. Video probabilistic diffusion models in projected latent space. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 18456–18466, 2023. 1

[80] Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:495–507, 2021. 4, 5

[81] Yan Zeng, Guoqiang Wei, Jiani Zheng, Jiaxin Zou, Yang Wei, Yuchen Zhang, and Hang Li. Make pixels dance: High-dynamic video generation. *arXiv preprint arXiv:2311.10982*, 2023. 3

[82] David Junhao Zhang, Jay Zhangjie Wu, Jia-Wei Liu, Rui Zhao, Lingmin Ran, Yuchao Gu, Difei Gao, and Mike Zheng Shou. Show-1: Marrying pixel and latent diffusion models for text-to-video generation. *arXiv preprint arXiv:2309.15818*, 2023. 1, 3, 9, 10

[83] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3836–3847, 2023. 2, 3, 6

[84] Yanzhe Zhang, Lu Jiang, Greg Turk, and Diyi Yang. Auditing gender presentation differences in text-to-image models. *arXiv preprint arXiv:2302.03675*, 2023. 11

[85] Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. Lima: Less is more for alignment. *arXiv preprint arXiv:2305.11206*, 2023. 7

[86] 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. 3, 9

[87] 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. 2023. 2

A. Appendix

A.1. Zero-shot text-to-video evaluation settings

We report the full details of our zero-shot text-to-video settings here. We note that some details are missing in previous papers and different papers use different settings. Hence, we provide all the details and hope this evaluation setting can serve as a standard text-to-video generation benchmark. Our results are reported on the 8B model and we adopt classifier-free guidance [33].

Resolution All metrics are evaluated on generated videos containing 16 frames with a resolution of 256 x 256. We first generate videos of 128 x 128 resolution and then resize to 256 x 256 via bicubic upsampling.

Zero-shot MSR-VTT For CLIP score, we used all 59,794 captions from the MSR-VTT test set. We use CLIP ViT-B/16 model following Phenaki [67]. We note that some papers use other CLIP models, *e.g.*, VideoLDM [5] uses ViT-B/32. Our CLIP score evaluated on the ViT-B/32 backbone for MSR-VTT is 30.01. For the FVD metric, to evaluate on a wide range of captions as well as to be comparable with previous papers that evaluate on 2,048 videos, we evaluate on the first 40,960 captions in the MSR-VTT test set. More specifically, we report the FVD metrics on 2048 videos with 20 repeats. The FVD real features are extracted from 2,048 videos sampled from the MSR-VTT test set. We sample the central 16 frames of each real video, without any temporal downsampling, *i.e.*, we use the original fps in the MSR-VTT dataset (30 fps as reported in [74]). The FVD is evaluated with an I3D model trained on Kinetics-400.

Zero-shot UCF-101 Following VDM [35], we sample 10,000 videos from the UCF-101 test set and use their categories as the text prompts to generate 10,000 videos. We use the class text prompts provided in PYoCo [26] to represent the 101 categories. To compute the FVD real features, we sample 10K videos from the training set, following TGAN2 [54]. We sample the central 16 frames for each real video , without any temporal downsampling, *i.e.*, we use the original fps in the UCF-101 dataset (25 fps as reported in [59]). The FVD metric is evaluated with an I3D model trained on Kinetics-400 and the IS metric is evaluated with a C3D model trained on UCF-101.

A.2. Super-resolution implementation details

We use a 1B model for the first $2\times$ spatial super-resolution stage and a 500M model for the second $2\times$ stage. The first super-resolution stage models videos of $17 \times 448 \times 256$ pixels with a token sequence of shape $(5, 56, 32)$. The second stage models videos of $17 \times 896 \times 512$ pixels with a token sequence of shape $(5, 112, 64)$. The token sequences are obtained with the same MAGVIT-v2 [78] tokenizer used for the base language model. The custom super-resolution transformer has local self-attention windows for *vertical*, *horizontal* and *temporal* layers of shape $(1, 56, 4), (1, 8, 32), (5, 8, 8)$ in the first stage and $(1, 112, 2), (1, 4, 64), (5, 8, 8)$ in the second stage, respectively (Figure 4). The cross-attention layers attend to local windows in the low-resolution sequence isomorphic to self-attention windows but with half the spatial size.

