
HunyuanVideo: A Systematic Framework For Large Video Generative Models

“Bridging the gap between closed-source and open-source video foundation models to accelerate community exploration.” — Hunyuan Foundation Model Team

Abstract

Recent advancements in video generation have profoundly transformed daily life for individuals and industries alike. However, the leading video generation models remain closed-source, creating a substantial performance disparity in video generation capabilities between the industry and the public community. In this report, we present HunyuanVideo, a novel open-source video foundation model that exhibits performance in video generation that is comparable to, if not superior to, leading closed-source models. HunyuanVideo features a comprehensive framework that integrates several key contributions, including data curation, advanced architecture design, progressive model scaling and training, and an efficient infrastructure designed to facilitate large-scale model training and inference. With those, we successfully trained a video generative model with over 13 billion parameters, making it the largest among all open-source models. We conducted extensive experiments and implemented a series of targeted designs to ensure high visual quality, motion dynamics, text-video alignment, and advanced filming techniques. According to professional human evaluation results, HunyuanVideo outperforms previous state-of-the-art models, including Runway Gen-3, Luma 1.6, and 3 top performing Chinese video generative models. By releasing the code of the foundation model and its applications, we aim to bridge the gap between closed-source and open-source communities. This initiative will empower everyone in the community to experiment with their ideas, fostering a more dynamic and vibrant video generation ecosystem. The code is publicly available at <https://github.com/Tencent/HunyuanVideo>.

Figure 1: Non-curated multi-ratio generation samples with HunyuanVideo, showing realistic, concept generalization and automatic scene-cut features.

1 Introduction

With extensive pre-training and advanced architectures, diffusion models [51, 65, 21, 72, 5, 25, 67, 47] have demonstrated superior performance in generating high-quality images and videos compared to previous generative adversarial network (GAN) methods [6]. However, unlike the image generation field, which has seen a proliferation of novel algorithms and applications across various open platforms, diffusion-based video generative models remain relatively inactive. We contend that one of the primary reasons for this stagnation is the lack of robust open-source foundation models as in T2I filed [47]. In contrast to the image generative model community, a significant gap has emerged between open-source and closed-source video generation models. Closed-source models tend to overshadow publicly available open-source alternatives, severely limiting the potential for algorithmic innovation from the public community. While the recent state-of-the-art model MovieGen [67] has demonstrated promising performance, its milestone for open-source release has yet to be established.

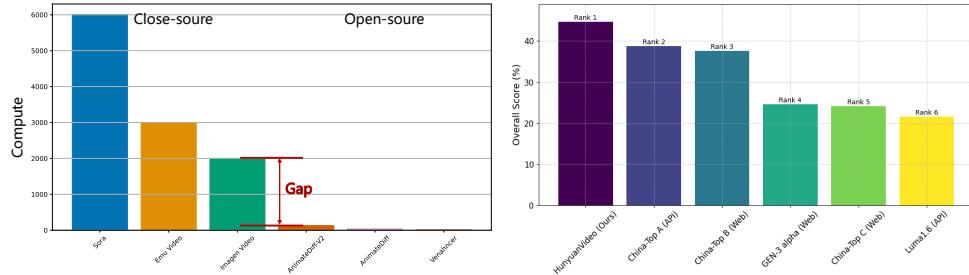


Figure 2: Left: Computation resources used for closed-source and open-source video generation models. Right: Performance comparison between HunyuanVideo and other selected strong baselines.

To address the existing gap and enhance the capabilities of the public community, this report presents our open-sourced foundational video generative model, HunyuanVideo. This systematic framework encompasses training infrastructure, data curation, model architecture optimization, and model training. Through our experiments, we discovered that randomly scaling the training data, computational resources, and model parameters of a simple Transformer-based generative model [65] trained with Flow Matching [52] was not sufficiently efficient. Consequently, we explored an effective scaling strategy that can reduce computational resource requirements by up to 5 \times while achieving the desired model performance. With this optimal scaling approach and dedicated infrastructure, we successfully trained a large video model comprising 13 billion parameters, pre-training it on internet-scale images and videos. After a dedicated progressive fine-tuning strategy, HunyuanVideo excels in four critical aspects of video generation: visual quality, motion dynamics, video-text alignment, and semantic scene cut. We conducted a comprehensive comparison of HunyuanVideo with leading global video generation models, including Gen-3 and Luma 1.6 and 3 top performing commercial models in China, using over 1,500 representative text prompts accessed by a group of 60 people. The results indicate that HunyuanVideo achieves the highest overall satisfaction rates, particularly excelling in motion dynamics.

2 Overview

HunyuanVideo is a comprehensive video training system encompassing all aspects from data processing to model deployment. This technical report is structured as follows:

- In **Section 3**, we introduce our data preprocessing techniques, including filtering and re-captioning models.
- **Section 4** presents detailed information about the architecture of all components of HunyuanVideo, along with our training and inference strategies.
- In **Section 5**, we discuss methods for accelerating model training and inference, enabling the development of a large model with 13 billion parameters.
- **Section 6** evaluates the performance of our text-to-video foundation models and compares them with state-of-the-art video generation models, both open-source and proprietary.

- Finally, in **Section 7**, we showcase various applications built on the pre-trained foundation model, accompanied by relevant visualizations as well as some video related functional models such as video to audio generative model.

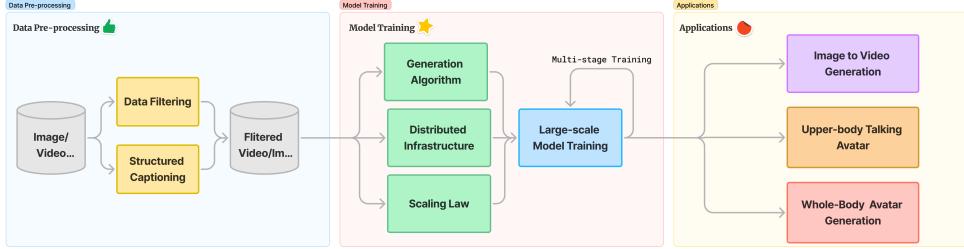


Figure 3: The overall training system for HunyuanVideo.

3 Data Pre-processing

We use an image-video joint training strategy. The videos are meticulously divided into five distinct groups, while images are categorized into two groups, each tailored to fit the specific requirements of their respective training processes. This section will primarily delve into the intricacies of video data curation.

Our data acquisition process is rigorously governed by the principles outlined in the General Data Protection Regulation (GDPR) [39] framework. Furthermore, we employ advanced techniques such as data synthesis and privacy computing to guarantee compliance with these stringent standards.

Our raw data pool initially comprised videos spanning a wide range of domains including people, animals, plants, landscapes, vehicles, objects, buildings, and animation. Each video was acquired with a set of basic thresholds, including minimum duration requirements. Additionally, a subset of the data was collected based on more stringent criteria, such as spatial quality, adherence to a specific aspect ratio, and professional standards in composition, color, and exposure. These rigorous standards ensure that our videos possess technical quality and aesthetic appeal. We experimentally verified that incorporating high-quality data is instrumental in significantly enhancing model performance.

3.1 Data Filtering

Our raw data from different sources exhibits varying durations and levels of quality. To address this, we employ a series of techniques to pre-process the raw data. Firstly, we utilize PySceneDetect [19] to split raw videos into single-shot video clips. Next, we employ the Laplacian operator from OpenCV [18] to identify a clear frame, serving as the starting frame of each video clip. Using an internal VideoCLIP model, we calculate embeddings for these video clips. These embeddings serve two purposes: (i) we deduplicate similar clips based on the Cosine distance of their embeddings; (ii) we apply k-means [59] to obtain $\sim 10K$ concept centroids for concept resampling and balancing.

To continuously enhance video aesthetics, motion, and concept range, we implement a hierarchical data filtering pipeline for constructing training datasets, as shown in Figure 4. This pipeline incorporates various *filters* to help us filter data from different perspectives which we introduce next.

We employ Dover [85] to assess the visual aesthetics of video clips from both aesthetic and technical viewpoints. Additionally, we train a model to determine clarity and eliminate video clips with visual blurs. By predicting the motion speed of videos using estimated optical flow [18], we filter out static or slow-motion videos. We combine the results from PySceneDetect [19] and Transnet v2 [76] to get scene boundary information. We utilize an internal OCR model to remove video clips with excessive text, as well as to locate and crop subtitles. We also develop YOLOX [24]-like visual models to detect and remove some occluded or sensitive information such as watermarks, borders, and logos. To assess the effectiveness of these filters, we perform simple experiments using a smaller HunyuanVideo

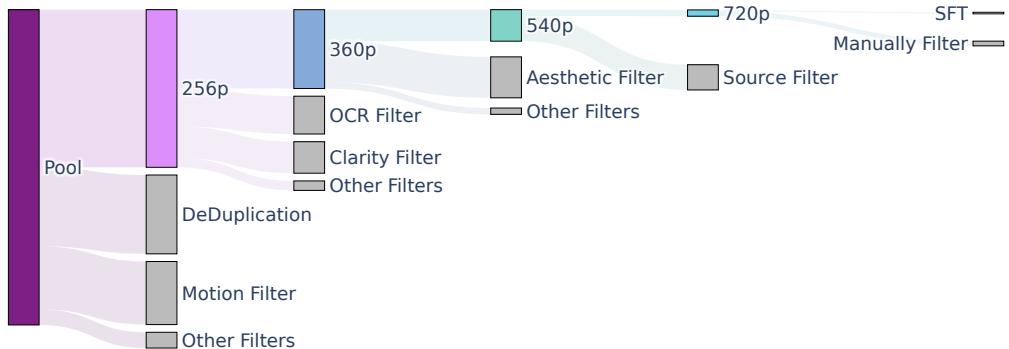


Figure 4: **Our hierarchical data filtering pipeline.** We employ various filters for data filtering and progressively increase their thresholds to build 4 training datasets, i.e., 256p, 360p, 540p, and 720p, while the final SFT dataset is built through manual annotation. This figure highlights some of the most important filters to use at each stage. A large portion of data will be removed at each stage, ranging from half to one-fifth of the data from the previous stage. Here, gray bars represent the amount of data filtered out by each filter while colored bars indicate the amount of remaining data at each stage.

model and observe the performance changes. The results obtained from these experiments play an important role in guiding the building of our data filtering pipeline, which is introduced next.

Our hierarchical data filtering pipeline for video data yields five training datasets, corresponding to the five training stages (Section 4.5). These datasets (except for the last fine-tuning dataset) are curated by progressively improving the thresholds of the aforementioned filters. The video spatial resolution increases progressively from $256 \times 256 \times 65$ to $720 \times 1280 \times 129$. We apply varying levels of strictness to the filters during the threshold adjustment process at different stages (see Figure 4). The last dataset used for fine-tuning is described next.

To improve the model’s performance in the final stage (Section 4.7), we build a fine-tuning dataset comprising $\sim 1M$ samples. This dataset is meticulously curated through human annotation. Annotators are assigned the task of identifying video clips that exhibit high visual aesthetics and compelling content motion. Each video clip undergoes evaluation based on two perspectives: (i) decomposed aesthetical views, including color harmony, lighting, object emphasis, and spatial layout; (ii) decomposed motion views, encompassing motion speed, action integrity, and motion blurs. Finally, our fine-tuning dataset consists of visually appealing video clips with intricate motion details.

We also establish a hierarchical data filtering pipeline for images by reusing most of the filters, excluding the motion-related ones. Similarly, we build two image training datasets by progressively increasing the filtering thresholds applied to an image pool of billions of image-text pairs. The first dataset contains billions of samples and is used for the initial stage of text-to-image pre-training. The second dataset contains hundreds of millions of samples and is utilized for the second stage of text-to-image pre-training.

3.2 Data Annotation

Structured Captioning. As evidenced in research [7, 4], the precision and comprehensiveness of captions play a crucial role in improving the prompt following ability and output quality of generative models. Most previous work focus on providing either brief captions [14, 50] or dense captions [93, 9, 10]. However, these approaches are not without shortcomings, suffering from incomplete information, redundant discourse and inaccuracies. In pursuit of achieving captions with higher comprehensiveness, information density and accuracy, we develop and implement an in-house Vision Language Model(VLM) designed to generate structured captions for images and videos. These structured captions, formatted in JSON, provide multi-dimensional descriptive information from various perspectives, including: 1) **Short Description:** Capturing the main content of the scene.

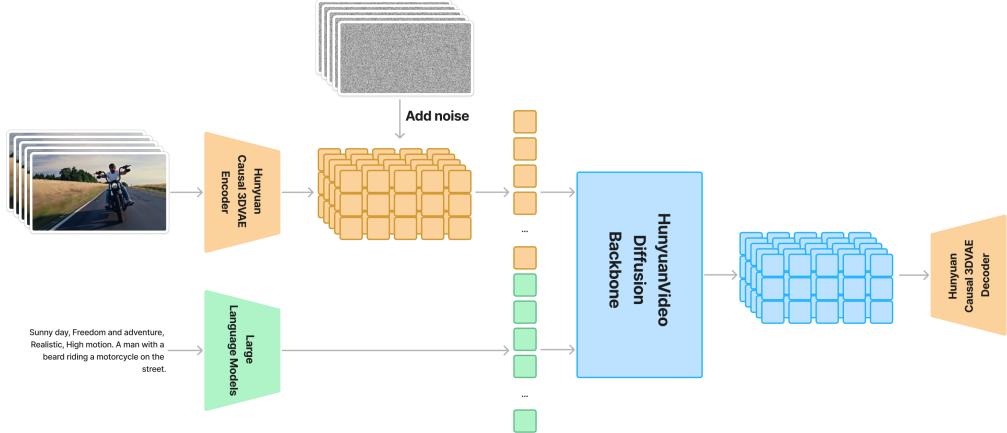


Figure 5: The overall architecture of HunyuanVideo. The model is trained on a spatial-temporally compressed latent space, which is compressed through Causal 3D VAE. Text prompts are encoded using a large language model, and used as the condition. Gaussian noise and condition are taken as input, our model generates a output latent, which is decoded into images or videos through the 3D VAE decoder.

2) Dense Description: Detailing the scene’s content, which notably includes scene transitions and camera movements that are integrated with the visual content, such as camera follows some subject. **3) Background:** Describing the environment in which the subject is situated. **4) Style:** Characterizing the style of the video, such as documentary, cinematic, realistic, or sci-fi. **5) Shot Type:** Identifying the type of video shot that highlights or emphasizes specific visual content, such as aerial shot, close-up shot, medium shot, or long shot. **6) Lighting:** Describing the lighting conditions of the video. **7) Atmosphere:** Conveying the atmosphere of the video, such as cozy, tense, or mysterious.

