

# Allegro: Open the Black Box of Commercial-Level Video Generation Model

Yuan Zhou, Qiuyue Wang, Yuxuan Cai, Huan Yang\*

Rhymes AI

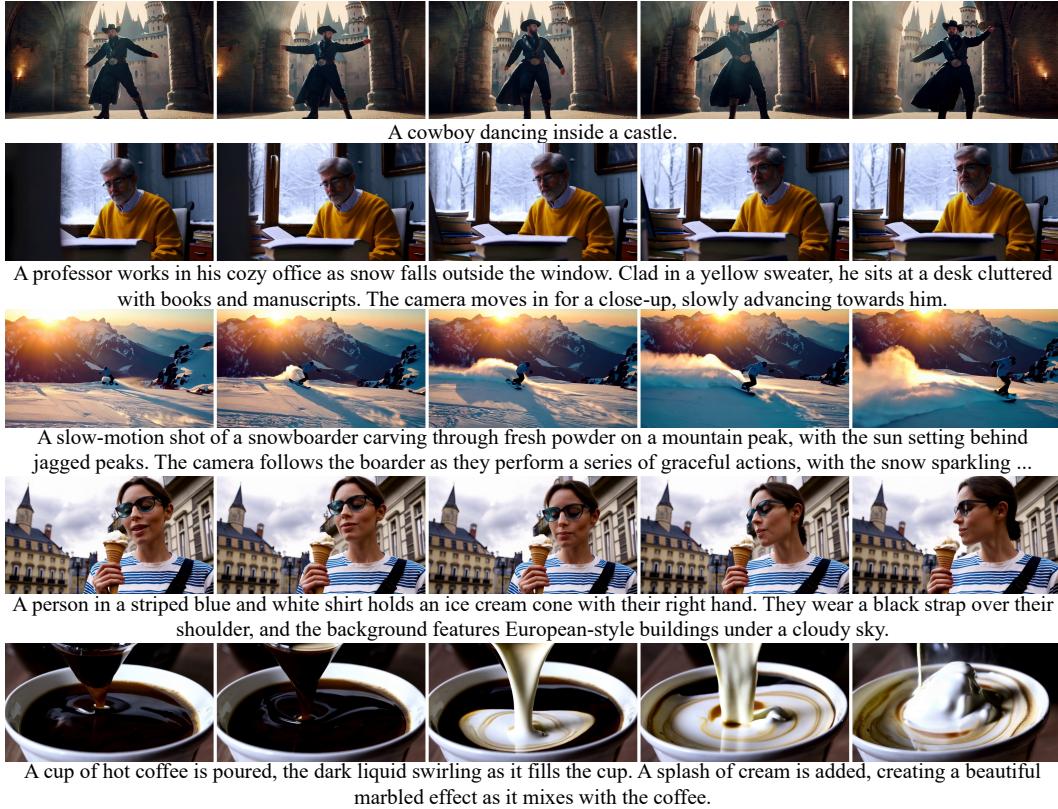


Figure 1: Allegro is capable of generating high-quality, dynamic videos from various descriptive text inputs.

## Abstract

Significant advancements have been made in the field of video generation, with the open-source community contributing a wealth of research papers and tools for training high-quality models. However, despite these efforts, the available information and resources remain insufficient for achieving commercial-level performance. In this report, we open the black box and introduce **Allegro**, an advanced video generation model that excels in both quality and temporal consistency. We also highlight the current limitations in the field and present a comprehensive methodology for training high-performance, commercial-level video generation models, addressing key aspects such as data, model architecture, training pipeline, and evaluation. Our user study shows that Allegro surpasses existing open-source models and most commercial models, ranking just behind Hailuo and Kling.

Code: <https://github.com/rhymes-ai/Allegro>

Model: <https://huggingface.co/rhymes-ai/Allegro>

Gallery: [https://rhymes.ai/allegro\\_gallery](https://rhymes.ai/allegro_gallery)

\* corresponding author: huanyang@rhymes.ai

## 1 Introduction

The demand for video content in digital media has surged in recent years, driving innovation in automated video generation technologies. As these technologies evolve, the process of creating videos has become more accessible, efficient, and streamlined. Among these advancements, text-to-video generation models have emerged as a breakthrough. These models allow users to generate dynamic visual content from descriptive text, offering a highly flexible and controllable method for video creation. By transforming textual input into visual narratives, they are democratizing video production and opening new possibilities for content creators across industries.

With the advent of diffusion models, the field of visual content generation has experienced significant growth [Ho et al., 2020, Rombach et al., 2022, Saharia et al., 2022, Blattmann et al., 2023, Ho et al., 2022, Singer et al., 2022, Voleti et al., 2022, Ruan et al., 2023, Wang et al., 2024, He et al., 2024]. In the realm of image generation, open-source models have achieved performance levels comparable to many commercial products. Meanwhile, the field of text-to-video generation has seen the emergence of numerous commercial systems, such as Runway Gen-3 [RunwayML, 2024], Luma Dream Machine [Luma Lab, 2024], OpenAI Sora [OpenAI, 2024], Kling [Kuaishou, 2024], Hailuo [MiniMax, 2024], and Pika [Pika Labs, 2024]. However, unlike text-to-image generation, many key aspects of developing commercial-level text-to-video generation models are still unclear. These include efficiently modeling video data, accurately aligning videos with textual semantics, incorporating more precise controls, and developing high-quality text-video datasets for training.

In this report, we open the black box of commercial-level video generation models and introduce **Allegro**, an elite video generation model with exceptional quality and temporal consistency. We examine the critical elements that contribute to enhancing the model’s output, offering detailed insights into the techniques and strategies employed for training a commercial-level video generation model. The report serves as an extensive guide for developing high-quality video generation models leveraging diffusion methods, with a focus on redesigning key components including:

- **Data curation.** We propose a systematic data curation pipeline for training a commercial-level video generation model using large-scale image and video datasets, providing detailed insights into constructing optimized training data. With this pipeline, we build our dataset containing 106M images and 48M videos, with highly associated text captions.
- **Model architecture.** Building on the Diffusion framework, we introduce modifications to the Variational Autoencoder (VAE) and Diffusion Transformer (DiT) architectures to better accommodate the specific requirements of video generation. We outline strategies to optimize the computational infrastructure, ensuring efficient training and inference.
- **Evaluation.** We build a diverse benchmark for text-to-video generation and conduct user studies and subjective evaluations in six dimensions to ensure the generated videos meet aesthetic standards and align with human preferences.

Through the implementation of the aforementioned components, Allegro achieved results in subjective evaluations comparable to those of other commercial systems. As shown in Figure 1, Allegro is capable of generating high-quality, dynamic videos from various descriptive text inputs. Our user study demonstrates that Allegro outperforms current open-source models across all six dimensions and shows significant advantages over commercial models in most areas. Notably, it surpasses all commercial models in video-text relevance and ranks just behind Hailuo and Kling in overall quality.

In addition, we offer further insights and constructive guidance on how to enhance the base capabilities of the model, including model scaling-up, prompt refiner adaptation, and video tokenizer designing. Meanwhile, we are working on extending Allegro’s foundational text-to-video model in two key directions: Image-to-Video with text conditions and more flexible motion control. These advancements aim not only to add more features to Allegro, but also to provide deeper insights on better aligning the model’s capabilities with user needs and boosting productivity. We release Allegro under the Apache 2.0 license, free for both academic and commercial use.

## 2 Data Curation

Data curation is the primary task in building video generation models, permeating the entire training process. Existing publicly available datasets, such as WebVid [Bain et al., 2021], Panda-70M [Chen et al., 2024b], HD-VILA [Xue et al., 2022], HD-VG [Wang et al., 2023] and OpenVid-1M [Nan et al., 2024], have provided solid foundation for data sourcing and acquisition, offering diverse and extensive video data. However, with the sheer volume of data now available, significant challenges arise in terms of processing efficiency, data redundancy, and ensuring high-quality inputs for model training.

One of the key issues with current data curation practices is the lack of efficient pipelines capable of handling and transforming these massive datasets into well-prepared, structured inputs. The time and computational resources required to process raw data are often prohibitive, leading to bottlenecks in the development cycle. Moreover, without carefully curated datasets, there is an increased risk of including noisy, redundant, or irrelevant data, which can degrade the performance of the models.

In addition to improving processing efficiency, another critical goal is to ensure that the curated datasets are not only large in volume but also diverse and of high quality. The success of video generation models hinges on the quantity of data and its richness in terms of variety, coherence, and prompt alignment. By addressing these issues, the objective is to build a robust data pipeline that optimizes the balance between data volume and quality, ultimately enhancing model performance in generating realistic and coherent videos.

In this section, we introduce a systematic and scalable data curation pipeline designed to meet the demands of large-scale video datasets. This pipeline aims to streamline the processing of raw video data and ensure that high-quality, diverse samples are used for training, providing insights for improving the overall dataset quality and efficiency. We also provide a detailed analysis of the data distribution on different dimensions and hopefully give some insights on building a better training dataset.

### 2.1 Data Processing

The data processing pipeline plays a pivotal role in ensuring the quality and diversity of the data used for different stages of the model training process. This pipeline consists of three critical parts: data filtering, annotation, and stratification.

#### 2.1.1 Data Filtering

The data filtering pipeline is illustrated in Figure 2. Before the fine-grained video captioning, which will be introduced in the next section, we apply seven filtering steps in sequence: 1) Duration and resolution filtering; 2) Scene segmentation; 3) Low-level metrics filtering; 4) Aesthetics filtering; 5) Content-irrelevant artifacts filtering; 6) Coarse-grained captioning; and 7) CLIP similarity filtering. This pipeline is adaptable for both image and video data, with only minor adjustments for each data format. We introduce the details of each filtering step below. Due to the varying data quality requirements at different training stages, we focus on describing the specific methods of each data filtering step. The specific threshold values for certain dimensions will be detailed in Section 2.1.3.

**Duration, FPS and Resolution Filtering.** We remove the raw images and videos with a resolution lower than 360p. We also remove raw videos that are shorter than 2 seconds or frame rates lower than 23 FPS. This step ensures that our dataset consists of high-quality image/video clips with sufficient length and visual fidelity for effective model training.

**Scene Segmentation.** We use PySceneDetect [PySceneDetect Contributors, 2024] to detect the scenes and divide the raw videos into single-scene video clips. We removed each video clip's first and last 10 frames to avoid false positives in scene detection. Only the clips with a duration of 2 to 16 seconds are kept to the next step.

**Low-level Metrics Filtering.** In this step, we focus on the visual quality of the data. We evaluate each video clip from four dimensions: brightness, clarity, semantic consistency, and motion amplitude. We calculate the average grayscale of the images and the intermediate

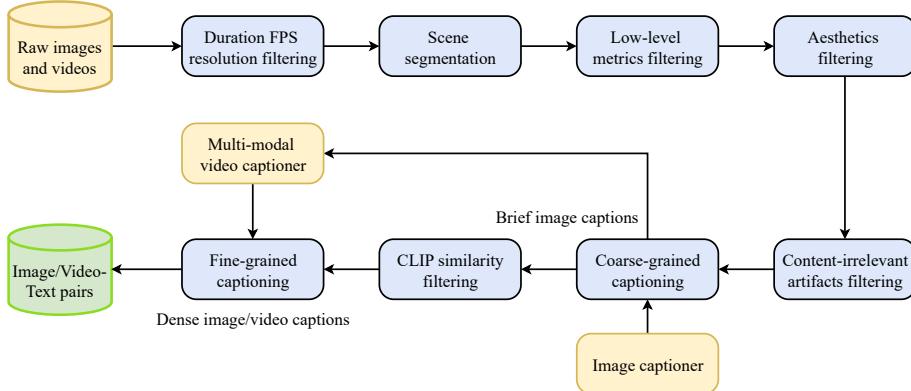


Figure 2: Allegro data filtering pipeline. A large pool of raw images and videos is fed into this pipeline to get a training set of high visual quality with associated text captions.

frame in video clips to filter out images and videos that are too dark or too bright. The DOVER score [Wu et al., 2023] is used to assess images and video clips’ clarity. We employ LPIPS [Zhang et al., 2018] and UniMatch [Xu et al., 2023] to evaluate video clips’ semantic consistency and motion amplitude, respectively. These two metrics are jointly considered to retain video clips with reasonable semantic consistency and motion amplitude.

**Aesthetics Filtering.** We use the LAION Aesthetics Predictor [Schuhmann et al., 2022] to evaluate the aesthetic score of images and the intermediate frames of each video clip, selecting the data with better visual appeal.

**Content-irrelevant Artifacts Filtering.** The raw videos may contain artifacts unrelated to the video content, such as black borders, text, watermarks, etc., which can negatively affect video generation models. We use detection tools (e.g., the text detection tool CRAFT [Baek et al., 2019] and the watermark detection tool [Watermark-Detection Contributors, 2022]) to detect these elements and calculate their area ratio in the image or video frame. When the ratio is below the set threshold, we crop the original image or video; otherwise, the data will be discarded.