We train the super-resolution stages on a dataset of 64M high-quality text-video pairs using the masked modeling objective of MAGVIT [77], with token factorization into $k = 2$ groups [78]. During inference, we use the sampling algorithm of MAGVIT-v2 [78] with 24 sampling steps for each stage and classifier-free guidance scale [7, 33] of $4.0/8.0$ for the text condition and $1.0/2.0$ for the low-resolution condition, in the first/second stage.

A.3. Comparison of 1B and 8B models

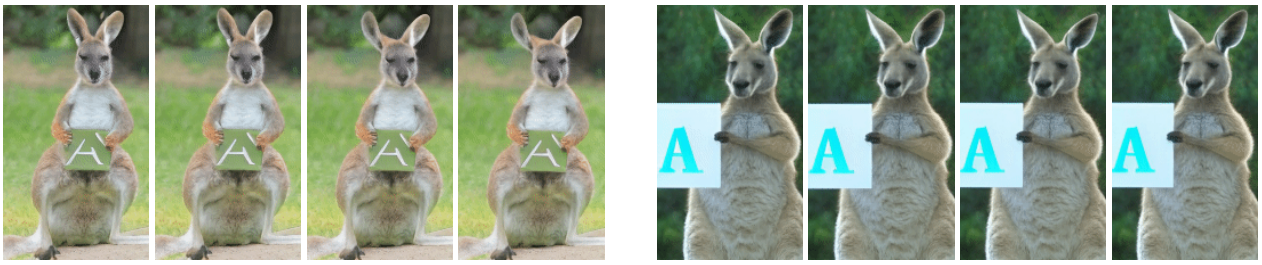
In Figure 14, we show outputs of 1B and 8B parameter models on the same prompts. Four frames from the best video output of each model in a batch of four text-to-video samples were selected to represent the model. In the first row, the 1B model is unstable with large changes to the subject over time and misses elements from the complex prompt. This prompt was originally used for scaling comparisons in [76], and compared to a dedicated image-only model, our model does not preserve text as well given the training data used. In the second row, we use a simpler text task and show that the 8B model can represent a single letter clearly, but the 1B model still produces artifacts. In the third row, we show that the 8B model learns spatial positioning such that the river is in front of the astronaut and horse. In the fourth row, we show that the 8B parameter model learned a stop motion style to have items disappear "one by one" and can follow a complicated layout from a long prompt. In contrast, the 1B model includes all of the nouns, but is unstable over time and does not follow the layout indicated in the prompt. In the bottom row, we show that the 8B model understands counts of objects in that it displays a full bouquet (though 12 roses are not explicitly in frame) and smooth consistent motion as opposed to the 1B model 5 roses and distorting objects produced by the 1B model. Overall, scaling the model improved temporal consistency, prompt fidelity, and motion dynamics while adding capabilities for limited text rendering, spatial understanding, and counting.

A.4. Stylization Evaluation on DAVIS

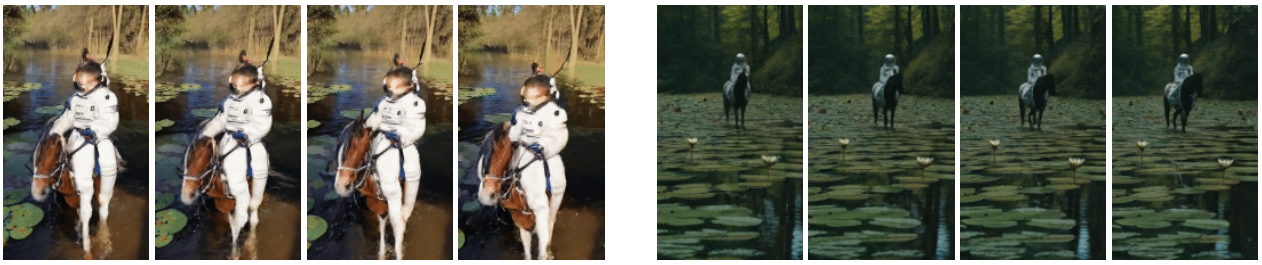
To evaluate the CLIP similarity score and human preference on video stylization, we use the following set of videos and prompts. We select 20 videos from DAVIS 2016 [45], and for each video we take 16 frames starting from the initial

frame specified below and evaluate stylization on the two text prompts specified below. To be easily reproducible, we use a central square crop at the height of the video and evaluate the output videos at 256x256 resolution. We use CLIP-B/16 for the similarity score. Several prompts below are inspired by previous work [16, 24, 40].

prompt: A portrait photo of a kangaroo wearing an orange hoodie and blue sunglasses standing on the grass in front of the Sydney Opera House holding a sign on the chest that says Welcome Friends!