Moreover, we extend the JSON structure to incorporate additional metadata-derived elements, including source tags, quality tags, and other pertinent tags from meta information of images and videos. Through the implementation of a carefully designed dropout mechanism coupled with permutation and combination strategies, we synthesize captions diverse in length and pattern by assembling these multi-dimensional descriptions for each image and video, aiming to improve the generalization ability of generative models and prevent overfitting. We utilize this captioner to provide structured captions for all images and videos in our training dataset.

Camera Movement Types. We also train a camera movement classifier capable of predicting 14 distinct camera movement types, including zoom in, zoom out, pan up, pan down, pan left, pan right, tilt up, tilt down, tilt left, tilt right, around left, around right, static shot and handheld shot. High-confidence predictions of camera movement types are integrated into the JSON-formatted structured captions, to enable camera movement control ability of generative models.

4 Model Architecture Design

The overview of our HunyuanVideo model is shown in Fig. 5. This section describes the Causal 3D VAE, diffusion backbone, and scaling laws experiments.

4.1 3D Variational Auto-encoder Design

Similar to previous work [67, 93], we train a 3DVAE to compress pixel-space videos and images into a compact latent space. To handle both videos and images, we adopt CausalConv3D [95]. For a video of shape $(T + 1) \times 3 \times H \times W$, our 3DVAE compresses it into latent features with shape $(\frac{T}{c_t} + 1) \times C \times (\frac{H}{c_s}) \times (\frac{W}{c_s})$. In our implementation, $c_t = 4$, $c_s = 8$, and $C = 16$. This compression significantly reduces the number of tokens for the subsequent diffusion transformer model, allowing us to train videos at the original resolution and frame rate. The model structure is illustrated in Figure 6.

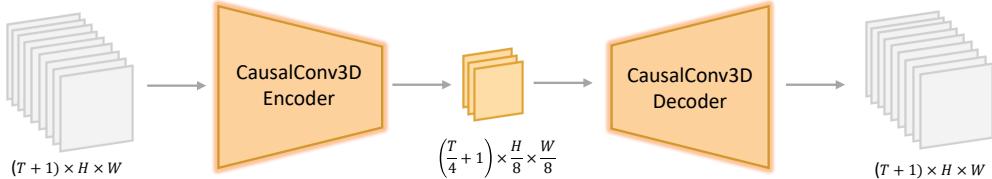


Figure 6: The architecture of our 3DVAE.

4.1.1 Training

In contrast to most previous work [67, 11, 104], we do not rely on a pre-trained image VAE for parameter initialization; instead, we train our model from scratch. To balance the reconstruction quality of videos and images, we mix video and image data at a ratio of 4 : 1. Besides the routinely used L_1 reconstruction loss and KL loss L_{kl} , we also incorporate perceptual loss L_{lpips} and GAN adversarial loss L_{adv} [22] to enhance the reconstruction quality. The complete loss function is shown in Equation 1.

$$\text{Loss} = L_1 + 0.1L_{lpips} + 0.05L_{adv} + 10^{-6}L_{kl} \quad (1)$$

During training, we employ a curriculum learning strategy, gradually training from low-resolution short video to high-resolution long video. To improve the reconstruction of high-motion videos, we randomly choose a sampling interval from the range $1 \sim 8$ to sample frames evenly across video clips.

4.1.2 Inference

Encoding and decoding high-resolution long videos on a single GPU can lead to out-of-memory (OOM) errors. To address this, we use a spatial-temporal tiling strategy, splitting the input video into overlapping tiles along the spatial and temporal dimensions. Each tile is encoded/decoded separately, and the outputs are stitched together. For the overlapping regions, we utilize a linear combination for blending. This tiling strategy allows us to encode/decode videos in arbitrary resolutions and durations on a single GPU.

We observed that directly using the tiling strategy during inference can result in visible artifacts due to inconsistencies between training and inference. To solve this, we introduce an additional finetuning phase where the tiling strategy is randomly enabled/disabled during training. This ensures the model is compatible with both tiling and non-tiling strategies, maintaining consistency between training and inference.

Table 1 compares our VAE with open-source state-of-the-art VAEs. On video data, our VAE demonstrates a significantly higher PSNR compared to other video VAEs. On images, our performance surpasses both video VAEs and image VAE. Figure 7 shows several cases at 256×256 resolution. Our VAE demonstrates significant advantages in text, small faces, and complex textures.

Table 1: VAE reconstruction metrics comparison.

Model	Downsample Factor	$ z $	ImageNet (256×256) PSNR↑	MCL-JCV ($33 \times 360 \times 640$) PSNR↑
FLUX-VAE [47]	$1 \times 8 \times 8$	16	32.70	-
OpenSora-1.2 [102]	$4 \times 8 \times 8$	4	28.11	30.15
CogvideoX-1.5 [93]	$4 \times 8 \times 8$	16	31.73	33.22
Cosmos-VAE [64]	$4 \times 8 \times 8$	16	30.07	32.76
Ours	$4 \times 8 \times 8$	16	33.14	35.39



Figure 7: VAE reconstruction case comparison.

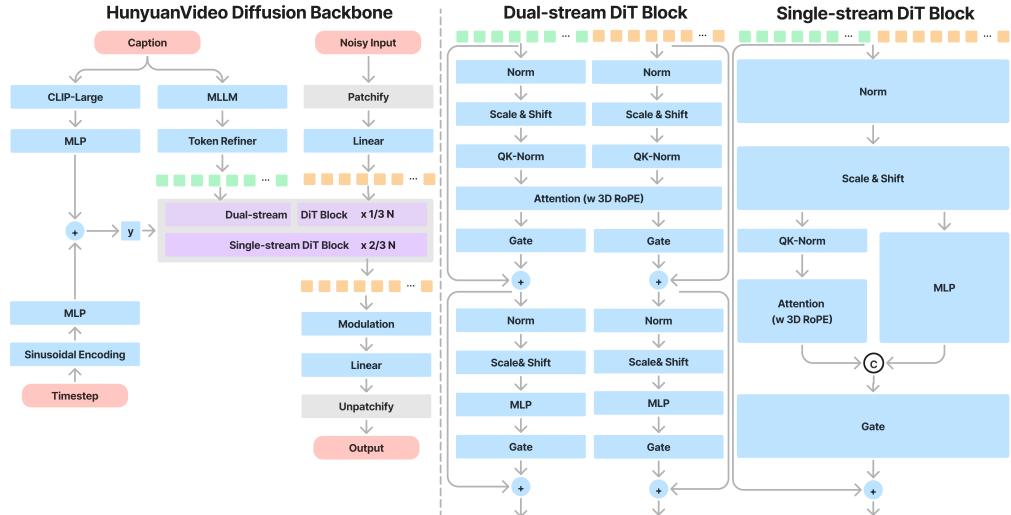


Figure 8: The architecture of our HunyuanVideo Diffusion Backbone.

4.2 Unified Image and Video Generative Architecture

In this section, we introduce the Transformer design in HunyuanVideo, which employs a unified Full Attention mechanism for three main reasons: Firstly, it has demonstrated superior performance compared to divided spatiotemporal attention [7, 67, 93, 79]. Secondly, it supports unified generation for both images and videos, simplifying the training process and improving model scalability. Lastly, it leverages existing LLM-related acceleration capabilities more effectively, enhancing both training and inference efficiency. The model structure is illustrated in Figure 8.

Inputs. For a given video-text pair, the model operates within the 3D latent space described in Section 4.1. Specifically, for the video branch, the input is first compressed into latents of shape $T \times C \times H \times W$. To unify input processing, we treat images as single-frame videos. These latents are then patchified and unfolded into a 1D sequence of tokens with a length of $\frac{T}{k_t} \cdot \frac{H}{k_h} \cdot \frac{W}{k_w}$ using a 3D convolution with a kernel size of $k_t \times k_h \times k_w$. For the text branch, we first use an advanced LLM to encode the text into a sequence of embeddings that capture fine-grained semantic information. Concurrently, we employ the CLIP model to extract a pooled text representation containing global

Table 2: Architecture hyperparameters for the HunyuanVideo 13B parameter foundation model.

Dual-stream Blocks	Single-stream Blocks	Model Dimension	FFN Dimension	Attention Heads	Head dim	(d_t, d_h, d_w)
20	40	3072	12288	24	128	$(16, 56, 56)$

information. This representation is then expanded in dimensionality and added to the timestep embedding before being fed into the model.

Model Design. To integrate textual and visual information effectively, we follow a similar strategy of "Dual-stream to Single-stream" hybrid model design as introduced in [47] for video generation. In the dual-stream phase, video and text tokens are processed independently through multiple Transformer blocks, enabling each modality to learn its own appropriate modulation mechanisms without interference. In the single-stream phase, we concatenate the video and text tokens and feed them into subsequent Transformer blocks for effective multimodal information fusion. This design captures complex interactions between visual and semantic information, enhancing overall model performance.

Position Embedding. To support multi-resolution, multi-aspect ratio, and varying duration generation, we use Rotary Position Embedding (RoPE) [77] in each Transformer block. RoPE applies a rotary frequency matrix to the embeddings, enhancing the model’s ability to capture both absolute and relative positional relationships, and demonstrating some extrapolation capability in LLMs. Given the added complexity of the temporal dimension in video data, we extend RoPE to three dimensions. Specifically, we compute the rotary frequency matrix separately for the coordinates of time (T), height (H), and width (W). We then partition the feature channels of the query and key into three segments (d_t, d_h, d_w) , multiply each segment by the corresponding coordinate frequencies and concatenate the segments. This process yields position-aware query and key embeddings, which are used for attention computation.

For detailed model settings, please refer to Table 2.

4.3 Text encoder

In generation tasks like text-to-image and text-to-video, the text encoder plays a crucial role by providing guidance information in the latent space. Some representative works [66, 21, 51] typically use pre-trained CLIP [69] and T5-XXL [71] as text encoders where CLIP uses Transformer Encoder and T5 uses an Encoder-Decoder structure. In contrast, we utilize a pre-trained Multimodal Large Language Model (MLLM) with a Decoder-Only structure as our text encoder, which has following advantages: (i) Compared with T5, MLLM after visual instruction finetuning has better image-text alignment in the feature space, which alleviates the difficulty of instruction following in diffusion models; (ii) Compared with CLIP, MLLM has been demonstrated superior ability in image detail description and complex reasoning [53]; (iii) MLLM can play as a zero-shot learner [8] by following system instructions prepended to user prompts, helping text features pay more attention to key information. In addition, as shown in Fig. 9, MLLM is based on causal attention while T5-XXL utilizes bidirectional attention that produces better text guidance for diffusion models. Therefore, we follow [55] to introduce an extra bidirectional token refiner for enhancing text features. We have configured HunyuanVideo with a series of MLLMs [78, 17, 26] for different purposes. Under each setting, MLLMs have shown superior performance over conventional text encoder.

In addition, CLIP text features are also valuable as the summary of the text information. As shown in Fig. 8 We adopt the final non-padded token of CLIP-Large text features as a global guidance, integrating into the dual-stream and single-stream DiT blocks.

4.4 Model Scaling

Neural scaling laws [41, 36] in language model training offer a powerful tool for understanding and optimizing the performance of machine learning models. By elucidating the relationships between model size (N), dataset size (D), and computational resources (C), these laws help drive the development of more effective and efficient models, ultimately advancing the success of large model training.

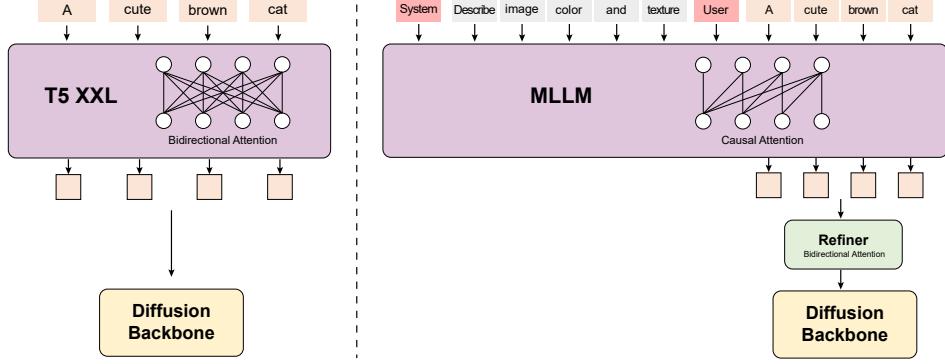


Figure 9: Text encoder comparison between T5 XXL and the instruction-guided MLLM introduced by HunyuanVideo.

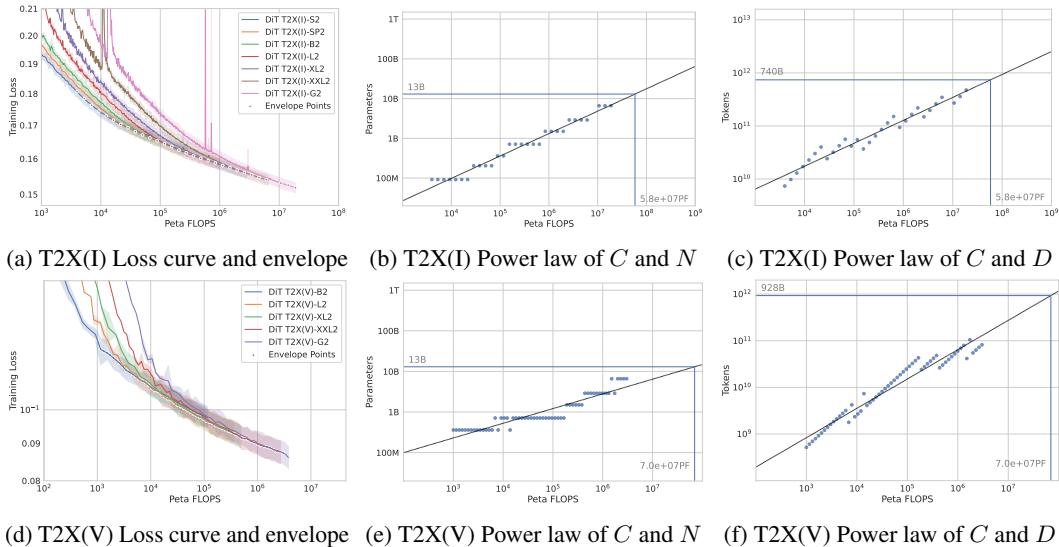


Figure 10: Scaling laws of DiT-T2X model family. On the top-left (a) we show the loss curves of the T2X(I) model on a log-log scale for a range of model sizes from 92M to 6.6B. We follow [36] to plot the envelope in gray points, which are used to estimate the power-law coefficients of the amount of computation (C) vs model parameters (N) (b) and the computation vs tokens (D) (c). Based on the scaling law of the T2X(I) model, we plot the scaling law of the corresponding T2X(V) model in (d), (e), and (f).