**Coarse-grained Captioning.** We perform preliminary annotation on the remaining data to obtain coarse-grained captions, which provide global semantic information for images and videos. These coarse-grained captions also serve as input in the following **CLIP similarity filtering** and fine-grained captioning. In this step, we use Tag2Text [Huang et al., 2023] to caption the middle frames of the images and videos, which significantly improves the efficiency of obtaining coarse-grained descriptions during data processing.

**CLIP Similarity Filtering.** The correlation of visual data and its caption is assessed by calculating the cosine similarity between the CLIP [Radford et al., 2021] embeddings of the caption and those of the image or the video middle frame. We filter out samples with low CLIP cosine similarity to ensure more accurate inputs for subsequent fine-grained captioning and provide high-quality annotations for the final training data.

To facilitate the usability of the data, all video clips are uniformly re-encoded using H.264, and the frame rate is fixed at 30 FPS. After the data filtering pipeline above, the final video clips are 2-16 seconds long, captured with a single-shot camera, and exhibit high visual quality, reasonable motion, and highly correlated coarse-grained captions.

### 2.1.2 Data Annotation

**Fine-grained Captioning.** We generate the fine-grained captions for both images and videos after data filtering by using Aria [Li et al., 2024b], which has been finetuned on the video captioning task. The coarse-grained captions are included in the video caption model’s instructions to enhance annotation accuracy. The fine-grained captions cover both the spatial and temporal information, including attributes of the subjects, interactions among

Filtering Step	Text-to-Image Pre-training	Text-to-Video Pre-training-360p	Text-to-Video Pre-training-720p	Text-to-Video Fine-tuning
Duration	—	[2s, 16s]	[2s, 16s], [6s, 16s]	[6s, 16s]
FPS	—	(23, 61)	(23, 61)	(23, 61)
Resolution (W=Width, H=Height)	W ≥ 640, H ≥ 368	W ≥ 640, H ≥ 368	W ≥ 1280, H ≥ 720	W ≥ 1280, H ≥ 720
Brightness	[20, 180]	[20, 180]	[20, 180]	[20, 180]
Clarity (DOVER Score)	—	—	—	≥ 0.07
Semantic Consistency (LPIPS score)	—	—	—	≥ 0.05
Motion (UniMatch Score)	—	—	—	[1.0, 100]
Aesthetics (Aesthetic Score)	≥ 4.8	≥ 4.8	≥ 5.0	≥ 5.3
Text Area Ratio in %	≤ 0.05	≤ 0.05	≤ 0.05	≤ 0.05
CLIP Similarity	≥ 0.17	≥ 0.17	≥ 0.20	≥ 0.20
Remains/Total Data Counts	107M/412M	48M/500M	18M/500M	2M/500M

Table 1: Data filtering threshold across different training stages.

the subjects, description of the backgrounds, environment, style, and atmosphere, camera angle and motion, and the process of the above content changing over time. To enhance the model’s capability in camera control, some data includes video captions with sentences that begin with "Camera [MOTION\_PATTERN]." to explicitly highlight the camera motion.

Based on data annotations in both coarse- and fine-grained captioning stages, we obtained the final image/video-text data pairs. For image data, there are coarse- and fine-grained captions. For video data, there are coarse- and fine-grained captions for both the entire video and middle frames. More examples of the data captions are listed in Appendix A.

### 2.1.3 Data Stratification

The annotated data is organized into different datasets tailored to the needs of various training stages. Specifically, the data is stratified to support three main stages of progressive training: text-to-image pre-training, text-to-video pre-training, and text-to-video fine-tuning. Table 1 illustrates the specific threshold for different dimensions at different training stages.

**Text-to-Image Pre-training Data.** This dataset only contains the image-text pairs and is designed for training models to only generate images from textual descriptions. In this stage, the model learns how to map textual input to visual features (e.g., object shapes, textures, colors). It serves as a foundational step before moving on to video generation tasks. We finally get 107M images filtered from 412M raw images.

**Text-to-Video Pre-training Data.** This dataset contains video-text pairs that allow the model to learn how to generate coherent sequences of images (i.e., video) from text prompts. The model builds upon the knowledge gained during text-to-image pre-training but is enforced to handle temporal consistency. This stage comprises two sub-stages based on different video resolutions (360p and 720p). We keep 48M and 18M (out of 500M) video clips for the two sub-stages, respectively.

**Text-to-Video Fine-tuning Data.** This data is used in the final fine-tuning stage, where the model is optimized for video generation tasks. It ensures that the model produces high-quality, temporally consistent videos that align with the input text. The model can better handle complex scenes, intricate motions, and detailed textures in the generated video. Only around 2M video clips are selected for this final stage.

## 2.2 Data Analysis

More details of data statistics, representative cases and data insights are in Appendix A.

## 3 Video Variational Auto-Encoder

Previous studies [Rombach et al., 2022, Podell et al., 2024] have demonstrated that the efficiency and performance of diffusion models can be significantly enhanced by adopting latent space modeling using a Variational Auto-Encoder (VAE) [Kingma, 2013]. The Sora [OpenAI, 2024] team has also employed a video spatio-temporal compression network and trained Sora to generate videos within this compressed latent space. Consequently, a

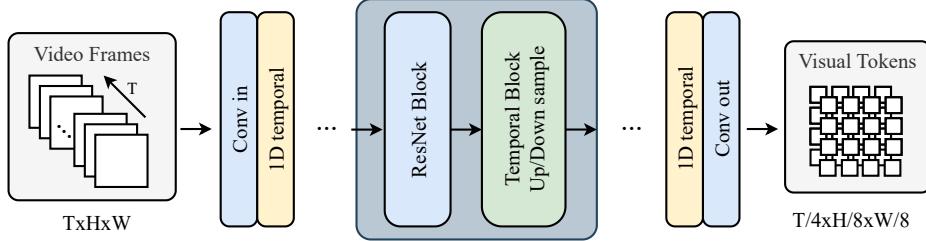


Figure 3: The architecture of our VideoVAE. The blue part represents the spatial layers of the original image VAE, while the other colored modules are the added temporal layers. With video frames of shape  $T \times 3 \times H \times W$  as input to the VideoVAE encoder, we can obtain latent video representation with shape  $T/S_T \times C_l \times H/S_H \times W/S_W$ . The figure shows the model structure and process of the encoder part. The decoder part is symmetric and it takes the latent as input and reconstructs RGB videos.

practical approach to generating long, high-resolution videos is to compress RGB videos in both spatial and temporal dimensions before modeling them in the latent space. The latent predicted by the Diffusion Transformer (DiT) is then decoded using a VAE decoder to produce the RGB videos. To facilitate this process, we constructed a Video Variational Auto-Encoder (VideoVAE), which encodes RGB videos into a compressed latent spatio-temporal space to improve the efficiency and performance of the diffusion model.

### 3.1 Model Architecture

The effectiveness of a VAE primarily depends on its ability to perform robust spatio-temporal modeling. To fully leverage the spatial compression capability of the existing image VAE, our VideoVAE design incorporates extended spatio-temporal modeling layers based on the image VAE. Specifically, We first adopt the architecture and weights of the image VAE from Playground v2.5 [Li et al., 2024a] to model spatial information. Then, we incorporate temporal modeling layers to capture temporal information and apply temporal compression to reduce its dimensionality.

Our VideoVAE architecture is depicted in the Figure 3. We add a 1D temporal CNN layer at the beginning and the end of both the encoder and the decoder in the image VAE. Additionally, we design a temporal block consisting of four 3D CNN temporal layers, which are inserted after every ResNet block in VAE. In temporal blocks, temporal downsampling is achieved through stridden convolution with a stride of 2, while upsampling is accomplished by frame repetition followed by a deconvolution operation. After incorporating these temporal layers, the parameter count of our VideoVAE has reached 174.96M.

In the preliminary experiments, the temporal CNN layer within the temporal block had a kernel size of 1 in the spatial dimension (i.e., 1D temporal CNN layer). However, we discover that a spatial kernel size of 3 could accelerate convergence and achieve better reconstruction performance. Therefore, we employ 3D CNN layers in all upsampling and downsampling temporal blocks.

The reconstruction upper bound of VideoVAE is highly correlated with the spatial compression stride, temporal compression stride, and the number of dimensions in the latent space. We continue to use the latent channel  $C_l = 4$  from image VAE without any expansion and have compression stride  $S_T \times S_H \times S_W = 4 \times 8 \times 8$  in temporal, height and width dimension respectively. With video frames  $T \times 3 \times H \times W$  as input to the VideoVAE encoder, we can obtain latent video representation with shape  $T' \times C_l \times H' \times W' = T/S_T \times C_l \times H/S_H \times W/S_W$ .

### 3.2 Training Strategy

The quality of reconstructed videos after compression and decompression by VideoVAE is closely tied to the quality of the training data. As a result, we prioritize videos and images with higher resolutions for training. Specifically, we only use images and videos where the

shortest side exceeds 720 pixels. We ultimately select 54.7K videos and 3.73M images from our filtered dataset to train VideoVAE.

The effectiveness of VideoVAE in video reconstruction is strongly linked to the level of image detail. Since resizing images can reduce detail, our spatial data augmentation only includes random cropping. For temporal data augmentation in videos, we randomly choose a sampling interval from the set [1, 3, 5, 10] to sample frames evenly across video clips. Experiments demonstrate that using larger sampling intervals significantly accelerates the convergence of the temporal layers.

During model training, we exclusively utilize L1 loss and LPIPS loss [Zhang et al., 2018], with respective loss weights of 1.0 and 0.1. L1 loss, or mean absolute error (MAE), measures the average absolute differences between predicted and actual values. LPIPS loss [Zhang et al., 2018] is a perceptual loss function that evaluates image similarity based on learned features from a deep neural network, aiming to capture human visual perception for tasks like image reconstruction and style transfer.

We conduct a two-stage training on VideoVAE. In the first stage, we perform joint training of the VideoVAE by incorporating both images and videos, utilizing a batch composition of one 16-frame video and four images, all with a spatial resolution of  $256 \times 256$ . This stage is conducted for 65K iterations on 32 Nvidia A100 GPUs, with batch size 1 per GPU.

Image data will exclusively pass through the original image VAE’s spatial layers, while video data will traverse both spatial and temporal layers. We anticipate that this mixed training strategy will not only effectively enhance the model’s spatiotemporal modeling capabilities but also maintain its excellent spatial modeling capabilities through the utilization of images.

In the second stage, we freeze the spatial layers from the image VAE and fine-tune the temporal layers using 24-frame  $256 \times 256$  resolution videos to enhance temporal modeling capabilities. In this stage, we fine-tune the temporal layers for 25K iterations on 32 Nvidia A100 GPUs, with batch size 1 per GPU.

### 3.3 Inference Strategy

Encoding and decoding high-resolution, long-duration videos directly is impractical due to memory limitations. To handle such videos during inference, we divide both the input video and the latent tensor into spatio-temporal tiles. Each tile is then encoded and decoded separately, and the results are reassembled to form the final output. In practice, we employ a tile size of  $24 \times p \times p$  (time, height, width) in Encoder, and a tile size of  $6 \times p/8 \times p/8$  in Decoder. While blending and reassembling the output, we choose an overlapping stride  $v_T \times v_H \times v_W$  among pixel tiles and  $v_T/4 \times v_H/8 \times v_W/8$  among latent tiles. For 320p video generation, we take  $p = 256$ ,  $v_T \times v_H \times v_W = 8 \times 64 \times 144$ . And we have  $p = 320$ ,  $v_T \times v_H \times v_W = 8 \times 80 \times 120$  for 720p video generation. Following the approach of the image VAE, we utilize a linear combination in both spatial and temporal dimensions for blending. For image encoding and decoding, we only use the spatial layers in VideoVAE, which is consistent with those in the training process.

## 4 Video Diffusion Transformer

The Video Diffusion Transformer (VideoDiT) represents a breakthrough in video generation by combining the strengths of diffusion models with the Transformer architecture [Ho et al., 2022, Ma et al., 2024, Dinesh Reddy et al., 2022, Yang et al., 2024, Lu et al., 2023, Chen et al., 2024a]. VideoDiT excels at modeling both spatial and temporal dependencies, ensuring improved coherence across video frames. This approach generates high-quality, realistic videos while scaling efficiently on larger datasets, outperforming traditional methods. Several benchmarks consistently demonstrate DiT’s superior performance in video generation tasks [Huang et al., 2024, Liu et al., 2024a, Sun et al., 2024, Liu et al., 2024b].

Recently, open-source projects such as Open-Sora [Zheng et al., 2024] and Open-Sora-Plan [PKU-Yuan Lab and Tuzhan AI etc., 2024] have significantly advanced the field of video generation. Notably, the Open-Sora-Plan has been widely adopted for its robust

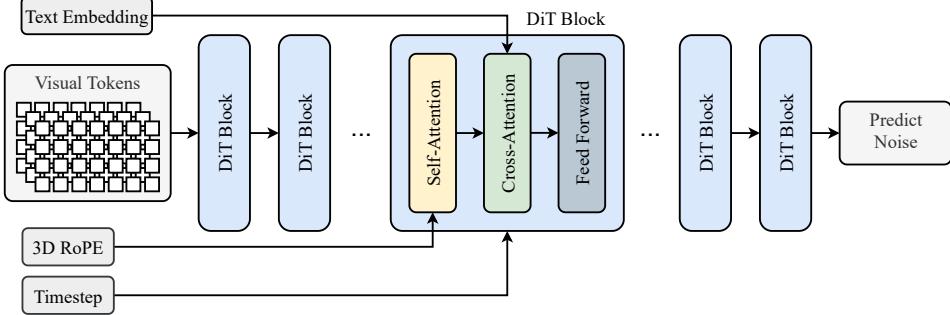


Figure 4: The architecture of our VideoDiT. The T5 text encoder converts natural language into text embeddings. The visual token is encoded by our proposed VideoVAE encoder. The video Transformer network, composed of multiple DiT blocks, predicts the noise based on visual tokens, text embedding, diffusion timestep, and 3D RoPE.

implementations. Our work builds upon the VideoDiT architecture introduced in Open-Sora-Plan v1.2.0 and aims at opening the black box of training a commercial-level video generation model.