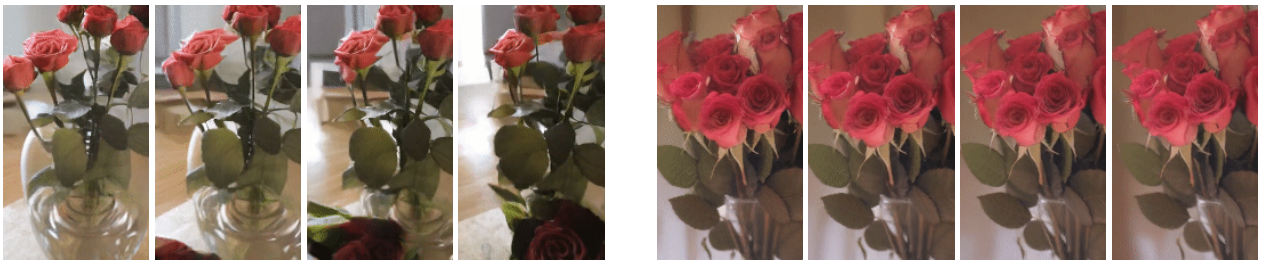


prompt: A kangaroo holding a sign with the letter A on it



prompt: A photo of an astronaut riding a horse in the forest. There is a river in front of them with water lilies

prompt: A zoomed out map of the United States made out of sushi. It is on a table next to a glass of red wine. Pieces of sushi disappear one by one



prompt: Rotating around a vase holding a dozen roses

Figure 14. **A comparison of a 1B (left) and 8B (right) parameter models** on the same prompt and settings.

video name	starting frame	first text prompt
elephant	10	oil painting of an elephant walking away
elephant	10	cartoon animation of an elephant walking through dirt surrounded by boulders
car-turn	40	car on a snowcovered road in the countryside
car-turn	40	8-bit pixelated car driving down the road
dog-agility	0	a dog in the style of a comic book
dog-agility	0	a dog running through a field of poles in the style of cyberpunk
bmx-bumps	10	riding a bicycle on a rainbow track in space with stars and planets in the background
bmx-bumps	10	riding a bicycle on a dirt track in the style of a graphic novel
train	0	a gingerbread steam train made of candy
train	0	a train in lava
bus	0	a black and white drawing of a bus
bus	0	a bus in cyberpunk style
lucia	0	an astronaut walking on mars
lucia	0	a claymation animation of a woman walking
tennis	15	a robot throwing a laser ball
tennis	15	astronaut playing tennis on the surface of the moon
bear	60	a polar bear exploring on an iceberg
bear	60	a space bear walking beneath the stars
flamingo	0	2D vector animation of a group of flamingos standing near some rocks and water
flamingo	0	oil painting of pink flamingos wading
hike	0	a green alien explorer hiking in the mountains
hike	0	paper cut-out mountains with a paper cut-out hiker
goat	59	a tiger prowling along the ridge above a jungle
goat	59	a dragon prowling over a crater on the moon
parkour	60	a man jumping over rocks in a red sandstone canyon
parkour	60	a robot dodging through an obstacle course
cows	10	a pig standing in the mud
cows	10	a robotic cow walking along a muddy road
camel	10	a camel robot on a snowy day
camel	10	toy camel standing on dirt near a fence
blackswan	0	a watercolor painting of a white swan
blackswan	0	a crochet black swan swims in a pond with rocks and vegetation
dog	20	a cat walking
dog	20	a dalmatian dog walking
kite-surf	10	a sand surfer kicking up sand in the desert
kite-surf	10	kite surfer in the ocean at sunset
libby	0	chinese ink painting of a dog running
libby	0	3D animation of a small dog running through grass
horsejump-high	0	a cartoon of a magical flying horse jumping over an obstacle
horsejump-high	0	person rides on a horse while jumping over an obstacle with an aurora borealis in the background

Table 5. DAVIS stylization evaluation settings.