In contrast to prior scaling laws on large language models [41, 36, 81, 1, 2] and image generation models [49, 42], video generation models typically rely on pre-trained image models. Consequently, our initial step involved establishing the foundational scaling laws pertinent to text-to-image. Building upon these foundational scaling laws, we subsequently derived the scaling laws applicable to the text-to-video model. By integrating these two sets of scaling laws, we were able to systematically determine the appropriate model and data configuration for video generation tasks.

4.4.1 Image model scaling law

Kaplan et.al [41] and Hoffmann et.al [36] explored emperical scaling laws for language models on cross-entropy loss. In the field of diffusion based visual generation, Li et.al [49] study the scaling properties on UNet, while transformer based works such as DiT [65], U-ViT [3], Lumina-T2X [23], and SD3 [21] only study the scaling behavior between sample quality and network complexity, leaving the power-laws about the computation resources and MSE loss used by diffusion models unexplored.

In order to fill the gap, we develop a family of DiT-like models, named as DiT-T2X to distinguish from the original DiT, where X can be the image (I) or the video (V). DiT-T2X applies T5-XXL [71] as the text encoder and the aforementioned 3D VAE as the image encoder. The text information is injected to the model according to cross-attention layers. The DiT-T2X family has seven sizes ranging from 92M to 6.6B. The models were trained using DDPM [34] and v-prediction [73] with consistent hyperparameters and the same dataset with 256px resolution. We follow the experiment method introduced by [36] and build the neural scaling laws to fit

$$N_{opt} = a_1 C^{b_1}, \quad D_{opt} = a_2 C^{b_2}. \quad (2)$$

As shown in Fig. 10 (a), the loss curve of each model decreases from top left to bottom right, and it always passes through the loss curve of the larger size model adjacent to it. It means that each curve will form two intersections with curves of the larger and the smaller models. Under the corresponding computation resources between the two intersections, the middle-sized model is optimal (with the lowest loss). After obtaining the envelope of lowest losses across all the x-axis values, we fill the Equation (2) to find out that $a_1 = 5.48 \times 10^{-4}$, $b_1 = 0.5634$, $a_2 = 0.324$ and $b_2 = 0.4325$, where the units of $a_1, a_2, N_{opt}, D_{opt}$ are billions while C has a unit of Peta FLOPs. The Fig. 10 (b) and Fig. 10 (c) show that DiT-T2X(I) family fits the power law quite well. Finally, given computation budgets, we can calculate the optimal model size and dataset size.

4.4.2 Video model scaling law

Based on the scaling law of the T2X(I) model, we select the optimal image checkpoint (*i.e.*, the model on the envelope) corresponding to each size model to serve as the initialization model for the video scaling law experiment. Fig. 10 (d), Fig. 10 (e), and Fig. 10 (f) illustrate the scaling law results of the T2X(V) model, where $a_1 = 0.0189$, $b_1 = 0.3618$, $a_2 = 0.0108$ and $b_2 = 0.6289$. Based on the results of Fig. 10 (b) and Fig. 10 (e), and taking into account the training consumption and inference cost, we finally set the model size to 13B. Then the number of tokens for image and video training can be calculated as shown in Fig. 10 (c) and Fig. 10 (f). It is worth noting that the amount of training tokens calculated by image and video scaling laws is only related to the first stage of training for images and videos respectively. The scaling property of progressive training from low-resolution to high-resolution will be left explored in future work.

4.5 Model-pretraining

We use Flow Matching [52] for model training and split the training process into multiple stages. We first pretrain our model on 256px and 512px images, then conduct joint training on images and videos from 256px to 960px.

4.5.1 Training Objective

In this work, we employ the Flow Matching framework [52, 21, 13] to train our image and video generation model. Flow Matching transforms a complex probability distribution into a simple probability distribution through a series of variable transformations of the probability density function, and generates new data samples through inverse transformations.

During the training process, given an image or video latent representation \mathbf{x}_1 in the training set. We first sample $t \in [0, 1]$ from a logit-normal distribution [21] and initialize a noise $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ following Gaussian distribution. The training sample \mathbf{x}_t is then constructed using a linear interpolation method [52]. The model is trained to predict the velocity $\mathbf{u}_t = d\mathbf{x}_t/dt$, which guides the sample \mathbf{x}_t towards the sample \mathbf{x}_1 . The model parameters are optimized by minimizing the mean squared error between the predicted velocity \mathbf{v}_t and the ground truth velocity \mathbf{u}_t , expressed as the loss function

$$\mathcal{L}_{\text{generation}} = \mathbb{E}_{t, \mathbf{x}_0, \mathbf{x}_1} \|\mathbf{v}_t - \mathbf{u}_t\|^2. \quad (3)$$

During the inference process, a noise sample $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is drawn initially. The first-order Euler ordinary differential equation (ODE) solver is then used to compute \mathbf{x}_1 by integrating the model's estimated values for $d\mathbf{x}_t/dt$. This process ultimately generates the final sample \mathbf{x}_1 .

4.5.2 Image Pre-training

At our early experiments, we found that a well pretrained model significantly accelerates the convergence of video training and improves video generation performance. Therefore, we introduce a two-stage progressive image pretraining strategy to serve as a warmup for video training.

Image stage 1 (256px training). The model is first pretrained with low-resolution 256px images. Specifically, we follow previous work [66] to enable multi-aspect training based on 256px, which helps the model learn to generate images with a wide range of aspect ratios while avoiding the text-image misalignments caused by the crop operation in image preprocessing. Meanwhile, pretraining with low resolution samples allows the model to learn more low-frequency concepts from a larger amount of samples.

Image stage 2 (mix-scale training). We introduce a second image-pretraining stage to further facilitate the model ability on higher resolutions, such as 512px. A trivial solution is to directly finetuning on images based on 512px. However, we found that the model performance finetuned on 512px images will degrade severely on 256px image generation, which may affect the following video pretraining on 256px videos. Therefore, we propose *mix-scale training*, where two or more scales of multi-aspect buckets are included for each training global batch. Each scale have an anchor size, and then the multi-aspect buckets are built based on the anchor size. We train the model on a two-scale dataset with anchor sizes 256px and 512px for learning higher resolution images while maintaining the ability on low resolutions. We also introduce dynamic batch sizes for micro batches with different image scales, maximaizing the GPU memory and computation utilization.

4.5.3 Video-Image joint training

Multiple aspect ratios and durations bucketization. After the data filtering process described in Section 3.1, the videos have different aspect ratios and durations. To effectively utilize the data, we categorize the training data into buckets based on duration and aspect ratio. We create B_T duration buckets and B_{AR} aspect ratio buckets, resulting in a total of $B_T \times B_{AR}$ buckets. As the number of tokens varies across buckets, we assign each bucket a maximum batch size that can prevent out-of-memory (OOM) errors, to optimize GPU resource utilization. Before training, all data is allocated to the nearest bucket. During training, each rank randomly pre-fetches batch data from a bucket. This random selection ensures the model is trained on varying data sizes at each step, which helps maintain model generalization by avoiding the limitations of training on a single size.

Progressive Video-Image Joint Training. Generating high-quality, long-duration video sequences directly from text often leads to difficulties in model convergence and suboptimal results. Therefore, progressive curriculum learning has become a widely adopted strategy for training text-to-video models. In HunyuanVideo, we designed a comprehensive curriculum learning strategy, starting with model initialization using T2I parameters and progressively increasing video duration and resolution.

- **Low-resolution, short video stage.** The model establishes the basic mapping between text and visual content, ensuring consistency and coherence in short-term actions.
- **Low-resolution, long video stage.** The model learns more complex temporal dynamics and scene changes, ensuring temporal and spatial consistency over a longer duration.
- **High-resolution, long video stage.** The model enhances video resolution and detail quality while maintaining temporal coherence and managing complex temporal dynamics.

Additionally, at each stage, we incorporate images in varying proportions for video-image joint training. This approach addresses the scarcity of high-quality video data, enabling the model to learn more extensive and diverse world knowledge. It also effectively prevents catastrophic forgetting of image-space semantics due to distributional discrepancies between video and image data.

4.6 Prompt Rewrite

To address the variability in linguistic style and length of user-provided prompts, we employ the Hunyuan-Large model [78] as our prompt rewrite model to adapt the original user prompt to the model-preferred prompt. Utilized within a training-free framework, the prompt rewrite model capitalizes on detailed prompt instructions and in-context learning examples to enhance its performance. The key functionalities of this prompt rewrite module are as follows:

- **Multilingual Input Adaptation:** The module is designed to process and comprehend user prompts across various languages, ensuring that meaning and context are preserved.
- **Standardization of Prompt Structure:** The module rephrases prompts to conform to a standardized information architecture, akin to training captions.
- **Simplification of Complex Terminology:** The module simplifies complex user wording into more straightforward expressions, all while maintaining the user’s original intent.

Furthermore, we implement a self-revision technique [43] to refine the final prompt. This involves a comparative analysis between the original prompt and the rewritten version, ensuring that the output is both accurate and aligned with the model’s capabilities.

To accelerate and simplify the application process, we also fine-tune a Hunyuan-Large model with LoRA for prompt rewriting. The training data for this LoRA tuning was sourced from the high-quality rewrite pairs collected through the training-free method.

4.7 High-performance Model Fine-tuning

In the pre-training stage, we utilized a large dataset for model training. While this dataset is rich in information, it displayed considerable variability in data quality. To create a robust generation model capable of producing high-quality, dynamic videos and improving its proficiency in continuous motion control and character animation, we carefully selected four specific subsets from the full dataset for fine-tuning. These subsets underwent an initial screening using automated data filtering techniques, followed by manual review. Additionally, we implemented various model optimization strategies to maximize generation performance.

5 Model Acceleration

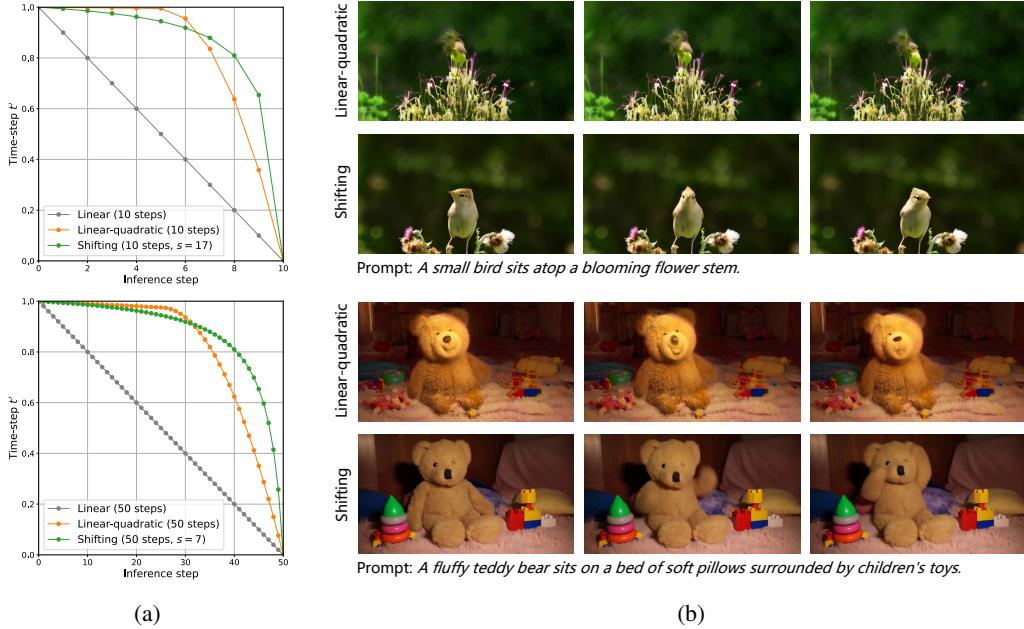


Figure 11: (a) Different time-step schedulers. For our shifting strategy, we set a larger shifting factor s for a lower inference step. (b) Generated videos with only 10 inference steps. The shifting strategy leads to significantly better visual quality.

5.1 Inference Step Reduction

To improve the inference efficiency, we firstly consider reducing the number of inference steps. Compared to image generation, it is more challenging to maintain the spatial and temporal quality

of the generated videos with lower inference steps. Inspired by a previous observation that the first time-steps contribute to most changes during the generation process [101, 67, 98, 99], we utilize the time-step shifting to handle the case of lower inference steps. Specifically, given the inference step $q \in \{1, 2, \dots, Q\}$, $t = 1 - \frac{q}{Q}$ is the input time condition for the generation model, where the noise is initialized at $t = 1$ and the generation process halts at $t = 0$. Instead of using t directly, we map t to t' with a shifting function $t' = \frac{s*t}{1+(s-1)*t}$, where t' is the input time condition and s is the shifting factor. If $s > 1$, the flow model is conditioned more on early time steps. A critical observation is that a lower inference step requires a larger shifting factor s . Empirically, s is set as 7 for 50 inference steps, while s should be increased to 17 when the number of inference steps is smaller than 20. The time-step shifting strategy enables the generation model to match the results of numerous inference steps with a reduced number of steps.

MovieGen [67] applies the linear-quadratic scheduler to achieve a similar goal. The schedulers are visualized in Figure 11a. However, we find that our time-step shifting is more effective than the linear-quadratic scheduler in the case of extremely low number of inference steps, e.g., 10 steps. As shown in Figure 11b, the linear-quadratic scheduler results in worse visual quality.

5.2 Text-guidance Distillation

Classifier-free guidance (CFG) [35] significantly improves the sample quality and motion stability of text-conditioned diffusion models. However, it increases computational cost and inference latency since it requires extra output for the unconditional input at each inference step. Especially for the large video model and high-resolution video generation, the inference burden is extremely expensive when generating text-conditional and text-unconditional videos, simultaneously. To solve this limitation, we distill the combined output for unconditional and conditional inputs into a single student model [60]. Specifically, the student model is conditioned on a guidance scale and shares the same structures and hyper-parameters as the teacher model. We initialize the student model with the same parameters as the teacher model and train with the guidance scale randomly sampled from 1 to 8. We experimentally find that text-guidance distillation approximatively brings 1.9x acceleration.