We introduce these key modifications to improve performance. First, we replace the original mT5 text encoder [Xue, 2020] with T5 [Raffel et al., 2020], enhancing text-to-video alignment and improving both quality and coherence. Second, we replace the causal VideoVAE with a custom-trained VideoVAE to address issues of temporal consistency and frame quality. Finally, we train the modified framework on a large-scale curated dataset with a multi-stage training strategy, achieving a commercial-level video generation model.

#### 4.1 Model Architecture

As shown in Figure 4, a typical VideoDiT framework comprises three key modules that collaboratively generate videos from textual descriptions: 1) a text encoder that converts natural language into text embeddings to guide video generation; 2) a VideoVAE that encodes videos into a latent space for efficient diffusion modeling; and 3) a video Transformer network that processes both visual tokens and text embeddings to predict noise for the diffusion generation.

**Text-Encoder.** The text encoder is essential in video diffusion models, translating natural languages into meaningful embeddings that guide video generation. Early models like Stable Diffusion [Rombach et al., 2022, Podell et al., 2024] adopted the CLIP text encoder [Radford et al., 2021], effectively aligning text with visual features. However, as tasks have become more complex, the T5 text encoder has gained popularity for its stronger language understanding and ability to produce richer embeddings, leading to better video quality and coherence. In Open-Sora-Plan v1.2.0, the mT5 text encoder offers improved multilingual compatibility, but the predominance of English captions in the training data limits the model’s efficiency in aligning textual semantics and visual information. To address these limitations, we adopt T5 as our text encoder based on prior visual generation research [Chen et al., 2024a, Saharia et al., 2022, DeepFloyd Lab at StabilityAI, 2024]. Experiments in Section 5.2 show that this choice significantly enhances the model’s semantic coherence, resulting in better text-to-video modeling. By leveraging T5, we achieve a more effective integration of textual and visual features, ultimately improving the quality and relevance of the generated videos.

**VideoVAE.** The VideoVAE module, crucial for compressing high-dimensional video data, is frequently employed in video diffusion models to improve efficiency. The idea of introducing VAEs into diffusion models was first proposed in Latent Diffusion Models (LDM) [Rombach et al., 2022], to enable effective training of text-to-image generation tasks. This integration has been expanded in subsequent efforts like Stable Diffusion [Esser et al., 2024, Podell et al., 2024]. Given the greater computational demands and information density of video data, we employ a VAE to compress videos before the VideoDiT module, significantly

Param.	T2I Pre-train	T2V Pre-train			T2V Fine-tune
Target GPUs	1×368×640 128×H100	40×368×640 256×H100	40×720×1280 256×H100	88×720×1280 256×H100	88×720×1280 256×H100
Batch-size	4,096	1,024	512	256	256
Steps	170K	85K	41K	31K	10K
Data	700M	87M	21M	8M	2.6M

Table 2: Training details and resource requirements for different stages.

enhancing training efficiency and convergence speed. As discussed in Section 3, we have developed a VideoVAE that outperforms existing open-source alternatives, offering superior visual quality and smoother motion. Our advanced VideoVAE can effectively encode videos, achieving a high compression ratio of  $4 \times 8 \times 8$ , while simultaneously preserving both motion dynamics and static details.

**Video Transformer.** The core of VideoDiT comprises a Video Transformer built from several Diffusion Transformer blocks (DiT blocks [Peebles and Xie, 2023, Chen et al., 2023]). Each DiT block includes these modules: 1) a self-attention module with 3D RoPE that models video tokens across both spatial and temporal dimensions; 2) a cross-attention module that injects textual conditions to guide the generation process; 3) a feed-forward layer; and 4) a set of AdaLN-single modules [Perez et al., 2018] that incorporate information from the diffusion timestep. The self-attention module is particularly critical for the quality of the generated videos. Earlier works tend to employ a combination of 2D spatial and 1D temporal attention to approximate full 3D attention. However, this limitation often results in videos with poor continuity and minimal motion. Recent research indicates that 3D attention significantly enhances video generation quality [PKU-Yuan Lab and Tuzhan AI etc., 2024, Polyak et al., 2024]. Adopting this strategy, we generate videos with improved consistency and dynamic motion.

More details about the model specifications can be found in Appendix B.

## 4.2 Multi-Stage Training Strategy

Our model training process can be divided into three stages: 1) the text-to-image pre-training stage, which establishes a mapping between text and images; 2) the text-to-video pre-training stage, which builds on this mapping to learn the motion relationships between objects; and 3) the text-to-video fine-tuning stage, which focuses on improving the overall visual quality of the generated videos. More details are listed in Table 2.

**Text-to-Image Pre-training.** The text-to-image stage aims to establish the connection between textual descriptions and visual elements. To improve training efficiency, we begin with the pre-trained text-to-image model from Open-Sora-Plan v1.2.0 and adjust the target image resolution to  $368 \times 640$ . We replace the text encoder with T5 (512 max token length) and the VAE with our custom-trained VideoVAE at this stage, as discussed in Section 4.1. To better align with the subsequent text-to-video training, we incorporate static video frames as images for mixed training. This stage is trained using 128 Nvidia H100 GPUs with a batch size of 4,096, completing 170K steps and processing a total of 700M images.

**Text-to-Video Pre-training.** We employ a three-stage progressive strategy to pre-train our text-to-video model. This pre-training is extended from our previous text-to-image model to learn dynamics from paired text-video data. First, we train with 40 frames at  $368 \times 640$  resolution to reduce the computational cost, utilizing 256 Nvidia H100 GPUs. This sub-stage has trained for 85K steps with a batch size of 1,024, processing a total of 87M videos, to equip the model with basic dynamics generation ability. Next, we scale up the resolution to  $720 \times 1280$  while keeping the video length. We resume from the previous sub-stage and train for 41K steps over 21M videos, using 256 Nvidia H100 GPUs with a batch size of 512 for 41K steps. After this stage, the model’s capabilities improved further, with the ability to generate higher-quality videos. However, compared to the detailed textual descriptions in the training data, the relatively short video duration can lead to some distortion and motion warping. In the final sub-stage, we increase the video length to 88 frames at  $720 \times 1280$ .

resolution. We use 256 Nvidia H100 GPUs with a batch size of 256, training for 31K steps and processing a total of 8M videos. All videos across the three sub-stages are sampled at 15 FPS, enabling the model to generate 6-second videos at  $720 \times 1280$  resolution.

**Text-to-Video Fine-tuning.** To further enhance video generation quality, we select a batch of high-quality dynamic videos and mix in various types of video captions for further fine-tuning. In this stage, we resume from the pre-trained model and train for 10K steps with 2.6M videos in total, using 256 Nvidia H100 GPUs with a batch size of 256. This fine-tuning enables the model to handle a wider variety of text inputs, both in length and format, and generate high-quality dynamic videos.

### 4.3 Infrastructure Optimization

We train our VideoDiT model using up to 256 Nvidia H100 GPUs with several optimization techniques.

**Decoupled Inference of VAE.** We observe that the VideoVAE processing and text encoding account for over 30% of the training computations. Since the inference process demands less VRAM and computation, we move the VideoVAE, text encoder to the mid-end inference GPUs, reserving the H100’s high performance and large VRAM for longer contexts and larger batch sizes in training. By JIT-compiling PyTorch code into optimized kernels with `torch.compile` and batching inference, we reduce the inference time of VideoVAE and text encoding during training by 62%. To avoid redundant computations, we cache the inference results offline for the high-quality data trained over multiple epochs.

**Efficient Attention.** We employ full bi-directional attention in the VideoDiT. For long sequences, attention accounts for 63.3% of computations during training and 84.5% during inference. We utilize FlashAttention2 [Dao, 2023] in self-attention and XFormers memory-efficient attention [Lefaudeux et al., 2022] (support mask) in cross-attention. Switching from XFormers to FlashAttention2 accelerates training in self-attention by 22%.

**Context Parallelism.** Our model supports 88 frames at  $720 \times 1280$  resolution video, which requires ~79.2K context length during training. To support longer context, we include Context Parallel (CP) in training. Instead of having a single device process the entire input, the input context is divided into smaller chunks, each processed in parallel by different devices. In *softmax-qk-attention* forward and backward, each device conducts necessary communications with other devices, ensuring that all devices have the full activations or gradients during computation. Ring-Attention [Liu et al., 2023] and DeepSpeed-Ulysses [Jacobs et al., 2023] provide efficient implementation. We also include CP in inference time.

## 5 Model Evaluation

We evaluate the VideoVAE and VideoDiT of our proposed Allegro model in Section 5.1 and Section 5.2, respectively.

### 5.1 Video Variational Auto-Encoder

Our VideoVAE and existing open-source 3D VAEs (such as Open-Sora v1.2 [Zheng et al., 2024], Open-Sora-Plan v1.2.0 [PKU-Yuan Lab and Tuzhan AI etc., 2024]) share the same latent channel  $C_l = 4$ , and employ identical compression strides of  $S_T \times S_H \times S_W = 4 \times 8 \times 8$  in the temporal, height, and width dimensions, respectively. We assess the performance of all available open-source VideoVAEs on a validation set comprising 100 videos, each with 120 frames and 720p resolution, utilizing metrics such as SSIM (Structural Similarity Index) [Wang et al., 2004], PSNR (Peak Signal-to-Noise Ratio) [Wang et al., 2004], and subjective evaluation criteria. PSNR [Wang et al., 2004] is an image quality evaluation metric based on the Mean Squared Error (MSE). It assesses the quality of an image by comparing the differences between the original image and the processed image. SSIM [Wang et al., 2004] is a quality evaluation metric based on structural similarity in images. It not only considers pixel-level differences but also considers the luminance, contrast, and structural information

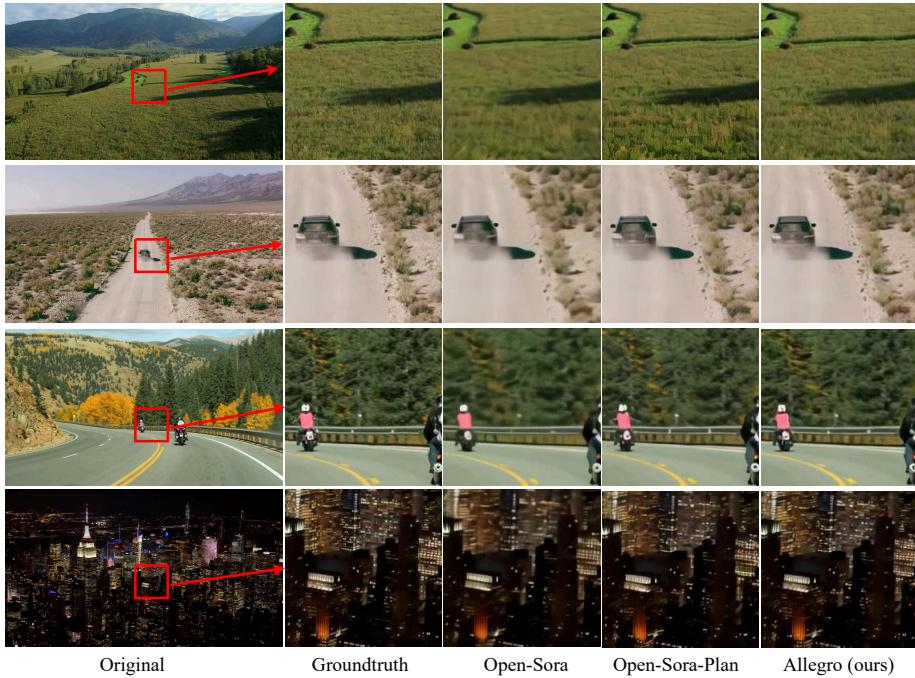


Figure 5: Visualization of Zoom-In on the sampled frame from reconstructed videos of VideoVAEs in our validation set.

Method	PSNR↑	SSIM↑
Open-Sora	30.64	0.8338
Open-Sora-Plan	27.65	0.8173
<b>Allegro (ours)</b>	<b>31.25</b>	<b>0.8553</b>

Table 3: Comparison of PSNR and SSIM among current open-source VideoVAE models.

of the image. For subjective evaluation criteria, we focus on identifying flickering and semantic distortions of videos that are not acceptable to human perception.