5.3 Efficient and Scalable Training

To achieve scalability and efficient training, we train HunyuanVideo on AngelPTM [62], the large-scale pretraining framework from Tencent Angel machine learning team. In this part, we first outline the hardware and infrastructure used for training, and then give a detailed introduction to the model parallel method and its optimization methods, followed by the automatic fault tolerance mechanism.

5.3.1 Hardware Infrastructure

To ensure efficient communication in large-scale distributed training, we setup a dedicated distributed training framework termed Tencent XingMai network [48] for highly efficient inter-server communication. The GPU scheduling for all training tasks is completed through the Tencent Angel machine learning platform, which provides powerful resource management and scheduling capabilities.

5.3.2 Parallel Strategy

HunyuanVideo training adopts 5D parallel strategies, including tensor parallelism (TP) [74], sequence parallelism (SP) [45], context parallelism (CP) [63], and data parallelism combined with Zero optimization (DP + ZeroCache [62]). The tensor parallelism (TP) is based on the principle of block calculation of matrices. The model parameters (tensors) are divided into different GPUs to reduce GPU memory usage and accelerate the calculation. Each GPU is responsible for the calculation of different parts of tensors in the layer.

Sequence parallelism (SP) is based on TP. The input sequence dimension is sliced to reduce the repeated calculation of operators such as LayerNorm and Dropout, and reduce the storage of the same activations, which effectively reduces the waste of computing resources and GPU memory. In addition, for input data that does not meet the SP requirements, the engineering equivalent SP Padding capability is supported.

Context parallelism (CP) is sliced in the sequence dimension to support long-sequence training. Each GPU is responsible for calculating the Attention of different sequence slices. Specifically, Ring

Attention [30] is used to achieve efficient training of long sequences through multiple GPUs, breaking through the GPU memory limitation of a single GPU.

In addition, data parallelism + ZeroCache is leveraged to support horizontal expansion through data parallelism to meet the demand for increasing training data sets. Then, based on data parallelism, the ZeroCache optimization strategy is used to further reduce the redundancy of model states (model parameters, gradients, and optimizer states), and we unify the use of GPU memory to maximize the GPU memory usage efficiency.

5.3.3 Optimization

Attention optimization. As the sequence length increases, the attention calculation becomes the main bottleneck of training. We accelerated the attention calculation with FusedAttention.

Recomputation and activations offload optimization. Recomputation is a technology that trade calculations for storage. It is mainly made up of three parts: a) specifying certain layers or blocks for recalculation, b) releasing the activations in the forward calculation, and c) obtaining the dependent activations through recalculation in the backward calculation, which significantly reduces the use of GPU memory during training. In addition, considering the PCIe bandwidth and the host memory size, a layer-based activation offload strategy is adopted. Without reducing the training performance, the activations in the GPU memory are offloaded to the host memory, further saving GPU memory.

5.3.4 Automatic fault tolerance

In terms of the large-scale training stability of HunyuanVideo, an automatic fault tolerance mechanism is used to quickly restore training for common hardware failures. This avoids frequent occurrence of the manual recovery of training tasks. By automatically detecting errors and quickly replacing healthy nodes to pull up training tasks, the training stability is 99.5%.

6 Fundation Model Performance

Text Alignment One of the key metrics for video generative models is their ability to follow text prompts accurately. This capability is essential for the effectiveness of these models. However, some open-source models often struggle to capture all subjects or accurately represent the relationships between multiple subjects, particularly when the input text prompt is complex. HunyuanVideo demonstrates robust capabilities in generating videos that closely adhere to the provided text prompts. As illustrated in Figure 12, it effectively manages multiple subjects within the scene.

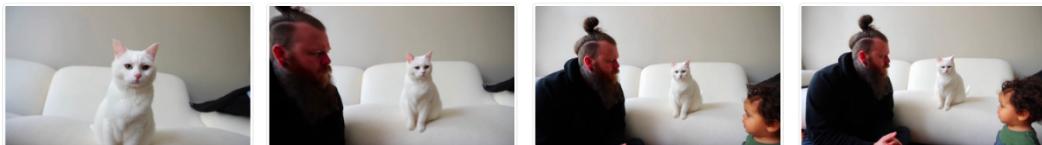


Figure 12: Prompt: A white cat sits on a white soft sofa like a person, while its long-haired male owner, with his hair tied up in a topknot, sits on the floor, gazing into the cat’s eyes. His child stands nearby, observing the interaction between the cat and the man.

High-quality We also perform a fine-tuning process to enhance the spatial quality of the generated videos. As illustrated in Figure 13, HunyuanVideo is capable of producing videos with ultra-detailed content.

High-motion Dynamics In this part, we demonstrate HunyuanVideo’s capabilities in producing high-dynamic videos based on given prompts. As shown in Figure 14, our model excels in generating videos that encompass a wide range of scenes and various types of motion.

Concept Generalization One of the most desirable features of a generative model is its ability to generalize concepts. As illustrated in Figure 15, the text prompt describes a scene: "In a distant galaxy, an astronaut floats on a shimmering, pink, gemstone-like lake that reflects the vibrant colors of the surrounding sky, creating a stunning scene. The astronaut gently drifts on the lake’s surface, while the soft sounds of water whisper the planet’s secrets. He reaches out, his fingertips gliding



(a) Prompt: the ultra-wide-angle lens follows closely from the hood, with raindrops continuously splattering against the lens. Ahead, a sports car speeds around a corner, its tires violently skidding against the wet road, creating a mist of water. Neon lights refract in the rain, leaving colorful streaks on the car's surface. The camera swiftly shifts to the side of the car, capturing the wheels spinning at high speed, before finally moving to the rear.



(b) Prompt: a stylish woman walks down a Tokyo street filled with warm glowing neon and animated city signage. She wears a black leather jacket, a long red dress, and black boots, and carries a black purse. She wears sunglasses and red lipstick. She walks confidently and casually. The street is damp and reflective, creating a mirror effect of the colorful lights. Many pedestrians walk about.

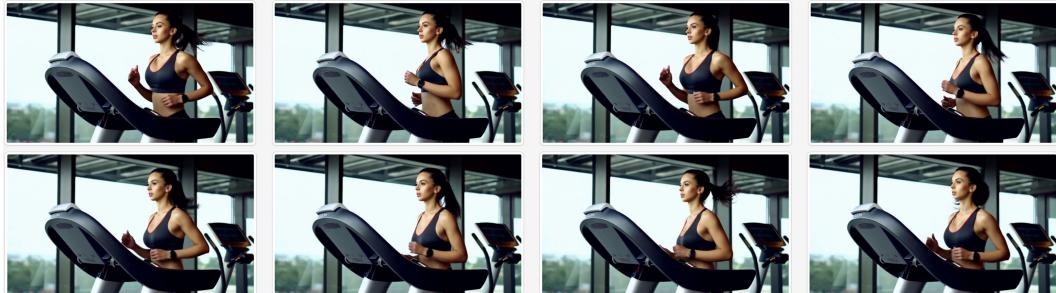
Figure 13: High-quality videos generated by HunyuanVideo.



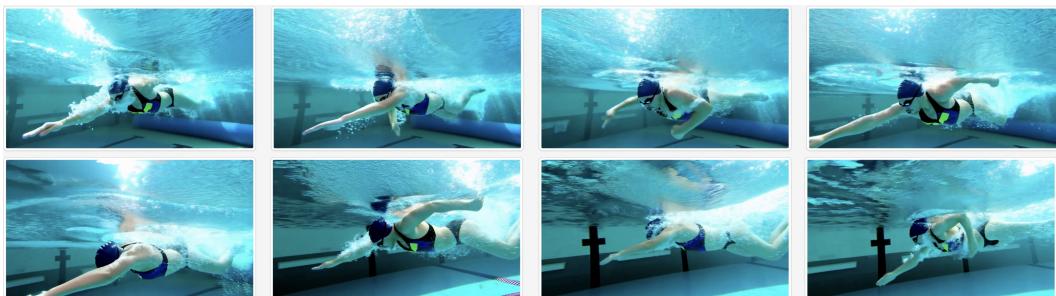
(a) Prompt: At sunset, a modified Ford F-150 Raptor roared past on the off-road track. The raised suspension allowed the huge explosion-proof tires to flip freely on the mud, and the mud splashed on the roll cage.



(b) Prompt: The panning camera moves forward slowly, with a depth of field in the middle focus, and warm sunset light covers the screen. The girl in the picture runs with her skirt fluttering, turns and jumps.



(c) Prompt: In the gym, a woman in workout clothes runs on a treadmill. Side angle. Realistic, Indoor lighting, Professional.



(d) Prompt: Swimmer swimming underwater, in slow motion. Realistic, Underwater lighting, Peaceful.



(e) Prompt: On the rooftop, there is an open-air basketball court, and five male students are playing basketball. Realistic, Natural lighting, Casual.

Figure 14: High-motion dynamics videos generated by HunyuanVideo.

over the cool, smooth water." Notably, this specific scenario has not been encountered in the training dataset. Furthermore, it is evident that the depicted scene combines several concepts that are also absent from the training data.

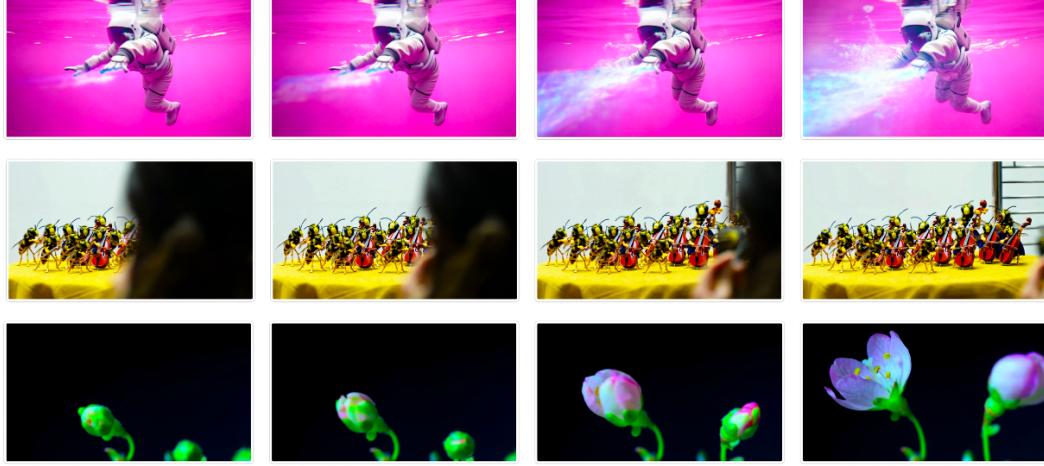


Figure 15: HunyuanVideo’s performance on concept generalization. The results of the three rows correspond to the text prompts (1) ‘In a distant galaxy, an astronaut floats on a shimmering, pink, gemstone-like lake that reflects the vibrant colors of the surrounding sky, creating a stunning scene. The astronaut gently drifts on the lake’s surface, the soft sounds of water whispering the planet’s secrets. He reaches out, his fingertips gliding over the cool, smooth water.’, (2) ‘A macro lens captures a tiny orchestra of insects playing instruments.’ and (3) ‘The night-blooming cactus flowers in the evening, with a brief, rapid closure. Time-lapse shot, extreme close-up. Realistic, Night lighting, Mysterious.’ respectively.

Action Reasoning and Planning Leveraging the capabilities of large language models, Hunyuan-Video can generate sequential movements based on a provided text prompt. As demonstrated in Figure 16, HunyuanVideo effectively captures all actions in a photorealistic style.



Figure 16: Prompt: The woman walks over and opens the red wooden door. As the door swings open, seawater bursts forth, in a realistic style.

Character Understanding and Writing HunyuanVideo is capable of generating both scene text and gradually appearing handwritten text as shown in Fig. 17.

6.1 Comparison with SOTA Models

To evaluate the performance of HunyuanVideo, we selected five strong baselines from closed-source video generation models. In total, we utilized 1,533 text prompts, generating an equal number of video samples with HunyuanVideo in a single run. For a fair comparison, we conducted inference only once, avoiding any cherry-picking of results. When comparing with the baseline methods, we maintained the default settings for all selected models, ensuring consistent video resolution. 60 professional evaluators performed the evaluation and the results are presented in Table 3. Videos



Figure 17: High text-video alignment videos generated by HunyuanVideo. Top row: Prompt: A close-up of a wave crashing against the beach, the sea foam spells out “WAKE UP” on the sand. Bottom row: Prompt: In a garden filled with blooming flowers, “GROW LOVE” has been spelled out with colorful petals.

were assessed based on three criteria: Text Alignment, Motion Quality, and Visual Quality. Notably, HunyuanVideo demonstrated the best overall performance, particularly excelling in motion quality. We randomly sample 600 videos out of 1533 for public access¹.

Table 3: Model Performance Evaluation

Model Name	Duration	Text Alignment	Motion Quality	Visual Quality	Overall	Ranking
HunyuanVideo (Ours)	5s	61.8%	66.5%	95.7%	41.3%	1
CNTopA (API)	5s	62.6%	61.7%	95.6%	37.7%	2
CNTopB (Web)	5s	60.1%	62.9%	97.7%	37.5%	3
GEN-3 alpha (Web)	6s	47.7%	54.7%	97.5%	27.4%	4
Luma1.6 (API)	5s	57.6%	44.2%	94.1%	24.8%	5
CNTopC (Web)	5s	48.4%	47.2%	96.3%	24.6%	6

7 Applications

7.1 Audio Generation based on Video

Our video-to-audio(V2A) module is designed to enhance generated video content by incorporating synchronized sound effects and contextually appropriate background music. Within the conventional film production pipeline, Foley sound design constitutes an integral component, significantly contributing to the auditory realism and emotional depth of visual media. However, the creation of Foley audio is both time-intensive and demands a high degree of professional expertise. With the advent of an increasing number of text-to-video (T2V) models, most of them lack the corresponding foley generation capabilities, thereby limiting their ability to produce fully immersive content. Our V2A module addresses this critical gap by autonomously generating cinematic-grade foley audio tailored to the input video and textual prompts, thus enabling the synthesis of a cohesive and holistically engaging multimedia experience.

7.1.1 Data

Unlike text-to-video (T2V) models, video-to-audio (V2A) models have different requirements for data. As mentioned above, we constructed a video dataset comprising of video-text pairs. However, not all data in this dataset are suitable for training the V2A model. For example, some videos lack an audio stream, others contain extensive voice-over content or their ambient audio tracks have been removed and replaced with unrelated elements. To address these challenges and ensure data quality, we designed a robust data filtering pipeline specifically tailored for V2A training.

First, we filter out videos without audio streams or those in which the silence ratio exceeds 80%. Next, we employ a frame-level audio detection model, like [38], to detect speech, music, and general sound

¹<https://github.com/Tencent/HunyuanVideo>

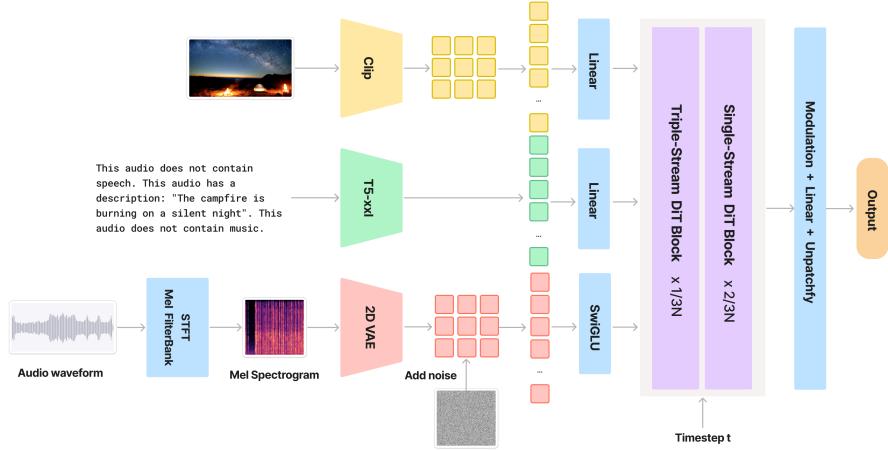


Figure 18: The architecture of sound effect and music generation model.

in the audio stream. Based on this analysis, we classify the data into four distinct categories: *pure sound*, *sound with speech*, *sound with music*, and *pure music*. Subsequently, to prioritize high-quality data, we train a model inspired by CAVP [54] to compute a visual-audio consistency score, which quantifies the alignment between the visual and auditory components of each video. Using this scoring system in conjunction with the audio category labels, we systematically sample portions of data from each category, retaining approximately 250,000 hours from the original dataset for pre-training. For the supervised fine-tuning stage, we further refine our selection, curating a subset of millions of high-quality clips (80,000 hours).

For feature extraction, we use CLIP [70] to obtain visual features at a temporal resolution of 4 fps and subsequently resample these features to align with the audio frame rate. To generate captions, we employ [29] as the sound captioning model and [20] as the music captioning model. When both sound and music captions are available, we merge them into a structured caption format, following the approach detailed in [67].

7.1.2 Model

Just like the above-mentioned text-to-video model, our video-to-audio generation model also adopts a flow-matching-based diffusion transformer (DiT) as its architectural backbone. The detailed design of the model is depicted in Figure 18, illustrating a transition from a triple-stream structure to a single-stream DiT framework.

The model operates within a latent space encoded by a variational autoencoder (VAE) trained on mel-spectrograms. Specifically, the audio waveform is first converted into a 2D mel-spectrogram representation. This spectrogram is subsequently encoded into a latent space using a pretrained VAE. For feature extraction, we leverage pretrained CLIP [70] and T5 [71] encoders to independently extract visual and textual features. These features are subsequently projected into the DiT-compatible latent space using independent linear projections followed by SwiGLU activation, as depicted in Figure 18.

To effectively integrate multimodal information, we incorporate stacked triple-stream transformer blocks, which independently process visual, audio, and textual modalities. These are later followed by single-stream transformer blocks to ensure seamless fusion and alignment across modalities. This design enhances the alignment between audio-video and audio-text representations, facilitating improved multimodal coherence.

Once the latent representation is generated by the diffusion transformer, the VAE decoder reconstructs the corresponding mel-spectrogram. Finally, the mel-spectrogram is converted back into an audio waveform using a pre-trained HiFiGAN vocoder [44]. This framework ensures a high-fidelity reconstruction of audio signals while maintaining strong multimodal alignment.

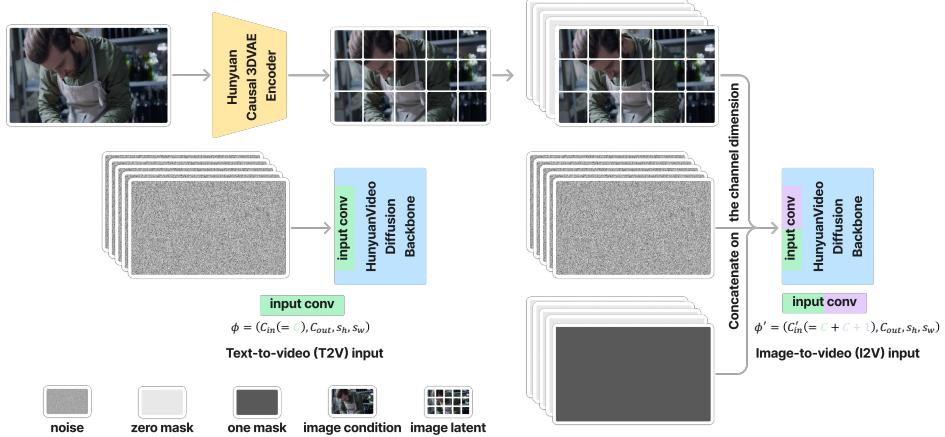


Figure 19: Differences between text-to-video (T2V) model and image-to-video (I2V) model.

7.2 Hunyuan Image-to-Video

7.2.1 Pre-training

Image-to-video (I2V) task is a common application in video generation tasks. It usually means that given an image and a caption, the model uses this image as the first frame to generate a video that matches the caption. Although the naive HunyuanVideo is a text-to-video (T2V) model, it can be easily extended to an I2V model. Specifically, as mentioned in Sec. 4.2, the T2V model’s input is a latent X with a shape of $T \times C \times H \times W$, where T , C , H and W represent the frame, channel, height and width of the compressed video respectively. Similar to Emu [25], in order to introduce image condition I , we treat I as the first frame of a video and apply zero-padding to create a $T \times C \times H \times W$ tensor I_o , as shown in Fig. 19. Additionally, we employ a binary mask m with dimensions $T \times 1 \times H \times W$, where the first temporal position is set to 1, and all other positions are set to zero. Then the latent X , the tensor I_o and the mask m are concatenated along the channel dimension to form the input for the model. Note that since the channel of the input tensor has changed from C to $2C + 1$, as shown in Fig. 19, we need to adjust the parameters of the first convolutional module of the model from $\phi = (C_{in} (= C), C_{out}, s_h, s_w)$ to $\phi' = (C'_{in} (= 2C + 1), C'_{out}, s_h, s_w)$, where each component corresponds to the input channel C_{in}/C'_{in} , output channel C_{out} , height of the convolutional kernel s_h , and width of the convolutional kernel s_w . In order to retain the representation ability of the T2V model, the first C input channels of ϕ' are directly copied to ϕ , and the rest are initialized to zero. We pre-train the I2V model on the same data as T2V model, and the results are shown in Fig. 20.

7.2.2 Downstream Task Fine-tuning: Portrait Image-to-Video Generation

We perform supervised finetuning of our I2V model on two million portrait videos to enhance human’s motion and overall aesthetics. In addition to the standard data filtering pipeline described in section 3, we also apply face and body detectors to filter out the training videos which have more than five persons. We also remove the videos in which the main subjects are small. Finally, the rest videos will be manually inspected to obtain the final high-quality portrait training dataset.

Regarding training, we adopt a progressive fine-tuning strategy, gradually unfreezing the model parameters of the respective layers while keeping the rest frozen during finetuning. This approach allows the model to achieve high performance in the portrait domain without compromising much of its inherent generalization ability, guaranteeing commendable performance in natural landscapes, animals, and plants domains. Moreover, our model also supports video interpolation by using the first and last frames as conditions. We randomly drop the text conditions at certain probability during training to enhance the model’s performance. Some demo results are shown in Fig. 21.

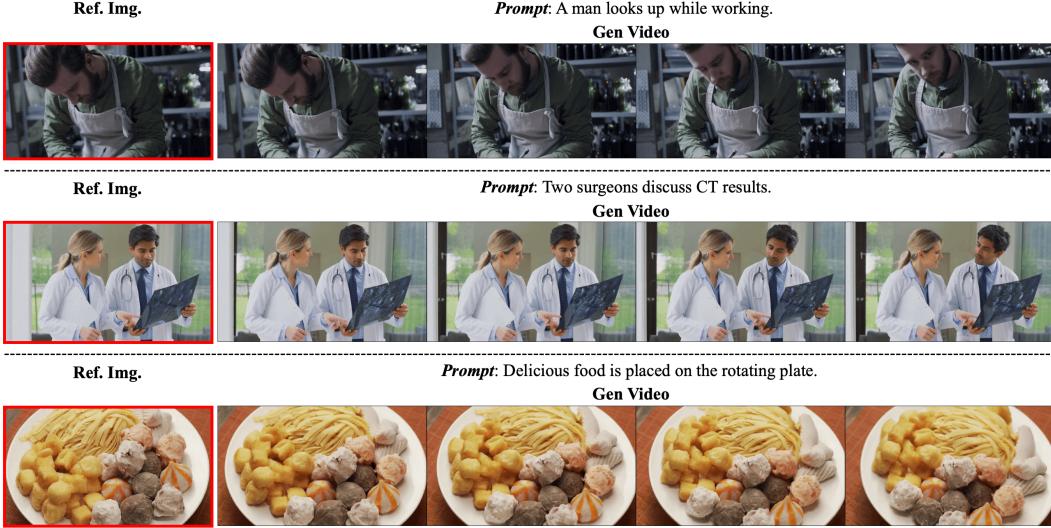


Figure 20: Sample results of the I2V pre-training model.

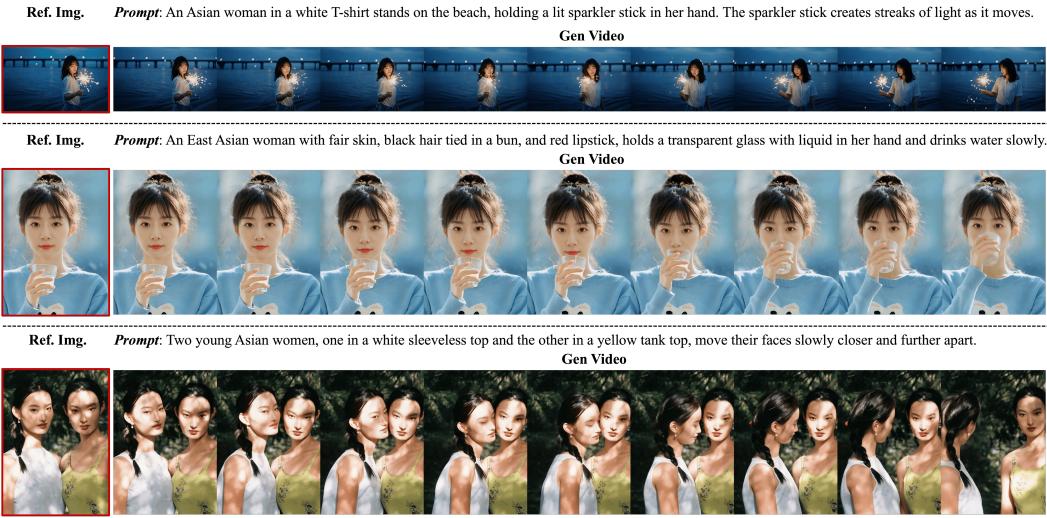


Figure 21: Sample results of our portrait I2V model.

7.3 Avatar animation

HunyuanVideo empowers controllable avatar animation in various aspects. It enables animating characters using explicit driving signals(e.g., speech signals, expression templates, and pose templates). In addition, it also integrates the implicit driving paradigm using text prompts. Fig. 22 shows how we leverage the power of HunyuanVideo to animate characters from multi-modal conditions. To maintain strict appearance consistency, we modify the HunyuanVideo architecture by inserting latent of reference image as strong guidance. As shown in Fig. 22 (b, c), we encode reference image using 3DVAE obtaining $z_{\text{ref}} \in \mathbb{R}^{1 \times c \times h \times w}$, where $c = 16$. Then we repeat it t times along temporal dimension and concatenate with z_t in channel dimension to get the modified noise input $\hat{z}_t \in \mathbb{R}^{t \times 2c \times h \times w}$. To achieve controllable animation, various adapters are employed. We describe them in following.

7.3.1 Upper-Body Talking Avatar Generation

In recent years, audio-driven digital human algorithms have made significant progress, especially in the performance of the talking head. Early algorithms, such as loopy [94], emo [80], and hallo [87],

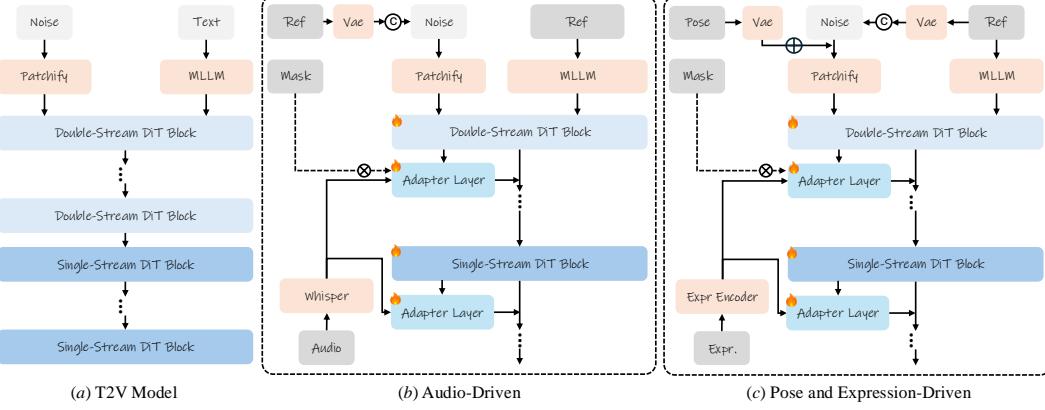


Figure 22: **Overview of Avatar Animation built on top of HunyuanVideo.** We adopt 3D VAE to encode and inject reference and pose condition, and use additional cross-attention layers to inject audio and expression signals. Masks are employed to explicitly guide where they are affecting.

mainly focused on the head area, driving the digital human’s facial expressions and lip shapes by analyzing audio signals. Even earlier algorithms, like wav2lip [68] and DINet [97], concentrated on modifying the mouth region in the input video to achieve lip shape consistency with the audio. However, these algorithms are usually limited to the head area, neglecting other parts of the body. To achieve a more natural and vivid digital human performance, we propose an audio-driven algorithm extended to the upper body. In this algorithm, the digital human not only synchronizes facial expressions and lip shapes with the audio while speaking but also moves the body rhythmically with the audio.