We observe that our VideoVAE model is highly sensitive to the temporal dimension of the tiles, with the best video reconstruction achieved when the tile duration matches the duration used during training. Deviations from the training duration lead to unpredictable flickering in the reconstructed videos. While we experiment with training on videos of varying lengths, this does not significantly enhance VideoVAE’s ability to model videos of different durations effectively. Fixing the video clip length to match the training duration yields the best performance during inference. Our analysis shows that when the clip length is too short, the model becomes overly sensitive, resulting in suboptimal performance.

As shown in Table 3, our VideoVAE outperforms other open-source models in both PSNR [Wang et al., 2004] and SSIM [Wang et al., 2004], achieving values of 31.25 and 0.8553, respectively. In subjective evaluations, the reconstructions from our model show reduced flickering and avoid issues of excessive sharpening or over-smoothing. The first column in Figure 5 displays the original frame, while the second column shows a zoomed-in view of the original frame. The third, fourth, and fifth columns provide zoomed-in visualizations using Open-Sora [Zheng et al., 2024], Open-Sora-Plan [PKU-Yuan Lab and Tuzhan AI etc., 2024], and our VideoVAE, respectively. In the first row, the lawn reconstruction demonstrates that our VideoVAE avoids both excessive sharpening and blurring, compared to other methods. In the third row, our model effectively preserves the shape and details of the helmet, while in the fourth row, it better retains the details of the skyscraper lights than the other methods.

Model	Total Score	Quality Score	Semantic Score	Subject Consist.	Background Consist.	Aesthetic Quality	Human Action	Scene
Open-Sora-Plan	78.00	80.91	66.38	95.73	96.73	56.85	86.8	27.17
Open-Sora	79.76	81.35	73.39	96.75	97.61	56.85	91.2	42.44
Pika	80.69	82.92	71.77	96.94	97.36	62.04	86.2	49.83
CogVideoX	80.91	82.18	75.83	96.78	96.63	60.82	98.0	51.14
Kling	81.85	83.39	75.68	98.33	97.60	61.21	93.4	50.86
Gen-3	82.32	84.11	75.17	97.10	96.62	63.34	96.4	54.57
Hailuo	83.41	84.85	77.65	97.51	97.05	63.03	92.4	50.68
<b>Allegro (ours)</b>	<b>81.09</b>	<b>83.12</b>	<b>72.98</b>	<b>96.33</b>	<b>96.74</b>	<b>63.74</b>	<b>91.4</b>	<b>46.72</b>

Table 4: Comparison of Allegro and other video generation models on VBench. The results are presented as percentages, with higher values indicating better performance. The version and date of each model can be found in the main text.

## 5.2 Video Diffusion Transformer

We evaluate the performance of our proposed Allegro model in this section. As our model outputs videos with 88 frames in 15 FPS and 720p resolution, we adopt a video frame interpolation model [Zhang et al., 2023] to enhance the FPS to 30 for better visual quality.

**Quantitative Evaluation.** We use VBench [Huang et al., 2024] for our quantitative evaluation, running our model on the 946 text prompts provided by VBench. To reduce randomness, we generate five videos for each prompt, following VBench’s guidelines. For prompts containing fewer than 9 words, we utilize a large language model (LLM) to refine them, enhancing robustness. In total, we generated 4,730 videos for the evaluation. Given the large number of videos required by VBench, we directly use the results from their leaderboard for other models.

As shown in Table 4, we compare our model with Open-Sora v1.2 (8s), Open-Sora-Plan v1.1.0, CogVideoX-2B, Pika 1.0 (Jun. 2024), Gen-3 (Jul. 2024), Hailuo (MiniMax-Video-01) and Kling (Jul. 2024, high-performance mode). The results show that our method outperforms other open-source methods and ranks just behind Gen-3, Kling, and Hailuo among commercial models. For a detailed evaluation of all dimensions in VBench, please refer to the Appendix C.

**Qualitative Evaluation.** Evaluating video generation results is a highly subjective task, and traditional quantitative metrics often show a significant deviation from the subjective experiences of human users. Therefore, we further conduct a user study to assess the subjective experience of the models.

To ensure the accuracy of the evaluation, we collect 46 diverse text inputs. These texts are from different sources: 1) text prompts from open-source models; 2) text prompts from commercial models; 3) captions generated by video captioning models; 4) texts constructed by large language models (LLMs) for video generation; and 5) real user inputs. The scenarios described in the texts cover a wide range, including people, animals, food, vehicles, landscapes, street scenes, sci-fi, and multi-object scenarios. The text lengths varied from five words to 96 words.

We compare our model with open-sourced models including Open-Sora-Plan v1.2.0, Open-Sora v1.2, CogVideoX-2B and commercial models including PixVerse v2 (Aug. 2024), Pika 1.0 (Aug. 2024), DreamMachine v1.6 (Aug. 2024), Vidu (Sept. 2024), Gen-3 (Sept. 2024), Hailuo (Sept. 2024) and Kling v1.5 (Sept. 2024). To simplify the naming, we will no longer include dates or version numbers in the names in the following sections.

To minimize ambiguity and obtain relatively accurate labeling results, we design six evaluation dimensions for annotators: 1) video-text relevance; 2) appearance distortion; 3) appearance aesthetics; 4) motion naturalness; 5) motion amplitude; and 6) overall quality. The evaluation is conducted using an A-B test format, where subjects are asked to rate whether A is better, B is better, or whether A and B are indistinguishable. For each trial, we randomly present results from our method and other methods, along with the correspond-

Dimensions	V.-T. Relevance			Appearance Dist.			Appearance Aest.		
Methods	Win	Lose	Tie	Win	Lose	Tie	Win	Lose	Tie
Open-Sora-Plan	<b>96%</b>	0%	4%	<b>57%</b>	11%	32%	<b>89%</b>	0%	11%
Open-Sora	<b>78%</b>	9%	13%	<b>87%</b>	4%	9%	<b>96%</b>	4%	0%
CogVideoX	<b>83%</b>	4%	13%	<b>67%</b>	7%	26%	<b>96%</b>	4%	0%
PixVerse	<b>57%</b>	2%	41%	<b>46%</b>	17%	37%	<b>61%</b>	11%	28%
Pika	<b>72%</b>	2%	26%	<b>43%</b>	20%	37%	<b>72%</b>	9%	20%
Dream Machine	<b>33%</b>	22%	45%	<b>39%</b>	13%	48%	17%	35%	48%
Vidu	<b>27%</b>	15%	58%	<b>27%</b>	22%	51%	<b>51%</b>	29%	20%
Gen-3	<b>42%</b>	18%	40%	<b>29%</b>	20%	51%	<b>45%</b>	33%	22%
Hailuo	<b>33%</b>	25%	42%	14%	26%	60%	26%	56%	18%
Kling	<b>29%</b>	27%	44%	4%	36%	60%	5%	82%	13%

Dimensions	Motion Nat.			Motion Amp.			Overall Quality		
Methods	Win	Lose	Tie	Win	Lose	Tie	Win	Lose	Tie
Open-Sora-Plan	<b>44%</b>	17%	39%	<b>98%</b>	2%	0%	<b>100%</b>	0%	0%
Open-Sora	<b>94%</b>	2%	4%	<b>65%</b>	33%	2%	<b>98%</b>	2%	0%
CogVideoX	<b>76%</b>	9%	15%	<b>74%</b>	24%	2%	<b>96%</b>	4%	0%
PixVerse	<b>50%</b>	13%	37%	<b>76%</b>	9%	15%	<b>96%</b>	4%	0%
Pika	<b>43%</b>	17%	39%	<b>91%</b>	4%	4%	<b>89%</b>	11%	0%
Dream Machine	<b>28%</b>	13%	59%	41%	48%	11%	<b>61%</b>	28%	11%
Vidu	<b>40%</b>	33%	27%	18%	60%	22%	<b>51%</b>	42%	7%
Gen-3	<b>24%</b>	20%	56%	38%	58%	4%	<b>49%</b>	40%	11%
Hailuo	14%	21%	65%	35%	46%	19%	47%	49%	4%
Kling	13%	36%	51%	13%	76%	11%	22%	76%	2%

Table 5: User study on our benchmark with 46 text prompts. The performance is assessed in six dimensions: video-text relevance, appearance distortion, appearance aesthetics, motion naturalness, motion amplitude, and overall quality. The version and date of each model can be found in the main text. The winning dimension is highlighted in **bold**.

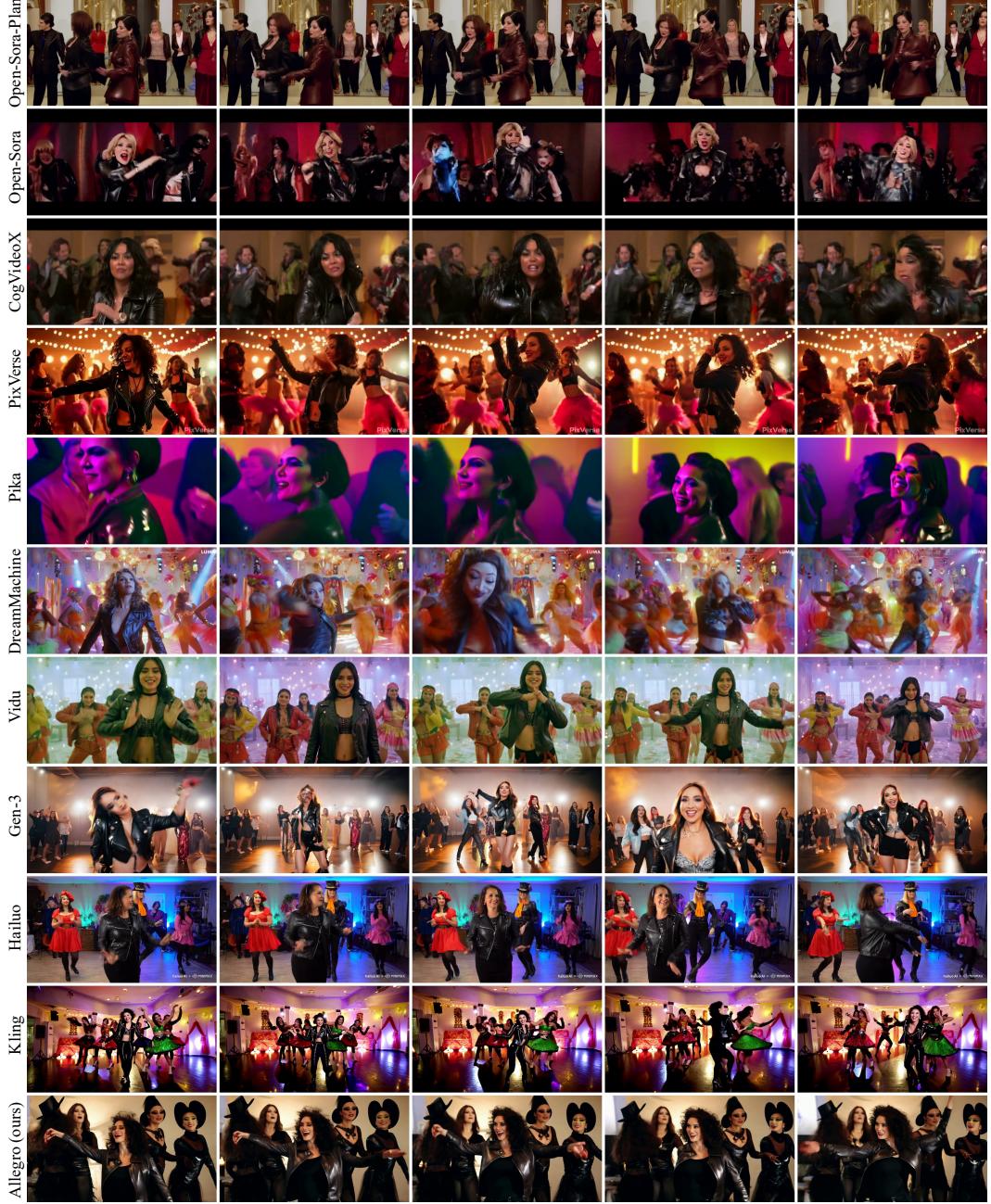
ing text, and ask the annotators to rate them. Each video pair is rated by two subjects, and we collect a total of 5,448 ratings.

As shown in Table 5, our model surpasses the current open-source models across all six dimensions. Compared to commercial models, our model demonstrates significant advantages in most aspects. Notably, in terms of video-text relevance, our model outperforms all commercial models. For the dimension of overall quality, our model outperforms most commercial models, ranking just behind Hailuo and Kling. However, the result also shows that there is still room for improvement in our model’s handling of large-scale motion compared to the top-performing commercial models. This can typically be addressed by increasing the amount of accelerated motion in the training data or further filtering the speed of the videos used for training. Increasing the model’s parameter size may also help improve its ability to model large-scale motion.

We show the visual result of our method and other methods in Figure 6. We can see that our model surpasses other models in both video-text relevance, video appearance quality, and video motion naturalness. More comparison results can be found in Appendix D.

**Text-Encoder.** As discussed in Section 4.1, we compare our fine-tuned text-to-image model with T5 text-encoder with the original text-to-image model provided in Open-Sora-Plan that uses mT5 text encoder. As shown in Figure 7, images generated with Open-Sora-Plan lack visual details and have less relevance to the given text inputs.

Some might speculate that the differences in performance are due to variations in data usage. To address this, we further fine-tuned the Open-Sora-Plan text-to-video model using the same dataset. As shown in the second row in Figure 8, even with extensive fine-tuning on high-quality data, the semantic modeling issues caused by using the mT5 text encoder in a fully English training dataset remain unresolved. From Figure 7 and the third row in Figure 8, we can observe that replacing the text encoder with T5 and training as described



A woman dancing with other dancers in costume, likely at a themed event or a performance. The woman is wearing a leather jacket, and the group of people is engaged in a lively dance routine. The scene is set in a room, and the atmosphere appears to be energetic and fun. The video captures a moment of enjoyment and camaraderie among the dancers as they perform together.

Figure 6: Qualitative comparisons of our Allegro with SOTA methods for the user study.

in Section 4.2 on high-quality data leads to significant improvements in both text-to-image and text-to-video tasks.

The results also indicate that it is challenging to obtain a high-quality video model by directly fine-tuning the Open-Sora-Plan text-to-video model. This is primarily due to the inaccurate alignment between text and visual elements established during the initial text-to-image pre-training stage. Since the text-to-video training process mainly focuses on modeling motion, this issue is difficult to address through text-to-video fine-tuning alone.

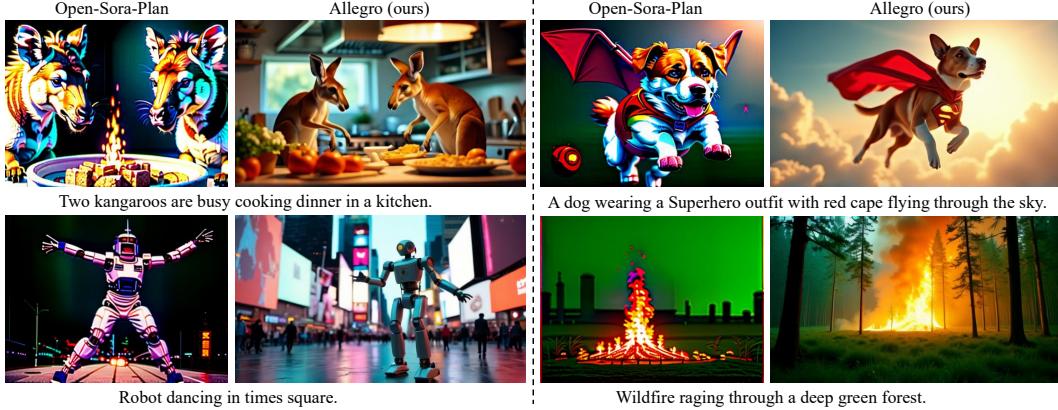


Figure 7: Qualitative comparison of generated images from our Allegro and the Open-Sora-Plan model in text-to-image generation.



Figure 8: Qualitative comparison of generated frames from our Allegro and the Open-Sora-Plan model in text-to-video generation.

**Multi-Stage Training.** As discussed in Section 4.2, we train our model with a coarse-to-fine multi-stage training strategy. As shown in Figure 9. In the initial  $40 \times 368 \times 640$  stage, the model is initialized from the pre-trained text-to-image model and begins learning the continuity between video frames, resulting in relatively small-scale motion in the generated videos. Constrained by the VideoVAE’s capacity for modeling at low resolutions, the overall video quality remains low, with noticeable jitter in the motion of finer details. In the subsequent  $40 \times 720 \times 1280$  stage, the increase in resolution significantly improves the visual quality of the generated videos. The model starts to better capture motion between frames, but due to the limited number of frames, there are still distortions and unnatural motion in the videos. Moving forward to the  $88 \times 720 \times 1280$  stage, the increased frame count greatly enhances the model’s ability to represent large-scale motion. Finally, by fine-tuning the model with a small amount of high-quality data at the same video size, the model’s overall visual aesthetics, motion dynamics, and responsiveness to different types of text inputs are further improved.

**Visual Results.** We show visual results of our proposed model, Allegro, in Figure 10. These results demonstrate that our model is capable of handling text inputs of varying lengths while generating high-quality, dynamic videos. More visual results can be found in Appendix E.

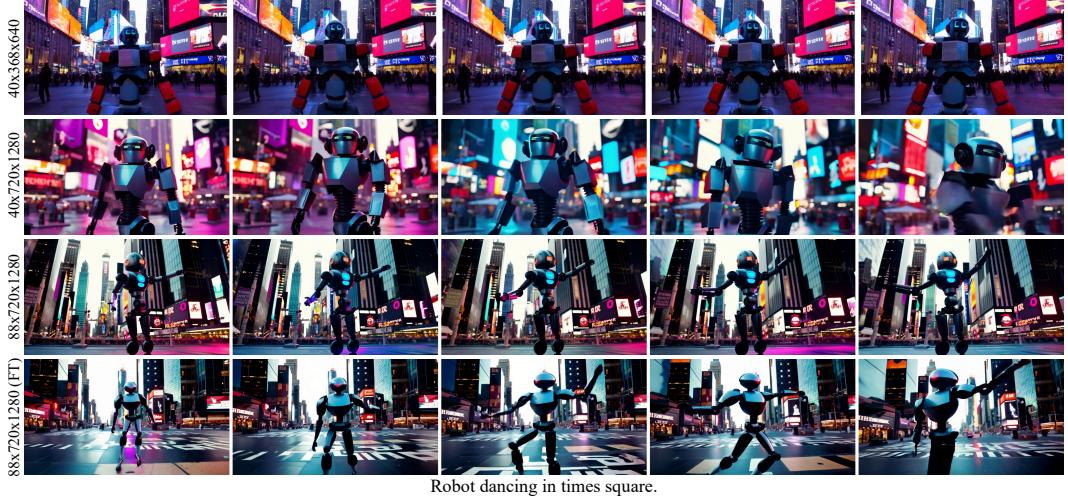


Figure 9: Qualitative comparison of generated video frames from our Allegro in different training stages.

## 6 More Thinkings

**VideoVAE.** Our VideoVAE, built on the architecture of a previous image VAE, incorporates numerous temporal CNN layers and applies overlapping and blending in both spatial and temporal dimensions, leading to extremely high computational costs.

For future developments, it will be important to strike a balance between reconstruction quality and computational efficiency when designing the spatiotemporal modeling structure of VideoVAE. Additionally, to improve latent feature extraction, we should avoid spatial tiling and increase the spatiotemporal unit size during video encoding.

Under the same model parameter constraints and compression stride, increasing the latent channel size is a simple yet effective approach to enhancing the detail in reconstructed videos. However, reconstructing line-like semantics remains a challenging task for VideoVAE models that rely on RGB images. We believe that incorporating frequency domain features into the input of VideoVAE could potentially improve its ability to reconstruct line-like semantics in the future.

**VideoDiT.** We have trained the current model with a sufficiently large dataset. However, as training progresses, improvements in certain areas have become limited. For example, the model demonstrates inadequate fine-grained instruction-following abilities, less natural interactions in more complex subject scenarios, and limited further enhancement in visual textures. Drawing from the experiences of LLMs and vision transformer models, increasing the model’s parameter count is a direct and promising approach to address these limitations. For instance, expanding the model’s width enhances its capacity to capture more complex visual patterns. Increasing depth enables it to learn hierarchical representations. These two approaches are both crucial for understanding intricate subject interactions and generating more refined textures.

Although we have not performed comprehensive, large-scale experiments to explore the scaling limits in our model, we have conducted small-scale ablation studies. These experiments allow us to investigate the effects of increasing width and depth on a manageable scale, and the results suggest that further exploration of model scaling could yield promising improvements. Therefore, our findings offer some practical guidance for researchers and engineers in the open-source community who are interested in scaling up models. They can consider methods such as expanding the width and depth of their architectures, which have been proven effective in both the LLM and vision transformer domains. These insights can serve as a starting point for more in-depth, resource-intensive studies aimed at unlocking further advancements in model performance.



A vampire running inside a castle.



Noodles falling into a bowl of soup.



The sun breaks through the clouds from the heights of a skyscraper.



A woman is hiking up a hill with her adventurous cat perched on her shoulder, the cat's tail flicking back and forth.



A nimble dirt bike races along a rugged trail, kicking up clouds of dust as it navigates sharp turns and steep hills.



A shy deer cautiously steps out of the forest, its eyes scanning the meadow as it takes delicate steps forward.



A vintage steam locomotive chugs along a mountainous railway, releasing puffs of white smoke into the crisp air.



A close-up of a chef meticulously preparing sushi, with each movement precise and deliberate. The camera captures the texture of the rice, the shine of the fish, and the delicate placement of each ingredient.



A person holding a guitar in their lap. The person is sitting on a couch, and the guitar is placed close to their body.

The scene suggests that the person might be practicing or learning a new song on the guitar.



A majestic eagle perched on an urban rooftop. The scene is set during a misty morning fog over a valley, with ambient lighting softly illuminating the bird. The eagle gracefully surveys its domain, embodying power and elegance amidst the urban landscape, all set against the backdrop of a tranquil, foggy morning.

Figure 10: The generated frames of our Allegro with text inputs of varying lengths.

**Prompt Refiner.** A general prompt refiner interprets, adjusts, or expands the given prompt to better match the model’s capabilities; in other words, adjust the given prompt’s distri-

bution to match the distribution of training data captions. Considering that our training pipeline does not include training for text encoders, we argue that structuring the caption in training data can lower the complexity of training and lead to faster convergence. All the captions or prompts follow a specific manner, as mentioned in Section 2.1.2. Through structuralization, the general text-to-video task can be simplified to label-to-video. However, such a simplification inevitably harms generalization abilities, and that’s why a prompt refiner is needed.

For large-scale pretraining and better generalization ability, we suggest enhancing the training data’s diversity and reducing the model’s dependency on the refiner. To be specific, annotation methods, including leveraging multiple captioners with different linguistic styles, mixing the data of real use cases, and selecting the data based on human preference, are all good for generalization. This aspect of the work is left for future exploration.

## 7 Future Work

We are working on several features as our future works to enhance the capability of our proposed Allegro text-to-video model.

**Image-to-Video.** An image-to-video generation model with text condition, i.e. Text-Image-to-Video, can be built upon the foundation of a well-trained text-to-video model, adding the ability to incorporate visual context (images) alongside textual input. This enhancement allows the model to generate more accurate and visually rich videos by combining the descriptive power of text with the specific visual cues provided by images. The necessity of such a model arises from the limitations of text-only inputs, which can sometimes be ambiguous or lack detailed visual context. By integrating images, the model can create more precise, contextually relevant, and higher-quality videos, bridging the gap between user intent and the generated output. This is particularly valuable in commercial applications, where brands and businesses often need to convey specific visual themes or brand elements that may not be fully captured through text alone. Our text-image-to-video model is under training, based on our foundational text-to-video model, and will be released as another extended model of the Allegro family in the near future.

**Motion Control.** Under the combined introduction of text descriptions and appearance prior conditions, many video producers still have some more precise control requirements that are difficult to specify with text, among which the most notable is the motion control of the area of interest in the first frame. Our motion control-related data has been preliminarily prepared, and the motion control model is currently being trained. We expect to release it subsequently to provide everyone with more fine-grained control options.

## 8 Acknowledgement

The authors appreciate Dongxu Li, Hanshu Yan, Haoning Wu, Huiguo He, Ruoyu Li, Wenhao Huang, Xin Yan, Yifan Ye, and Zixi Tuo (in alphabet order) for their valuable input and suggestions.

## References

- Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee. Character region awareness for text detection. In *CVPR*, pages 9365–9374, 2019.
- M. Bain, A. Nagrani, G. Varol, and A. Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *ICCV*, pages 1728–1738, 2021.
- A. Blattmann, R. Rombach, H. Ling, T. Dockhorn, S. W. Kim, S. Fidler, and K. Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *CVPR*, pages 22563–22575, 2023.
- J. Chen, J. Yu, C. Ge, L. Yao, E. Xie, Y. Wu, Z. Wang, J. Kwok, P. Luo, H. Lu, et al. PixArt- $\alpha$ : Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*, 2023.