Audio-Driven Based on the input audio signal, our model can adaptively predict the digital human’s facial expressions and posture action information . This allows the driven character to speak with emotion and expression, enhancing the digital human’s expressiveness and realism. As shown in Fig. 22 (b), for the single audio signal-driven part, the audio passes through the whisper feature extraction module to obtain audio features, which are then injected into the main network in a cross-attention manner. It should be noted that the injection process will be multiplied by the face-mask to control the audio’s effect area. While enhancing the head and shoulder control ability, it will also greatly reduce the probability of body deformation. To obtain more lively head movements, head pose motion parameters and expression motion parameters are introduced and added to the time step in an embedding manner. During training, the head motion parameters are given by the variance of the nose tip keypoint sequence, and the expression parameters are given by the variance of the facial keypoints.

7.3.2 Fully-Controlled Whole-Body Avatar Generation

Controlling digital character’s motion and expression explicitly has been a long-standing problem in both academia and industry, and recent advancement of diffusion models paved the first step to realistic avatar animation. However, current avatar animation solutions suffer from partial controllability due to limited capability of foundation video generation model. We demonstrate that a stronger T2V model boosts the avatar video generation to fully-controllable stage. We show how HunyuanVideo serves as strong foundation with limited modifications to extent general T2V model to fully-controllable avatar generation model in Fig. 22 (c).

Pose-Driven We can control the digital character’s body movements explicitly using pose templates. We use Dwpose [92] to detect skeletal video from any source video, and use 3DVAE to transform it to latent space as z_{pose} . We argue that this eases the fine-tuning process because both input and driving videos are in image representation, and are encoded with shared VAE, resulting same latent space. We then inject the driving signals to the model by element-wise add as $\hat{z}_t + z_{\text{pose}}$. Note that \hat{z}_t contains the appearance information of reference image. We use full-parameters finetune with pretrained T2V weights as initialization.

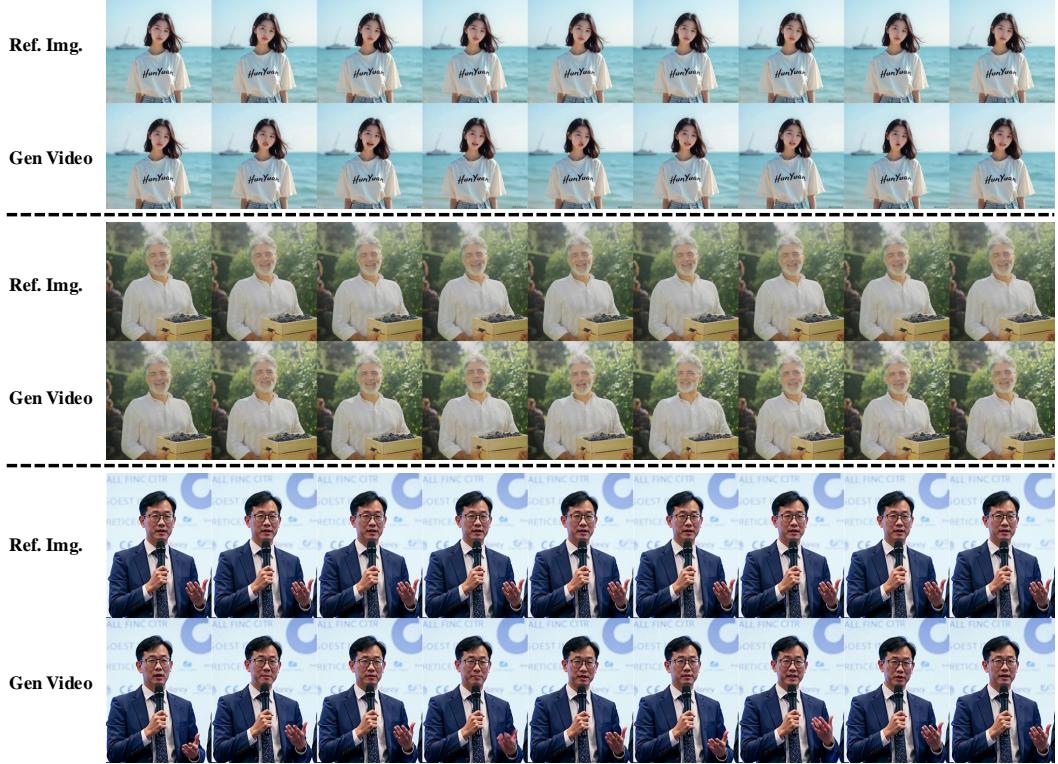


Figure 23: **Audio-Driven.** HunyuanVideo can generate vivid talking avatar videos.

Expression-Driven We can also control the facial expressions of digital character using implicit expression representations. Although facial landmarks are widely adopted in this area [58, 16], we argue using landmarks brings ID leak due to cross-ID misalignment. Instead, we use implicit representations as driving signals for their ID and expression disentanglement capabilities. In this work, we use VASA [88] as expression extractor. As shown in Fig. 22 (c), we adopt a light-weight expression encoder to transform the expression representation to token sequence in latent space as $z_{\text{exp}} \in \mathbb{R}^{t \times n \times c}$, where n is the number of tokens per frame. Typically, we set $n = 16$. Unlike pose condition, we inject z_{exp} using cross-attention because \hat{z}_t and z_{exp} are not naturally aligned in spatial aspect. We add cross-attention layer $\text{Attn}_{\text{exp}}(q, k, v)$ every K double and single-stream DiT layers to inject expression latent. Denote the hidden states after i -th DiT layer as h_i , the injection of expression z_{exp} to h_i could be derived as: $h_i + \text{Attn}_{\text{exp}}(h_i, z_{\text{exp}}, z_{\text{exp}}) * \mathcal{M}_{\text{face}}$, where $\mathcal{M}_{\text{face}}$ is the face region mask that guides where z_{exp} should be applied at, and $*$ stands for element-wise multiplication. Also, full-parameters tuning strategy is adopted.

Hybrid Condition Driven Combining both pose and expression driven strategies derives hybrid control approach. In this scenario, the body motion is controlled by explicit skeletal pose sequence, and the facial expression is determined by implicit expression representation. We jointly fine-tune T2V modules and added modules in an end-to-end fashion. During inference, the body motion and facial motion could be controlled by separate driving signals, empowering richer editability.

7.4 Application Demos

We present extensive results of avatar animations to show the superiority and potential of bringing avatar animation empowered by HunyuanVideo to next generation.

Audio-Driven Fig. 23 shows that HunyuanVideo serves as a strong foundation model for audio-driven avatar animation, which can synthesize vivid and high-fidelity videos. We summarize the superiority of our method in three folds:

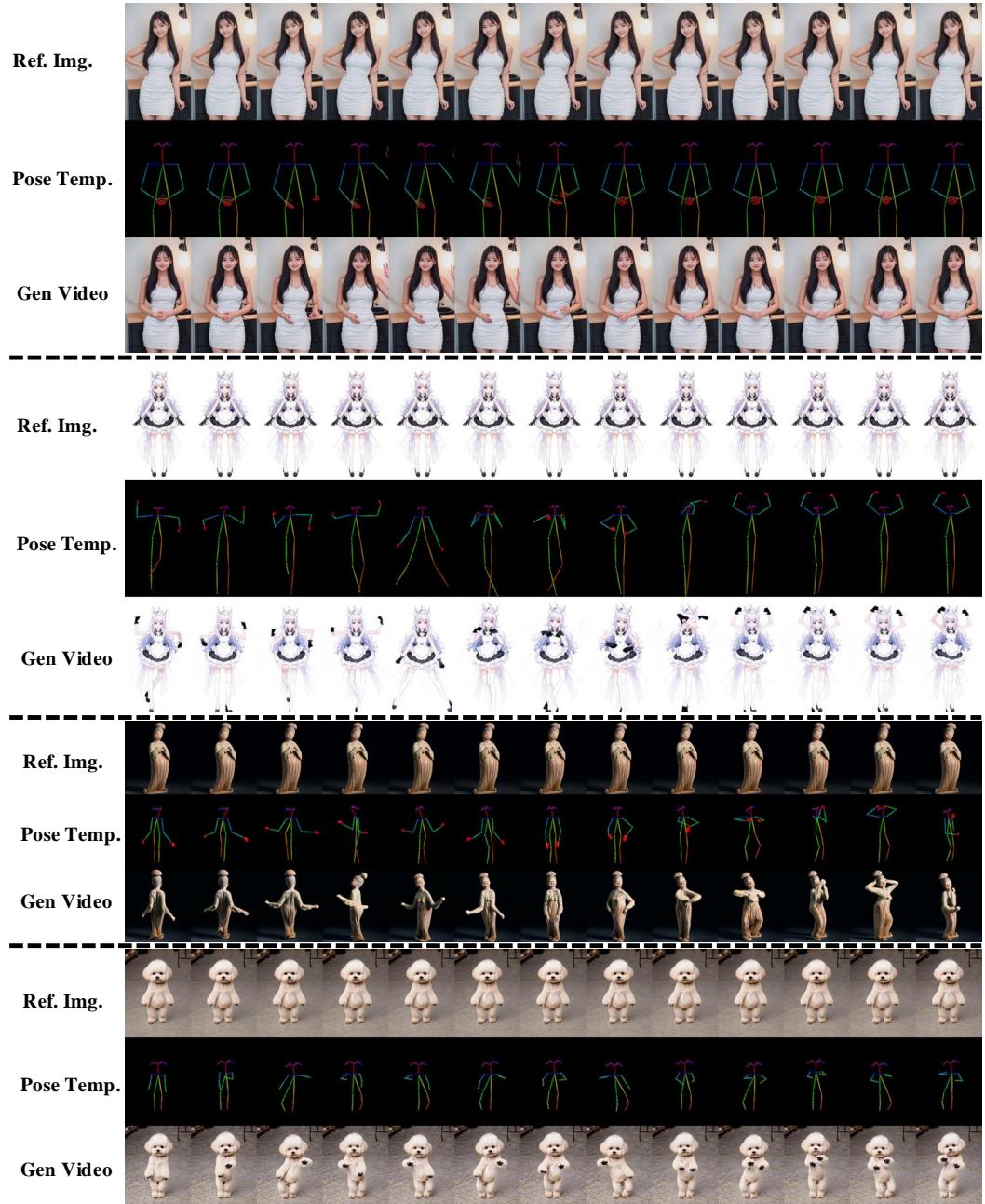


Figure 24: **Pose-Driven.** HunyuanVideo can animate wide variety of characters with high quality and appearance consistency under various poses.

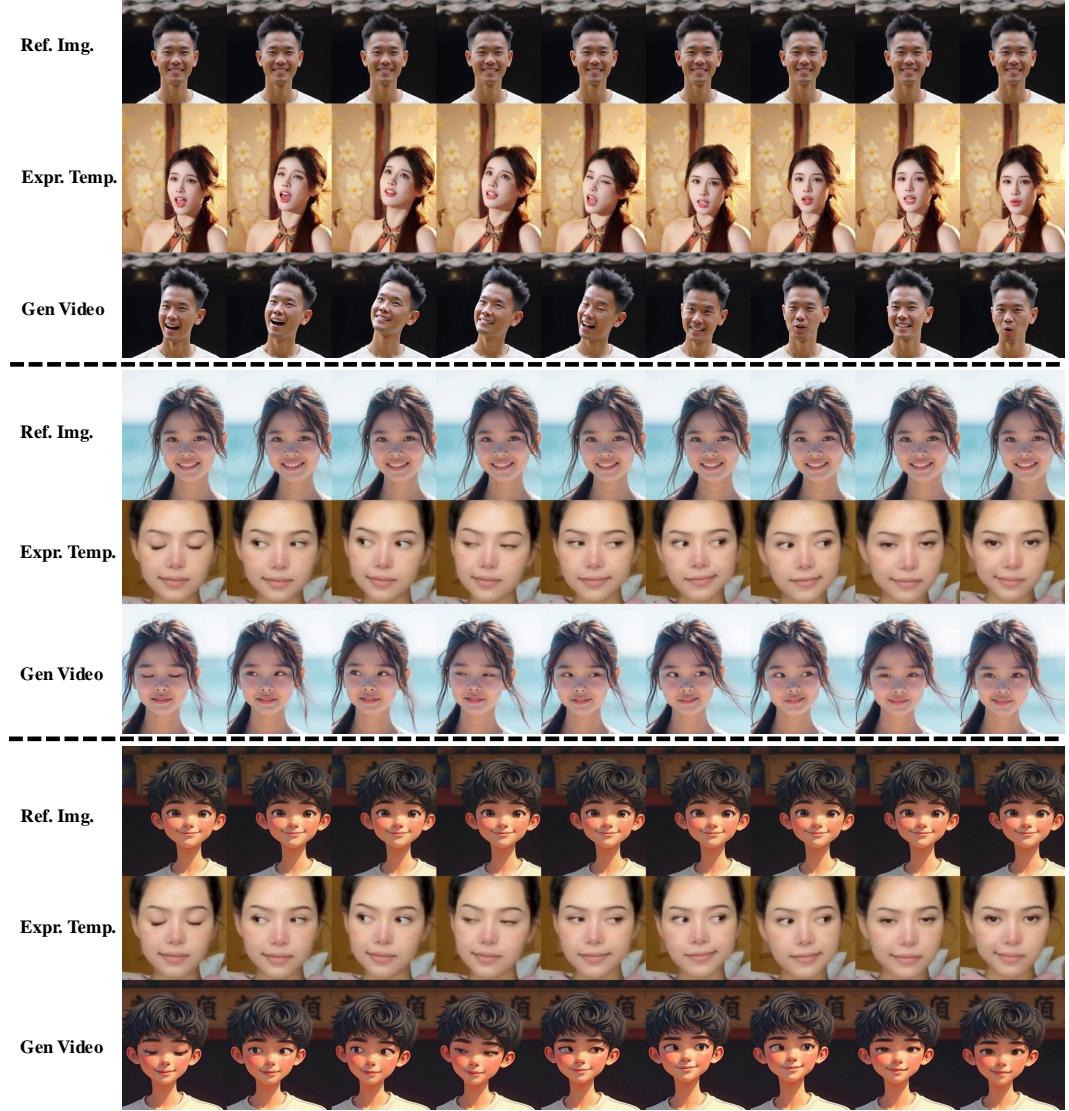


Figure 25: **Expression-Driven.** HunyuanVideo can accurately control facial movements of wide-variety of avatar styles.