- S. Chen, M. Xu, J. Ren, Y. Cong, S. He, Y. Xie, A. Sinha, P. Luo, T. Xiang, and J.-M. Perez-Rua. GenTron: Diffusion transformers for image and video generation. In *CVPR*, pages 6441–6451, 2024a.
- T.-S. Chen, A. Siarohin, W. Menapace, E. Deyneka, H.-w. Chao, B. E. Jeon, Y. Fang, H.-Y. Lee, J. Ren, M.-H. Yang, et al. Panda-70M: Captioning 70M videos with multiple cross-modality teachers. In *CVPR*, pages 13320–13331, 2024b.
- T. Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023.
- DeepFloyd Lab at StabilityAI. DeepFloyd IF. <https://github.com/deep-floyd/IF?tab=readme-ov-file/>, 2024.
- N. Dinesh Reddy, R. Tamburo, and S. G. Narasimhan. Walt: Watch and learn 2d amodal representation from time-lapse imagery. In *CVPR*, 2022.
- P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024.
- H. He, H. Yang, Z. Tuo, Y. Zhou, Q. Wang, Y. Zhang, Z. Liu, W. Huang, H. Chao, and J. Yin. DreamStory: Open-domain story visualization by llm-guided multi-subject consistent diffusion. *arXiv preprint arXiv:2407.12899*, 2024.
- J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 33:6840–6851, 2020.
- J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet. Video diffusion models. *NeurIPS*, 35:8633–8646, 2022.
- X. Huang, Y. Zhang, J. Ma, W. Tian, R. Feng, Y. Zhang, Y. Li, Y. Guo, and L. Zhang. Tag2Text: Guiding vision-language model via image tagging. *arXiv preprint arXiv:2303.05657*, 2023.
- Z. Huang, Y. He, J. Yu, F. Zhang, C. Si, Y. Jiang, Y. Zhang, T. Wu, Q. Jin, N. Chanpaisit, et al. VBench: Comprehensive benchmark suite for video generative models. In *CVPR*, pages 21807–21818, 2024.
- S. A. Jacobs, M. Tanaka, C. Zhang, M. Zhang, S. L. Song, S. Rajbhandari, and Y. He. DeepSpeed Ulysses: System optimizations for enabling training of extreme long sequence transformer models. *arXiv preprint arXiv:2309.14509*, 2023.
- D. P. Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- Kuaishou. Kling AI. <https://klingai.kuaishou.com>, 2024.
- B. Lefauideux, F. Massa, D. Liskovich, W. Xiong, V. Caggiano, S. Naren, M. Xu, J. Hu, M. Tintore, S. Zhang, P. Labatut, D. Haziza, L. Wehrstedt, J. Reizenstein, and G. Sizov. xFormers: A modular and hackable transformer modelling library. <https://github.com/facebookresearch/xformers>, 2022.
- D. Li, A. Kamko, E. Akhgari, A. Sabet, L. Xu, and S. Doshi. Playground v2.5: Three insights towards enhancing aesthetic quality in text-to-image generation. *arXiv preprint arXiv:2402.17245*, 2024a.
- D. Li, Y. Liu, H. Wu, Y. Wang, Z. Shen, B. Qu, X. Niu, G. Wang, B. Chen, and J. Li. Aria: An open multimodal native mixture-of-experts model, 2024b. URL <https://arxiv.org/abs/2410.05993>.
- H. Liu, M. Zaharia, and P. Abbeel. Ring attention with blockwise transformers for near-infinite context. *arXiv preprint arXiv:2310.01889*, 2023.
- Y. Liu, X. Cun, X. Liu, X. Wang, Y. Zhang, H. Chen, Y. Liu, T. Zeng, R. Chan, and Y. Shan. EvalCrafter: Benchmarking and evaluating large video generation models. In *CVPR*, pages 22139–22149, 2024a.

- Y. Liu, L. Li, S. Ren, R. Gao, S. Li, S. Chen, X. Sun, and L. Hou. FETV: A benchmark for fine-grained evaluation of open-domain text-to-video generation. *NeurIPS*, 36, 2024b.
- H. Lu, G. Yang, N. Fei, Y. Huo, Z. Lu, P. Luo, and M. Ding. VDT: General-purpose video diffusion transformers via mask modeling. *arXiv preprint arXiv:2305.13311*, 2023.
- Luma Lab. Dream Machine. <https://lumalabs.ai/dream-machine>, 2024.
- X. Ma, Y. Wang, G. Jia, X. Chen, Z. Liu, Y.-F. Li, C. Chen, and Y. Qiao. Latte: Latent diffusion transformer for video generation. *arXiv preprint arXiv:2401.03048*, 2024.
- MiniMax. Hailuo AI. <https://hailuoai.video>, 2024.
- K. Nan, R. Xie, P. Zhou, T. Fan, Z. Yang, Z. Chen, X. Li, J. Yang, and Y. Tai. OpenVid-1M: A large-scale high-quality dataset for text-to-video generation. *arXiv preprint arXiv:2407.02371*, 2024.
- OpenAI. Sora. <https://openai.com/index/sora>, 2024.
- W. Peebles and S. Xie. Scalable diffusion models with transformers. In *ICCV*, pages 4195–4205, 2023.
- E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville. FiLM: Visual reasoning with a general conditioning layer. In *AAAI*, pages 3942–3951, 2018.
- Pika Labs. Pika. <https://pika.art>, 2024.
- PKU-Yuan Lab and Tuzhan AI etc. Open-Sora-Plan. <https://doi.org/10.5281/zenodo.10948109>, Apr. 2024.
- D. Podell, Z. English, K. Lacey, A. Blattmann, T. Dockhorn, J. Müller, J. Penna, and R. Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *ICLR*, 2024.
- A. Polyak, A. Zohar, A. Brown, A. Tjandra, A. Sinha, A. Lee, A. Vyas, B. Shi, C.-Y. Ma, C.-Y. Chuang, D. Yan, D. Choudhary, D. Wang, G. Sethi, G. Pang, H. Ma, I. Misra, J. Hou, J. Wang, K. Jagadeesh, K. Li, L. Zhang, M. Singh, M. Williamson, M. Le, M. Yu, M. K. Singh, P. Zhang, P. Vajda, Q. Duval, R. Girdhar, R. Sumbaly, S. S. Rambhatla, S. Tsai, S. Azadi, S. Datta, S. Chen, S. Bell, S. Ramaswamy, S. Sheynin, S. Bhattacharya, S. Motwani, T. Xu, T. Li, T. Hou, W.-N. Hsu, X. Yin, X. Dai, Y. Taigman, Y. Luo, Y.-C. Liu, Y.-C. Wu, Y. Zhao, Y. Kirstain, Z. He, Z. He, A. Pumarola, A. Thabet, A. Sanakoyeu, A. Mallya, B. Guo, B. Araya, B. Kerr, C. Wood, C. Liu, C. Peng, D. Vengertsev, E. Schonfeld, E. Blanchard, F. Juefei-Xu, F. Nord, J. Liang, J. Hoffman, J. Kohler, K. Fire, K. Sivakumar, L. Chen, L. Yu, L. Gao, M. Georgopoulos, R. Moritz, S. K. Sampson, S. Li, S. Parmeggiani, S. Fine, T. Fowler, V. Petrovic, and Y. Du. Movie gen: A cast of media foundation models, 2024. URL <https://arxiv.org/abs/2410.13720>.
- PySceneDetect Contributors. PySceneDetect. <https://www.scenedetect.com>, 2024.
- A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. PMLR, 2021.
- C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 21(140):1–67, 2020.
- R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022.
- L. Ruan, Y. Ma, H. Yang, H. He, B. Liu, J. Fu, N. J. Yuan, Q. Jin, and B. Guo. MM-Diffusion: Learning multi-modal diffusion models for joint audio and video generation. In *CVPR*, pages 10219–10228, 2023.

- RunwayML. Gen-3 Alpha. <https://runwayml.com/research/introducing-gen-3-alpha>, 2024.
- C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Goncalves Lopes, B. Karagol Ayan, T. Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *NeurIPS*, 35:36479–36494, 2022.
- C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, et al. LAION-5B: An open large-scale dataset for training next generation image-text models. *NeurIPS*, 35:25278–25294, 2022.
- U. Singer, A. Polyak, T. Hayes, X. Yin, J. An, S. Zhang, Q. Hu, H. Yang, O. Ashual, O. Gafni, et al. Make-A-Video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022.
- K. Sun, K. Huang, X. Liu, Y. Wu, Z. Xu, Z. Li, and X. Liu. T2V-CompBench: A comprehensive benchmark for compositional text-to-video generation. *arXiv preprint arXiv:2407.14505*, 2024.
- V. Voleti, A. Jolicoeur-Martineau, and C. Pal. MCVD: Masked conditional video diffusion for prediction, generation, and interpolation. *NeurIPS*, 35:23371–23385, 2022.
- W. Wang, H. Yang, Z. Tuo, H. He, J. Zhu, J. Fu, and J. Liu. VideoFactory: Swap attention in spatiotemporal diffusions for text-to-video generation. *arXiv preprint arXiv:2305.10874*, 2023.
- X. Wang, H. Yuan, S. Zhang, D. Chen, J. Wang, Y. Zhang, Y. Shen, D. Zhao, and J. Zhou. VideoComposer: Compositional video synthesis with motion controllability. *NeurIPS*, 36, 2024.
- Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIP*, 13(4):600–612, 2004.
- Watermark-Detection Contributors. Watermark-Detection. <https://github.com/boomb0om/watermark-detection>, 2022.
- H. Wu, E. Zhang, L. Liao, C. Chen, J. Hou, A. Wang, W. Sun, Q. Yan, and W. Lin. Exploring video quality assessment on user generated contents from aesthetic and technical perspectives. In *ICCV*, pages 20144–20154, 2023.
- H. Xu, J. Zhang, J. Cai, H. Rezatofighi, F. Yu, D. Tao, and A. Geiger. Unifying flow, stereo and depth estimation. *TPAMI*, 2023.
- H. Xue, T. Hang, Y. Zeng, Y. Sun, B. Liu, H. Yang, J. Fu, and B. Guo. Advancing high-resolution video-language representation with large-scale video transcriptions. In *CVPR*, pages 5036–5045, 2022.
- L. Xue. mT5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*, 2020.
- Z. Yang, J. Teng, W. Zheng, M. Ding, S. Huang, J. Xu, Y. Yang, W. Hong, X. Zhang, G. Feng, et al. CogVideoX: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024.
- G. Zhang, Y. Zhu, H. Wang, Y. Chen, G. Wu, and L. Wang. Extracting motion and appearance via inter-frame attention for efficient video frame interpolation. In *CVPR*, pages 5682–5692, 2023.
- R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018.
- Z. Zheng, X. Peng, T. Yang, C. Shen, S. Li, H. Liu, Y. Zhou, T. Li, and Y. You. Open-Sora: Democratizing efficient video production for all. <https://github.com/hpcaitech/Open-Sora>, March 2024.

## A Data Analysis

In this section, we conduct a statistical analysis of the training data, showing the statistical distribution across various dimensions. We compare the distribution differences in different training stages. Additionally, specific examples are shown to demonstrate how the data filtering process improves data quality.

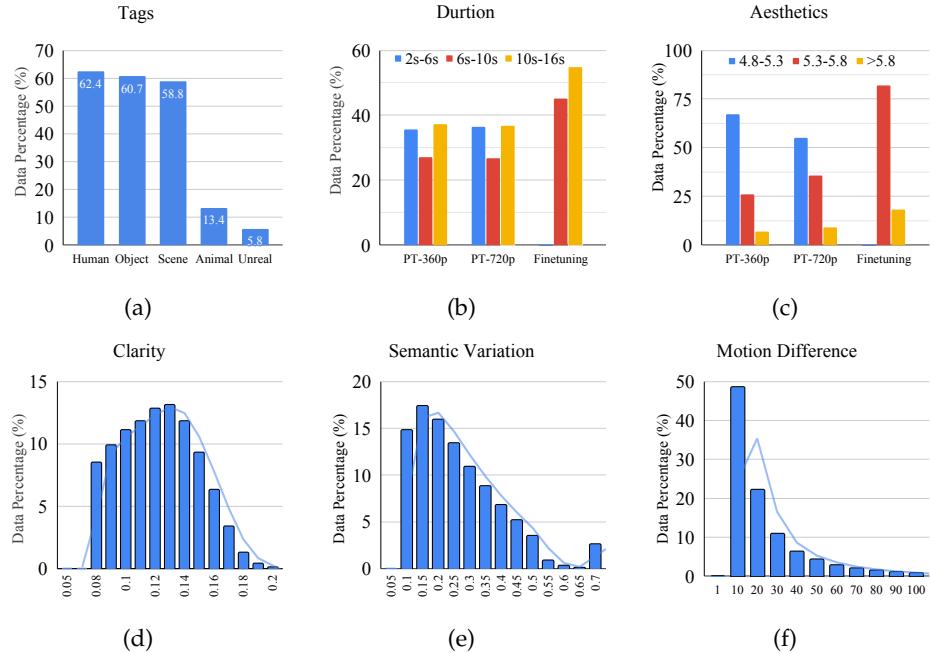


Figure 11: Data statistics in different dimensions. (a) Distribution of different tags in the training data. (b) and (c) shows the duration and aesthetic score of the data used in different training stages, respectively. (d)-(f) shows the data distributions of clarity, semantic variation, and motion difference, estimated with the scores of DOVER, LPIPS, and UniMatch, respectively.

**Video duration, variation, and consistency.** Figure 11b shows how the data is distributed based on video duration across Text-to-Video training stages. The data are categorized into three duration buckets: 2s-6s (short-duration), 6s-10s (medium-duration), and 10s-16s (long-duration). The pre-training stages show a balanced distribution across the three buckets in both 360p and 720p. In contrast, the fine-tuning stage shifts focus entirely to medium and long-duration videos, with no short-duration videos being used at this stage. In the early stages of training, the model focuses more on learning a large amount of semantic information. Single-frame images or videos with minimal motion and semantic changes play a more important role. As training progresses, there will be a greater demand for diversity in the richness of video content and the range of motion, which inevitably requires longer videos. Additionally, it is also necessary to further quantify the degree of content variation and motion within the videos. Therefore, by introducing the evaluation metrics of the DOVER score, LPIPS score, and UniMatch score, it is possible to better control the selection of data that exhibits these characteristics. To improve data processing efficiency, these three metrics are only calculated during the fine-tuning stage. By jointly adjusting the thresholds of these three metrics, we obtained data with high clarity but moderate changes and motion during the fine-tuning stage. The distribution of clarity, semantic variation, and motion are shown in Figure 11d, 11e and 11f, respectively. This suggests that longer videos with appropriate variations become more important as the model approaches final tuning to improve the model on long-duration video generation capability by this progressive training recipe.

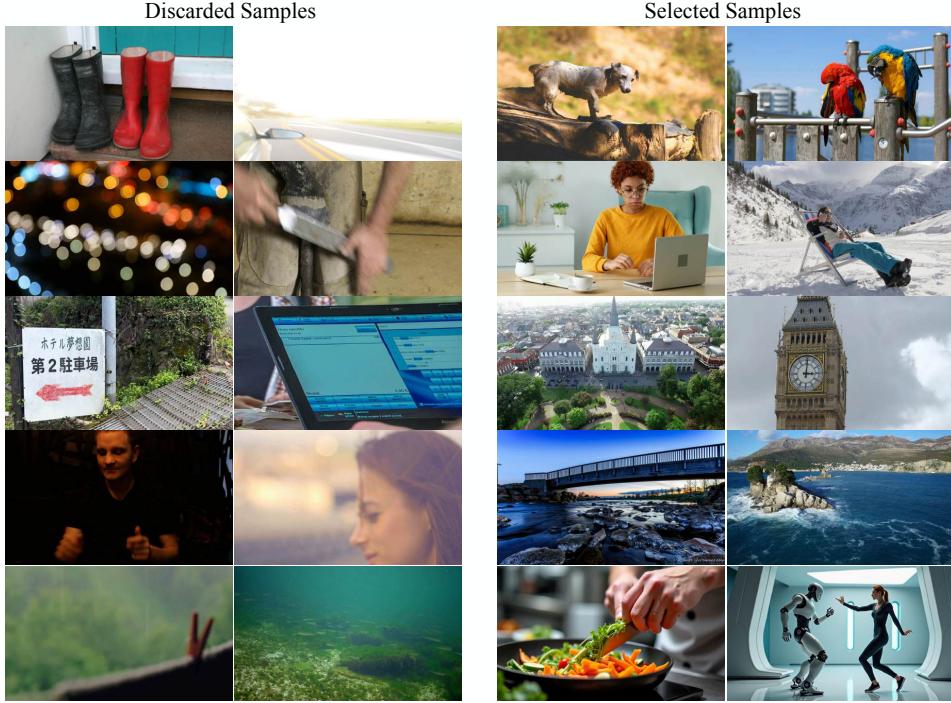


Figure 12: The selected and discarded data samples by the data filtering. The discarded samples include blurry samples, poorly focused, large text areas, or lack of clear subject matter, such as a pair of boots, unfocused city lights, and random objects. The selected samples are sharper, more aesthetically appealing, and have well-defined subjects, including vibrant scenes of nature, architecture, and human activities like cooking or working on a laptop.

**Resolution and Aesthetics.** Figure 11c shows the distribution of aesthetic scores of the images/videos used during the Text-to-Video pre-training at 360p, 720p, and the fine-tuning stages. Resolution during training impacts aesthetic quality but alone is not enough to achieve high levels of aesthetics. fine-tuning is crucial in boosting the model’s ability to generate more aesthetically pleasing content. To achieve higher-quality aesthetics in video generation, incorporating a fine-tuning step after pre-training at higher resolutions (like 720p) is essential. This ensures the model learns to focus not just on structural aspects but on visual appeal. Aesthetic filtering can help in curating training datasets for fine-tuning, ensuring the model receives feedback on what constitutes high-quality, aesthetically pleasing video content.

We provide some training examples from different training stages to visually demonstrate the differences in data quality before and after the data filtering pipeline, shown in Figure 12. The discarded samples include blurry samples, poorly focused, large text areas, or lack of clear subject matter, such as a pair of boots, unfocused city lights, and random objects. The selected samples are sharper, more aesthetically appealing, and have well-defined subjects, including vibrant scenes of nature, architecture, and human activities like cooking or working on a laptop. The overall quality and clarity of the selected samples are visibly superior.

**Tags and Annotations** Figure 11a shows the proportion of different tags in the training data, which are derived from the coarse-grained captions obtained during data filtering. We can observe that, due to the natural distribution of the data, content related to people, objects, and landscapes occupies the majority. On the one hand, this helps the model better understand the semantics and motion patterns of the physical world by aligning with real-world data distributions. On the other hand, it enables the model to more effectively respond to instructions that are close to real-life scenarios, which is a crucial capability for a commercial-level video generation model.



**Mid-frame caption:** A gray cat is sitting next to a beige cat tree with a red sign that reads "DO NOT DISTURB". **The cat appears to be looking at the sign with a grumpy expression.**

**Video caption:** A gray cat interacting with a red "Do Not Disturb" sign attached to a beige cat tree. The camera captures the scene from an eye-level shot, focusing on the cat's movements as **it curiously sniffs and examines the sign**. The background is a simple indoor setting with neutral lighting, creating a calm and cozy atmosphere. Throughout the video, **the cat moves closer to the sign**, occasionally pawing at it, while the camera remains steady, maintaining a clear view of the interaction.



**Mid-frame caption:** A bird's-eye view of a field of tall, green wheat with a brown dog walking through it. The wheat **appears to be swaying slightly**, creating a pattern of light and shadow across the landscape. **The dog is in motion**, with its legs and tail visible amidst the sea of wheat.

**Video caption:** A German Shepherd **dog running** through a lush, green field of tall grass. The camera captures the scene from a bird's-eye view, \textcolor{blue}{following the dog's movement as it weaves through the dense vegetation}. The lighting is natural, with sunlight casting soft shadows on the grass, creating a vibrant and lively atmosphere. The background consists entirely of the expansive field, emphasizing the freedom and energy of the dog's run. The overall **photography style is dynamic and immersive**, providing a sense of motion and excitement as the dog explores its surroundings.



**Mid-frame caption:** A brown horse with a blonde mane grazes in a field of green grass and leafy vines. The horse is wearing a red halter and **is standing still while eating**. The background is slightly blurred, with trees and foliage visible in the distance.

**Video caption:** A small brown horse with a red halter is grazing in a lush green field. The camera captures the scene from a low-angle shot, focusing on the horse's movements as **it lowers its head to eat grass**. The environment is serene and natural, with trees and greenery surrounding the area. The lighting is bright and natural, suggesting a sunny day. The background remains slightly blurred, keeping the attention on the horse. The atmosphere is calm and peaceful, emphasizing the tranquility of the rural setting. There is **no significant camera movement**, maintaining a steady view of the horse throughout the video.

Figure 13: Fine-grained captions for the middle frame and video, sampled from training data. The blue text in the captions highlights the difference between the two types of captions. The **mid-frame caption** focuses more on fine-grained spatial information, while the missing temporal information is provided by the **video caption**.

As mentioned in Section 2.1.2, for each video sample, we provide two captions: one for the middle frame and one for the entire video. The former tends to describe spatial information, while the latter incorporates temporal changes. This helps the model achieve fine-grained language-to-visual mapping across both spatial and temporal dimensions. Meanwhile, we use a video captioner based on Aria, which provides multi-level information, including foreground, background, style, environment, and camera movement. This offers the model potential capabilities to handle multi-granularity control, which is often required in commercial environments at the data annotation level. Figure 13 shows some examples of fine-grained captions for the middle frame and video in the training data.

## B Model Specifications

The model parameter settings of Allegro are illustrated in Table 6.

Param. Name	Param. Value	Param. Name	Param. Value
in_channels	4	caption_channels	4096
out_channels	4	cross_attention_dim	2304
num_layers	32	attention_head_dim	96
interpolation_scale_h	2.0	num_attention_heads	24
interpolation_scale_t	2.2	attention_bias	true
interpolation_scale_w	2.0	dropout	0.0
patch_size	2	activation_fn	gelu-approximate
patch_size_t	1	norm_type	ada_norm_single
sample_size	[90, 160]	norm_num_groups	32
sample_size_t	22	norm_eps	1e-06
use_rope	true	num_embeds_ada_norm	1000

Table 6: The parameter settings of Allegro.

## C More Results on VBench

We show all dimensions of VBench in Figure 14.

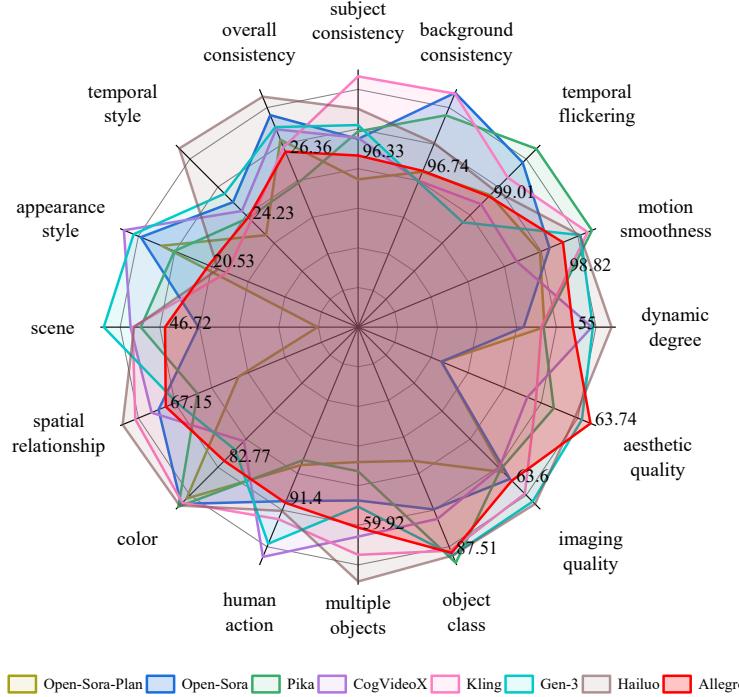


Figure 14: Comparison of Allegro and other video generation models on VBench.

## D More Visual Comparisons

We show more visual results of our method and other methods in Figure 15, Figure 16, Figure 17 and Figure 18.

## E More Visual Results

We show more visual results of our Allegro in Figure 19, Figure 20 and Figure 21.