- **Upper-body Animation.** Our method can drive not only portrait characters, but also upper-body avatar images, enlarging its range of application scenarios.
- **Dynamic Scene Modelling.** Our method can generate videos with vivid and realistic background motion, such as the wave undulation, crowd movement, and breeze stirring leaves.
- **Vivid Avatar Movements.** Our method is able to animate the character talking while gesturing vividly with audio solely.

Pose-Driven We also show that HunyuanVideo boosts the performance of pose-driven animation largely in many aspects in Fig. 24:

- **High ID-Consistency.** Our method maintains the ID-consistency well over the frames even with large poses, making it face-swapping free, thereby, could be used as real end-to-end animation solution.
- **Following Complex Poses Accurately.** Our method is able to handle very complex poses such as turning around and hands crossed.

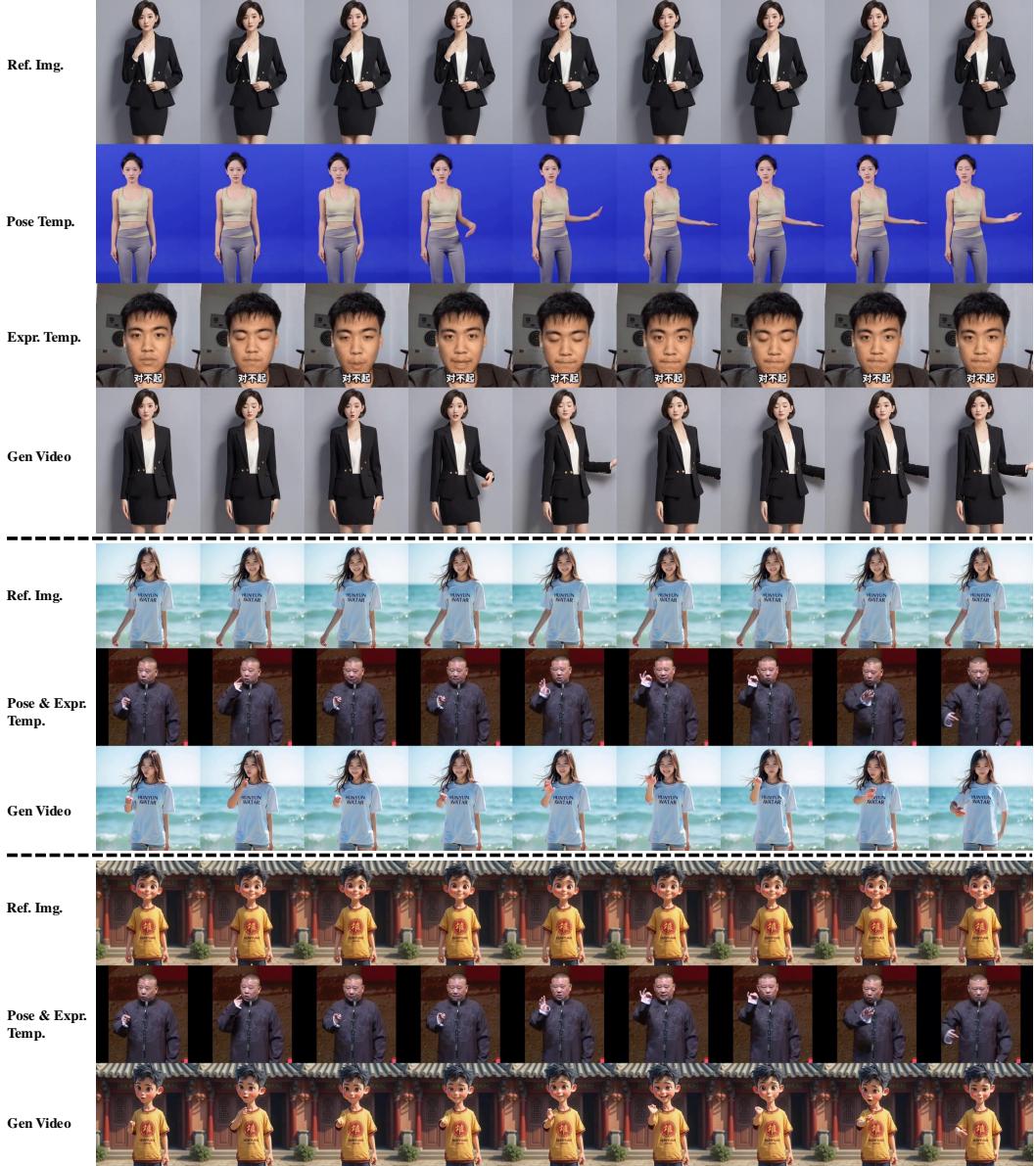


Figure 26: **Hybrid Condition-Driven.** HunyuanVideo supports full control with multiple driving sources across various avatar characters.

- **High Motion Quality.** Our method has remarkable capability in dynamic modelling. For instance, the results show promising performance in terms of garment dynamics and texture consistency.
- **Generalizability.** Our method presents surprisingly high generalizability. It can animate wide variety of avatar images, such as real human, anime, pottery figurine, and even animals.

Expression-Driven Fig. 25 presents how HunyuanVideo enhances the portrait expression animating in three folds:

- **Exaggerated Expression.** Our method is able to animate given portrait to mimic any facial movements even with large poses and exaggerated expressions.
- **Mimicing Eye Gaze Accurately.** We can control the portraits’ eye movements accurately given any expression template, even with extreme and large eye balls movements.

- **Generalizability.** Our method has high generalizability. It can animate not only real human portraits, but also anime or CGI characters.

Hybrid-Driven Lastly, we show that hybrid condition control reveals the potential of fully controllable and editable avatars in Fig. 26. We highlight the superiority as follow:

- **Hybrid Condition Control.** For the first time, our method is able to conduct full control over body and facial motions with siloed or multiple signals, paving the route from demo to applications for avatar animation.
- **Half-body Animation.** Our method supports upper-body full control, enabling rich editability while maintaining high quality and fidelity.
- **Generalizability.** Our method generalize to both real human images and CGI characters.

8 Related Works

Due to the success of diffusion models in the field of image generation [72, 34], the exploration in the domain of video generation [28, 40, 75, 84, 91, 96, 58, 12, 90, 57] is also becoming popular. VDM [32] is among the first that extends the 2D U-Net from image diffusion models to a 3D U-Net to achieve text-based generation. Later works, such as MagicVideo [103] and Mindscope [82], introduce 1D temporal attention mechanisms, reducing computations by building upon latent diffusion models. In this report, we do not use the 2D + 1D temporal block manner for motion learning. Instead, we use similar dual flow attention blocks as in FLUX [47], which are used for processing all video frames. Following Imagen, Imagen Video [33] employs a cascaded sampling pipeline that generates videos through multiple stages. In addition to traditional end-to-end text-to-video (T2V) generation, video generation using other conditions is also an important direction. This type of methods generates videos with other auxiliary controls, such as depth maps [27, 31], pose maps [89, 37, 83, 56], RGB images [5, 15, 61], or other guided motion videos [100, 86]. Despite the excellent generation performance of the recent open-source models such as Stable video diffusion [5], Open-sora [102], Open-sora-plan [46], Mochi-1 [79] and Allegro [104], their performance still falls far behind the closed-source state-of-the-art video generation models such as Sora [7] and MovieGen [67].

Project Contributors

- **Project Sponsors:** Jie Jiang, Yuhong Liu, Di Wang, Yong Yang
- **Project Leaders:** Caesar Zhong, Hongfa Wang, Dax Zhou, Songtao Liu, Qinglin Lu, Yangyu Tao
- **Core Contributors:**
 - **Infrastructure:** Rox Min, Jinbao Xue, Yuanbo Peng, Fang Yang, Shuai Li, Weiyang Wang, Kai Wang
 - **Data & Recaptioning:** Zuozhuo Dai, Xin Li, Jin Zhou, Junkun Yuan, Hao Tan, Xinchi Deng, Zhiyu He, Duojun Huang, Andong Wang, Mengyang Liu, Pengyu Li
 - **VAE & Model Distillation:** Bo Wu, Rox Min, Changlin Li, Jiawang Bai, Yang Li, Jianbing Wu
 - **Algorithm & Model Architecture & Pre-training:** Weijie Kong, Qi Tian, Jianwei Zhang, Zijian Zhang, Kathrina Wu, Jiangfeng Xiong, Yanxin Long
 - **Downstream Tasks:** Jacob Song, Jin Zhou, Yutao Cui, Aladdin Wang, Wenqing Yu, Zhiyong Xu, Zixiang Zhou, Zhentao Yu, Yi Chen, Hongmei Wang, Zunnan Xu, Joey Wang, Qin Lin
- **Contributors:** Jihong Zhang, Meng Chen, Jianchen Zhu, Winston Hu, Yongming Rao, Kai Liu, Lifei Xu, Sihuan Lin, Yifu Sun, Shirui Huang, Lin Niu, Shisheng Huang, Yongjun Deng, Kaibo Cao, Xuan Yang, Hao Zhang, Jiaxin Lin, Chao Zhang, Fei You, Yuanbin Chen, Yuhui Hu, Liang Dong Zheng Fang, Dian Jiao, Zhijiang Xu

References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 9
- [2] 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. 9
- [3] Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22669–22679, 2023. 9
- [4] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science*. <https://cdn.openai.com/papers/dall-e-3.pdf>, 2(3):8, 2023. 4
- [5] 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*, 2023. 2, 27
- [6] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 2
- [7] Tim Brooks, Bill Peebles, Connor Homes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Wing Yin Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. 4, 7, 27
- [8] Tom B Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. 8
- [9] Lin Chen, Jisong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. Sharegpt4v: Improving large multi-modal models with better captions. *arXiv preprint arXiv:2311.12793*, 2023. 4
- [10] Lin Chen, Xilin Wei, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Bin Lin, Zhenyu Tang, et al. Sharegpt4video: Improving video understanding and generation with better captions. *arXiv preprint arXiv:2406.04325*, 2024. 4
- [11] Liuhan Chen, Zongjian Li, Bin Lin, Bin Zhu, Qian Wang, Shenghai Yuan, Xing Zhou, Xinghua Cheng, and Li Yuan. Od-vae: An omni-dimensional video compressor for improving latent video diffusion model. *arXiv preprint arXiv:2409.01199*, 2024. 6
- [12] Qihua Chen, Yue Ma, Hongfa Wang, Junkun Yuan, Wenzhe Zhao, Qi Tian, Hongmei Wang, Shaobo Min, Qifeng Chen, and Wei Liu. Follow-your-canvas: Higher-resolution video outpainting with extensive content generation. *arXiv preprint arXiv:2409.01055*, 2024. 27
- [13] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018. 10
- [14] Tsai-Shien Chen, Aliaksandr Siarohin, Willi Menapace, Ekaterina Deyneka, Hsiang-Wei Chao, Byung Eun Jeon, Yuwei Fang, Hsin-Ying Lee, Jian Ren, Ming-Hsuan Yang, and Sergey Tulyakov. Panda-70m: Captioning 70m videos with multiple cross-modality teachers. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 13320–13331. IEEE, June 2024. 4
- [15] Xinyuan Chen, Yaohui Wang, Lingjun Zhang, Shaobin Zhuang, Xin Ma, Jiashuo Yu, Yali Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. Seine: Short-to-long video diffusion model for generative transition and prediction. *arXiv preprint*, 2023. 27
- [16] Zhiyuan Chen, Jiajiong Cao, Zhiqian Chen, Yuming Li, and Chenguang Ma. Echomimic: Lifelike audio-driven portrait animations through editable landmark conditions. *arXiv preprint arXiv:2407.08136*, 2024. 23

- [17] XTuner Contributors. Xtuner: A toolkit for efficiently fine-tuning llm. <https://github.com/InternLM/xtuner>, 2023. 8
- [18] OpenCV Developers. Opencv. <https://opencv.org/>. 3
- [19] PySceneDetect Developers. Pyscenedetect. <https://www.scenedetect.com/>. 3
- [20] SeungHeon Doh, Keunwoo Choi, Jongpil Lee, and Juhan Nam. Lp-musiccaps: Llm-based pseudo music captioning. *arXiv preprint arXiv:2307.16372*, 2023. 19
- [21] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024. 2, 8, 9, 10
- [22] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021. 6
- [23] Peng Gao, Le Zhuo, Ziyi Lin, Chris Liu, Junsong Chen, Ruoyi Du, Enze Xie, Xu Luo, Longtian Qiu, Yuhang Zhang, et al. Lumina-t2x: Transforming text into any modality, resolution, and duration via flow-based large diffusion transformers. *arXiv preprint arXiv:2405.05945*, 2024. 9
- [24] Z Ge. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021. 3
- [25] Rohit Girdhar, Mannat Singh, Andrew Brown, Quentin Duval, Samaneh Azadi, Sai Saketh Rambhatla, Akbar Shah, Xi Yin, Devi Parikh, and Ishan Misra. Emu video: Factorizing text-to-video generation by explicit image conditioning. *arXiv preprint arXiv:2311.10709*, 2023. 2, 20
- [26] Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, and Zihan Wang. Chatglm: A family of large language models from glm-130b to glm-4 all tools, 2024. 8
- [27] Yuwei Guo, Ceyuan Yang, Anyi Rao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Sparsectrl: Adding sparse controls to text-to-video diffusion models. *arXiv preprint*, 2023. 27
- [28] Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *ICLR*, 2024. 27
- [29] Moayed Haji-Ali, Willi Menapace, Aliaksandr Siarohin, Guha Balakrishnan, Sergey Tulyakov, and Vicente Ordonez. Taming data and transformers for audio generation. *arXiv preprint arXiv:2406.19388*, 2024. 19
- [30] Pieter Abbeel, Hao Liu, Matei Zaharia. Ring attention with blockwise transformers for near-infinite context. *arXiv preprint arXiv:2310.01889*, 2023. 14
- [31] Yingqing He, Menghan Xia, Haoxin Chen, Xiaodong Cun, Yuan Gong, Jinbo Xing, Yong Zhang, Xintao Wang, Chao Weng, Ying Shan, et al. Animate-a-story: Storytelling with retrieval-augmented video generation. *arXiv preprint*, 2023. 27
- [32] J Ho, T Salimans, A Gritsenko, W Chan, M Norouzi, and DJ Fleet. Video diffusion models. arxiv 2022. *arXiv preprint*, 2022. 27
- [33] 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*, 2022. 27