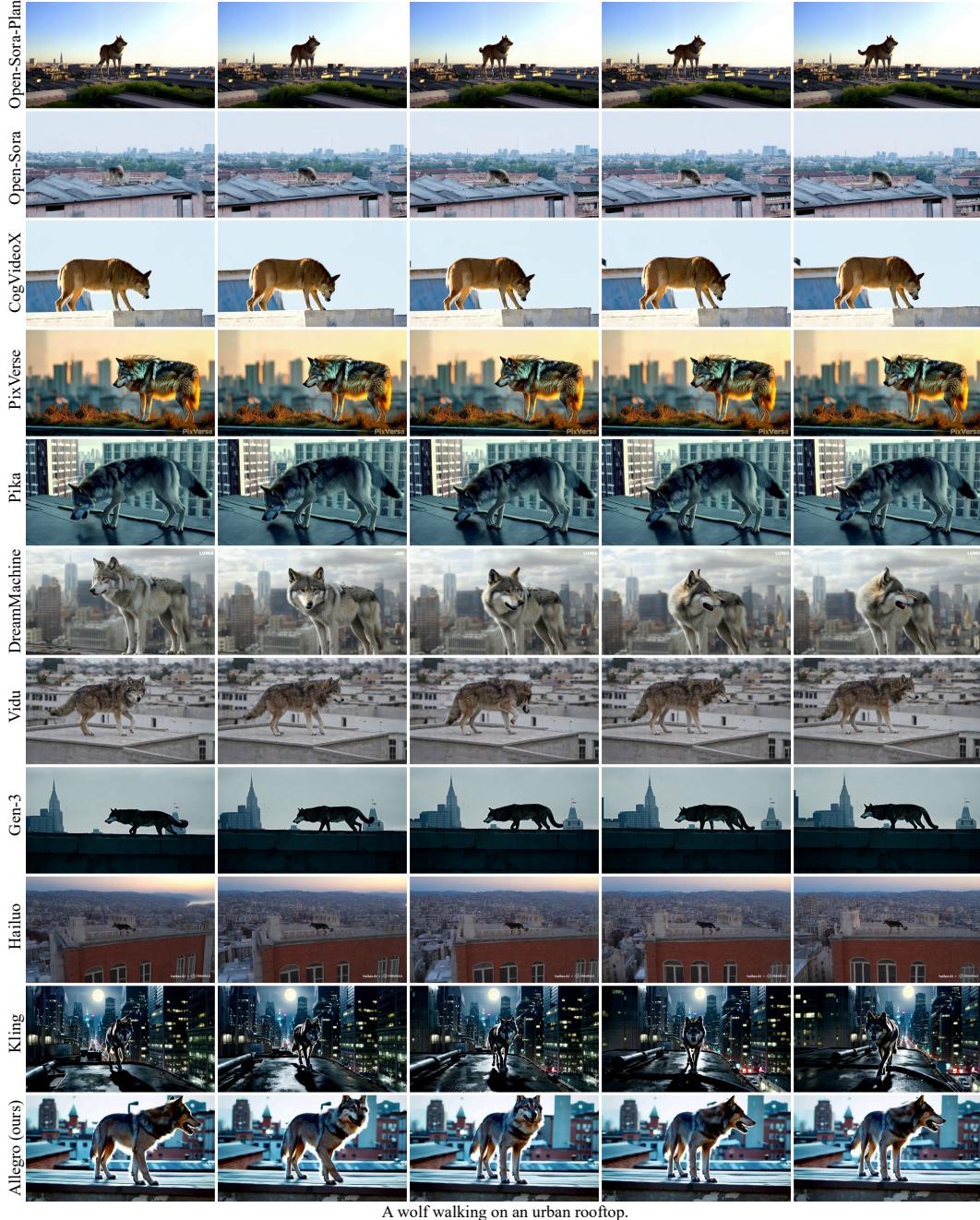
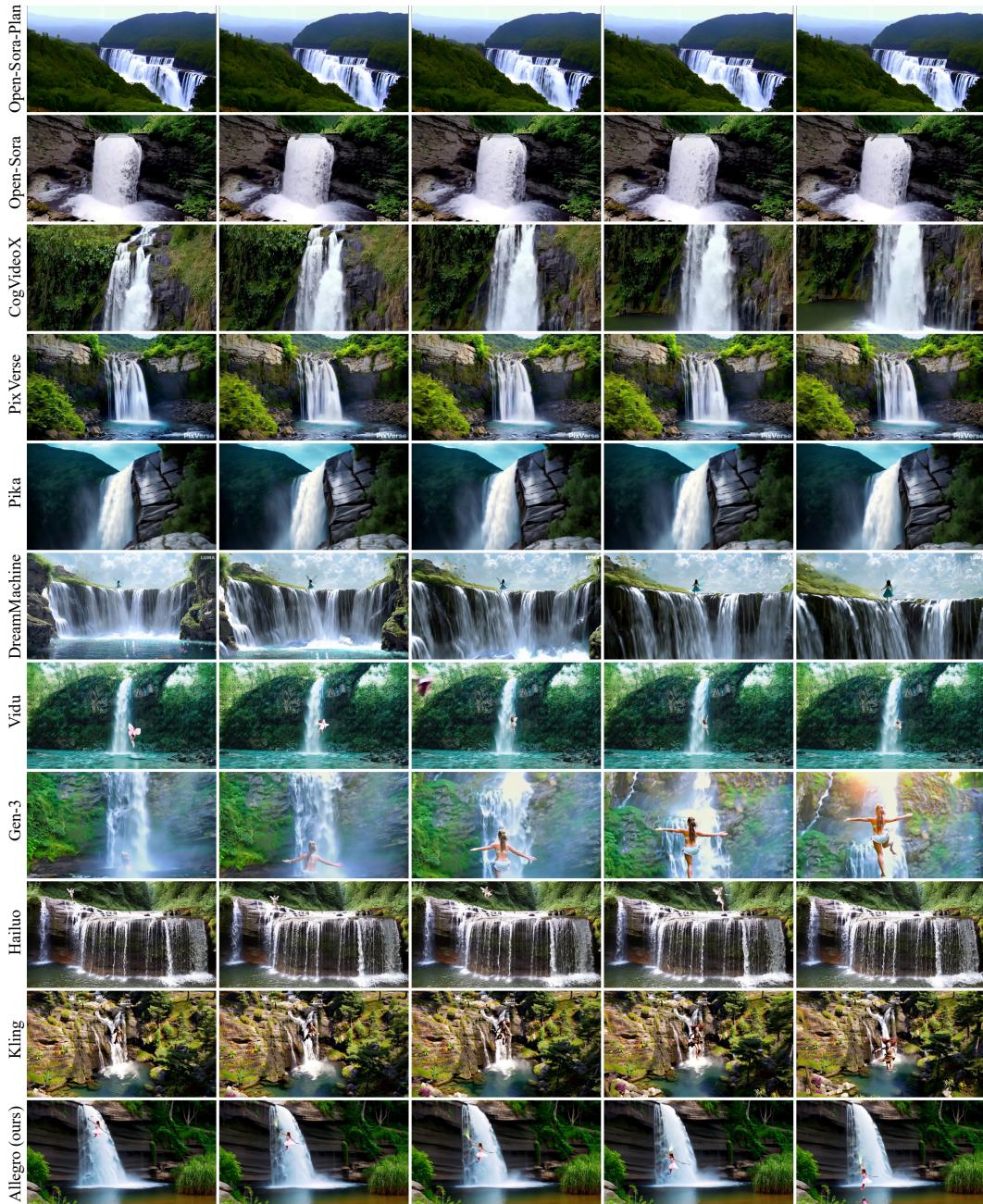


Figure 15: Qualitative comparisons of our Allegro with SOTA methods for the user study.



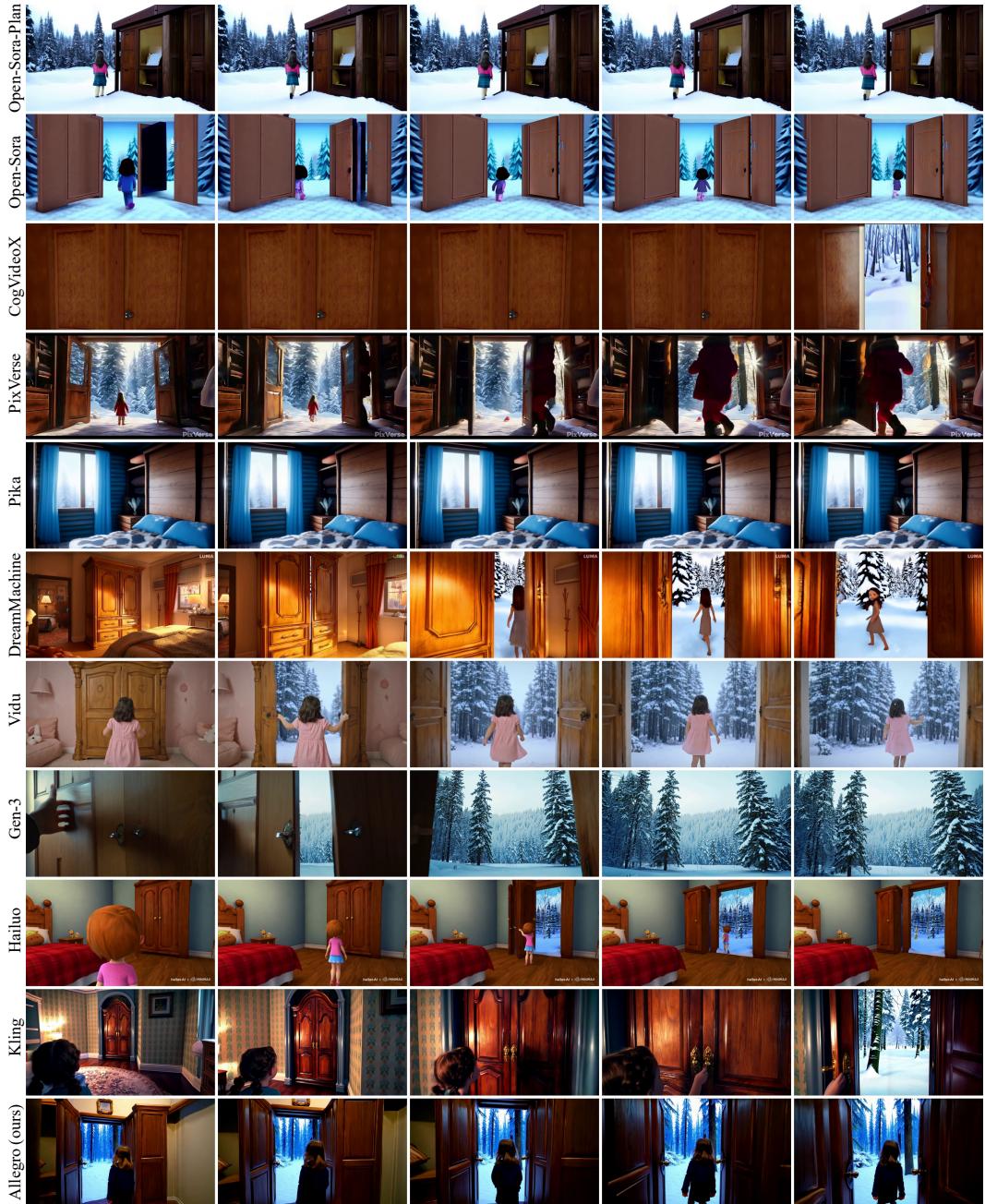
A waterfall cascades down a rocky cliff, forming a small lake below and then a river. The area around the waterfall is surrounded by lush green plants. A girl resembling a fairy leaps from the top of the waterfall.

Figure 16: Qualitative comparisons of our Allegro with SOTA methods for the user study.



A little girl is running by the seaside, holding a seashell in her hand, wanting to show it to her parents who are beside her.

Figure 17: Qualitative comparisons of our Allegro with SOTA methods for the user study.



A little girl walks into a cozy bedroom, where there is a wooden wardrobe with double doors. The girl approaches and opens the wardrobe, which surprisingly serves as a portal to another world—a snowy forest. She steps through the wardrobe and into the icy forest.

Figure 18: Qualitative comparisons of our Allegro with SOTA methods for the user study.



Figure 19: The generated video frames of our Allegro with text inputs of varying lengths.



Close-up shot of several skewers of grilled meat on a barbecue grill, emitting wisps of smoke, looks extremely delicious.



A curious raccoon explores a suburban backyard at night, its nimble paws rummaging through a trash can in search of food.



A woman is jogging through a forest trail with her husky. The dog runs ahead, occasionally stopping to sniff the ground and look back at her.



A pair of deer cautiously step out of a dense forest into a serene meadow. They graze on the lush grass, their ears twitching at every sound from the trees.



A cat prowls through a narrow alleyway in an ancient city. Shadows play on the cobblestone streets as it moves stealthily, pausing to listen to distant sounds.



An isolated iron lighthouse shining light out to sea at night as it sits on a rocky stone island being battered by huge ocean waves, smoky orange clouds filling the sky.



Tomato sauce pouring over fries, depth of field, bokeh, soft light, centered, extremely detailed, Nikon D850,(35mm|50mm|85mm), award winning photography.



In a gloomy afternoon, a little boy dressed in white is wandering in the garden. He makes his way through the grass, a large python lunges at him. The scene is captured from a third-person perspective, trailing the boy from behind.



A sleek black motorcycle roars through a dense forest, the trees forming a green tunnel around the rider. Sunlight filters through the leaves, creating a dappled pattern on the road as the bike navigates sharp turns and narrow paths, with the sound of the engine echoing through the woods. The camera follows closely behind the bike.

Figure 20: The generated video frames of our Allegro with text inputs of varying lengths.



Two young women standing in the street, looking at a map of the city. they are both wearing striped shirts and are standing near a pole. the woman on the left is looking at the map, while the other woman appears to be looking at the camera. the scene captures a moment of exploration and curiosity, as they navigate the city streets together.



A cityscape with a tall building and a parking lot in the foreground. the sky is filled with clouds, creating a dramatic backdrop for the scene. as the sun sets or rises, the city lights come on, illuminating the area. the presence of the parking lot suggests that this could be a busy urban area with various vehicles and infrastructure. the combination of the city lights, the clouds, and the sun creates a visually striking and dynamic atmosphere in the video.



a heartwarming moment between an elderly man and an older boy. the man is embracing the boy's head, and they are both standing on a boat. the scene is set in a picturesque outdoor setting, with the man and the boy sharing a close moment, possibly as a bonding experience or a special occasion. the presence of the boat suggests that they might be on a vacation or a leisurely outing together, enjoying each other's company and creating lasting memories.



A moment of a woman in a white wedding dress, adorned with a pearl necklace and veil, standing in front of a window. the window is draped with white curtains, allowing soft light to filter into the room. the woman's gaze is directed off to the side, and her lips are slightly parted, as if she is in the midst of a conversation or lost in thought. the background is blurred, drawing focus to the woman and the window. the overall scene suggests a quiet, intimate moment, possibly before or after a wedding ceremony.



A young man wearing glasses and a plaid jacket, standing in a forest surrounded by trees. he is holding a cell phone to his ear, possibly talking on it or using it for navigation. the man appears to be engaged in a conversation or activity while being outdoors in the woods. the presence of a pipe in the scene suggests that he might be enjoying a leisurely activity or a break from his phone conversation. the image captures a moment of connection or engagement with the natural environment while still being connected to the modern world through the use of a cell phone.



Create a monochromatic video clip featuring a vintage-themed superhero in a restaurant. the superhero, dressed in a vintage-style classic suit, appears angry. the scene unfolds in a restaurant with sunlight streaming through large windows, providing natural lighting. the superhero, visibly frustrated, performs the highly stylized action of swimming indoors. the camera gradually zooms out, capturing the unique setting and the superhero's intense facial expression, emphasizing the peculiar combination of actions within the well-lit, monochromatic interior.

Figure 21: The generated video frames of our Allegro with text inputs of varying lengths.