- [34] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 10, 27
- [35] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint*, 2022. 13
- [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. 8, 9, 10
- [37] Li Hu, Xin Gao, Peng Zhang, Ke Sun, Bang Zhang, and Liefeng Bo. Animate anyone: Consistent and controllable image-to-video synthesis for character animation. *arXiv preprint*, 2023. 27
- [38] Yun-Ning Hung, Chih-Wei Wu, Iroro Orife, Aaron Hippel, William Wolcott, and Alexander Lerch. A large tv dataset for speech and music activity detection. *EURASIP Journal on Audio, Speech, and Music Processing*, 2022(1):21, 2022. 18
- [39] Investopedia. General data protection regulation (gdpr), n.d. Accessed October 10, 2023. 3
- [40] Yuming Jiang, Shuai Yang, Tong Liang Koh, Wayne Wu, Chen Change Loy, and Ziwei Liu. Text2performer: Text-driven human video generation. *arXiv preprint*, 2023. 27
- [41] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 8, 9
- [42] Maciej Kilian, Varun Japan, and Luke Zettlemoyer. Computational tradeoffs in image synthesis: Diffusion, masked-token, and next-token prediction. *arXiv preprint arXiv:2405.13218*, 2024. 9
- [43] Juyeon Kim, Jeongeun Lee, Yoonho Chang, Chanyeol Choi, Junseong Kim, and Jy yong Sohn. Re-ex: Revising after explanation reduces the factual errors in llm responses, 2024. 12
- [44] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in neural information processing systems*, 33:17022–17033, 2020. 19
- [45] Vijay Anand Korthikanti, Jared Casper, Sangkug Lym, Lawrence McAfee, Michael Andersch, Mohammad Shoeybi, and Bryan Catanzaro. Reducing activation recomputation in large transformer models. *Proceedings of Machine Learning and Systems*, 5:341–353, 2023. 13
- [46] PKU-Yuan Lab and Tuzhan AI etc. Open-sora-plan, April 2024. 27
- [47] Black Forest Labs. Flux, 2024. 2, 6, 8, 27
- [48] Baojia Li, Xiaoliang Wang, Jingzhu Wang, Yifan Liu, Yuanyuan Gong, Hao Lu, Weizhen Dang, Weifeng Zhang, Xiaojie Huang, Mingzhuo Chen, et al. Tccl: Co-optimizing collective communication and traffic routing for gpu-centric clusters. In *Proceedings of the 2024 SIGCOMM Workshop on Networks for AI Computing*, pages 48–53, 2024. 13
- [49] Hao Li, Yang Zou, Ying Wang, Orchid Majumder, Yusheng Xie, R Manmatha, Ashwin Swaminathan, Zhuowen Tu, Stefano Ermon, and Stefano Soatto. On the scalability of diffusion-based text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9400–9409, 2024. 9
- [50] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022. 4
- [51] Zhimin Li, Jianwei Zhang, Qin Lin, Jiangfeng Xiong, Yanxin Long, Xinchi Deng, Yingfang Zhang, Xingchao Liu, Minbin Huang, Zedong Xiao, Dayou Chen, Jiajun He, Jiahao Li, Wenyue Li, Chen Zhang, Rongwei Quan, Jianxiang Lu, Jiabin Huang, Xiaoyan Yuan, Xiaoxiao Zheng, Yixuan Li, Jihong Zhang, Chao Zhang, Meng Chen, Jie Liu, Zheng Fang, Weiyan Wang,

- Jinbao Xue, Yangyu Tao, Jianchen Zhu, Kai Liu, Sihuan Lin, Yifu Sun, Yun Li, Dongdong Wang, Mingtao Chen, Zhichao Hu, Xiao Xiao, Yan Chen, Yuhong Liu, Wei Liu, Di Wang, Yong Yang, Jie Jiang, and Qinglin Lu. Hunyuan-dit: A powerful multi-resolution diffusion transformer with fine-grained chinese understanding, 2024. [2](#), [8](#)
- [52] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. [2](#), [10](#)
- [53] Haotian Liu, Chunyuan Li, Qingsong Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024. [8](#)
- [54] Simian Luo, Chuanhao Yan, Chenxu Hu, and Hang Zhao. Diff-foley: Synchronized video-to-audio synthesis with latent diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024. [19](#)
- [55] Bingqi Ma, Zhuofan Zong, Guanglu Song, Hongsheng Li, and Yu Liu. Exploring the role of large language models in prompt encoding for diffusion models. *arXiv preprint arXiv:2406.11831*, 2024. [8](#)
- [56] Yue Ma, Yingqing He, Xiaodong Cun, Xintao Wang, Ying Shan, Xiu Li, and Qifeng Chen. Follow your pose: Pose-guided text-to-video generation using pose-free videos. *arXiv preprint*, 2023. [27](#)
- [57] Yue Ma, Yingqing He, Hongfa Wang, Andong Wang, Chenyang Qi, Chengfei Cai, Xiu Li, Zhifeng Li, Heung-Yeung Shum, Wei Liu, et al. Follow-your-click: Open-domain regional image animation via short prompts. *arXiv preprint arXiv:2403.08268*, 2024. [27](#)
- [58] Yue Ma, Hongyu Liu, Hongfa Wang, Heng Pan, Yingqing He, Junkun Yuan, Ailing Zeng, Chengfei Cai, Heung-Yeung Shum, Wei Liu, et al. Follow-your-emoji: Fine-controllable and expressive freestyle portrait animation. *arXiv preprint arXiv:2406.01900*, 2024. [23](#), [27](#)
- [59] J MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability/University of California Press*, 1967. [3](#)
- [60] Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14297–14306, 2023. [13](#)
- [61] Haomiao Ni, Changhao Shi, Kai Li, Sharon X Huang, and Martin Renqiang Min. Conditional image-to-video generation with latent flow diffusion models. In *CVPR*, 2023. [27](#)
- [62] Xiaonan Nie, Yi Liu, Fangcheng Fu, Jinbao Xue, Dian Jiao, Xupeng Miao, Yangyu Tao, and Bin Cui. Angel-ptm: A scalable and economical large-scale pre-training system in tencent. *arXiv preprint arXiv:2303.02868*, 2023. [13](#)
- [63] NVIDIA. Context parallelism overview. 2024. [13](#)
- [64] NVIDIA. Cosmos-tokenizer, 2024. [6](#)
- [65] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. [2](#), [9](#)
- [66] 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*, 2023. [8](#), [11](#)
- [67] Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024. [2](#), [5](#), [6](#), [7](#), [13](#), [19](#), [27](#)
- [68] KR Prajwal, Rudrabha Mukhopadhyay, Vinay P Namboodiri, and CV Jawahar. A lip sync expert is all you need for speech to lip generation in the wild. In *Proceedings of the 28th ACM international conference on multimedia*, pages 484–492, 2020. [22](#)

- [69] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 8
- [70] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. PMLR, 2021. 19
- [71] 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(140):1–67, 2020. 8, 10, 19
- [72] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 2, 27
- [73] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022. 10
- [74] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019. 13
- [75] 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*, 2022. 27
- [76] Tomáš Souček and Jakub Lokoč. Transnet v2: An effective deep network architecture for fast shot transition detection. *arXiv preprint arXiv:2008.04838*, 2020. 3
- [77] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023. 8
- [78] Xingwu Sun, Yanfeng Chen, Yiqing Huang, Ruobing Xie, Jiaqi Zhu, Kai Zhang, Shuaipeng Li, Zhen Yang, Jonny Han, Xiaobo Shu, Jiahao Bu, Zhongzhi Chen, Xuemeng Huang, Fengzong Lian, Saiyong Yang, Jianfeng Yan, Yuyuan Zeng, Xiaoqin Ren, Chao Yu, Lulu Wu, Yue Mao, Tao Yang, Suncong Zheng, Kan Wu, Dian Jiao, Jinbao Xue, Xipeng Zhang, Decheng Wu, Kai Liu, Dengpeng Wu, Guanghui Xu, Shaohua Chen, Shuang Chen, Xiao Feng, Yigeng Hong, Junqiang Zheng, Chengcheng Xu, Zongwei Li, Xiong Kuang, Jianglu Hu, Yiqi Chen, Yuchi Deng, Guiyang Li, Ao Liu, Chenchen Zhang, Shihui Hu, Zilong Zhao, Zifan Wu, Yao Ding, Weichao Wang, Han Liu, Roberts Wang, Hao Fei, Peijie She, Ze Zhao, Xun Cao, Hai Wang, Fusheng Xiang, Mengyuan Huang, Zhiyuan Xiong, Bin Hu, Xuebin Hou, Lei Jiang, Jiajia Wu, Yaping Deng, Yi Shen, Qian Wang, Weijie Liu, Jie Liu, Meng Chen, Liang Dong, Weiwen Jia, Hu Chen, Feifei Liu, Rui Yuan, Huilin Xu, Zhenxiang Yan, Tengfei Cao, Zhichao Hu, Xinhua Feng, Dong Du, Tinghao She, Yangyu Tao, Feng Zhang, Jianchen Zhu, Chengzhong Xu, Xirui Li, Chong Zha, Wen Ouyang, Yinben Xia, Xiang Li, Zekun He, Rongpeng Chen, Jiawei Song, Ruibin Chen, Fan Jiang, Chongqing Zhao, Bo Wang, Hao Gong, Rong Gan, Winston Hu, Zhanhui Kang, Yong Yang, Yuhong Liu, Di Wang, and Jie Jiang. Hunyan-large: An open-source moe model with 52 billion activated parameters by tencent, 2024. 8, 11
- [79] Genmo Team. Mochi 1: A new sota in open-source video generation. <https://github.com/genmoai/models>, 2024. 7, 27
- [80] Linrui Tian, Qi Wang, Bang Zhang, and Liefeng Bo. Emo: Emote portrait alive – generating expressive portrait videos with audio2video diffusion model under weak conditions, 2024. 21
- [81] 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. 9
- [82] Jiuniu Wang, Hangjie Yuan, Dayou Chen, Yingya Zhang, Xiang Wang, and Shiwei Zhang. Modelscope text-to-video technical report. *arXiv preprint*, 2023. 27

- [83] Tan Wang, Linjie Li, Kevin Lin, Chung-Ching Lin, Zhengyuan Yang, Hanwang Zhang, Zicheng Liu, and Lijuan Wang. Disco: Disentangled control for referring human dance generation in real world. *arXiv preprint*, 2023. 27
- [84] Yaohui Wang, Xinyuan Chen, Xin Ma, Shangchen Zhou, Ziqi Huang, Yi Wang, Ceyuan Yang, Yinan He, Jiashuo Yu, Peiqing Yang, et al. Lavie: High-quality video generation with cascaded latent diffusion models. *arXiv preprint*, 2023. 27
- [85] Haoning Wu, Erli Zhang, Liang Liao, Chaofeng Chen, Jingwen Hou, Annan Wang, Wenxiu Sun, Qiong Yan, and Weisi Lin. Exploring video quality assessment on user generated contents from aesthetic and technical perspectives. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20144–20154, 2023. 3
- [86] Ruiqi Wu, Liangyu Chen, Tong Yang, Chunle Guo, Chongyi Li, and Xiangyu Zhang. Lamp: Learn a motion pattern for few-shot-based video generation. *arXiv preprint*, 2023. 27
- [87] Mingwang Xu, Hui Li, Qingkun Su, Hanlin Shang, Liwei Zhang, Ce Liu, Jingdong Wang, Yao Yao, and Siyu Zhu. Hallo: Hierarchical audio-driven visual synthesis for portrait image animation, 2024. 21
- [88] Sicheng Xu, Guojun Chen, Yu-Xiao Guo, Jiaolong Yang, Chong Li, Zhenyu Zang, Yizhong Zhang, Xin Tong, and Baining Guo. Vasa-1: Lifelike audio-driven talking faces generated in real time. *arXiv preprint arXiv:2404.10667*, 2024. 23
- [89] Zhongcong Xu, Jianfeng Zhang, Jun Hao Liew, Hanshu Yan, Jia-Wei Liu, Chenxu Zhang, Jiashi Feng, and Mike Zheng Shou. Magicanimate: Temporally consistent human image animation using diffusion model. *arXiv preprint*, 2023. 27
- [90] Jingyun Xue, Hongfa Wang, Qi Tian, Yue Ma, Andong Wang, Zhiyuan Zhao, Shaobo Min, Wenzhe Zhao, Kaihao Zhang, Heung-Yeung Shum, et al. Follow-your-pose v2: Multiple-condition guided character image animation for stable pose control. *arXiv preprint arXiv:2406.03035*, 2024. 27
- [91] Mengjiao Yang, Yilun Du, Bo Dai, Dale Schuurmans, Joshua B Tenenbaum, and Pieter Abbeel. Probabilistic adaptation of text-to-video models. *arXiv preprint*, 2023. 27
- [92] Zhendong Yang, Ailing Zeng, Chun Yuan, and Yu Li. Effective whole-body pose estimation with two-stages distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4210–4220, 2023. 22
- [93] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024. 4, 5, 6, 7
- [94] Zhenhui Ye, Tianyun Zhong, Yi Ren, Ziyue Jiang, Jiawei Huang, Rongjie Huang, Jinglin Liu, Jinzheng He, Chen Zhang, Zehan Wang, Xize Chen, Xiang Yin, and Zhou Zhao. Mimictalk: Mimicking a personalized and expressive 3d talking face in minutes, 2024. 21
- [95] Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Vighnesh Birodkar, Agrim Gupta, Xiuye Gu, et al. Language model beats diffusion–tokenizer is key to visual generation. *arXiv preprint arXiv:2310.05737*, 2023. 5
- [96] 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*, 2023. 27
- [97] Zhimeng Zhang, Zhipeng Hu, Wenjin Deng, Changjie Fan, Tangjie Lv, and Yu Ding. Dinet: Deformation inpainting network for realistic face visually dubbing on high resolution video. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 3543–3551, 2023. 22
- [98] Zijian Zhang, Zhou Zhao, and Zhijie Lin. Unsupervised representation learning from pre-trained diffusion probabilistic models. *Advances in neural information processing systems*, 35:22117–22130, 2022. 13

- [99] Zijian Zhang, Zhou Zhao, Jun Yu, and Qi Tian. Shiftddpms: exploring conditional diffusion models by shifting diffusion trajectories. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 3552–3560, 2023. [13](#)
- [100] Rui Zhao, Yuchao Gu, Jay Zhangjie Wu, David Junhao Zhang, Jiawei Liu, Weijia Wu, Jussi Keppo, and Mike Zheng Shou. Motiondirector: Motion customization of text-to-video diffusion models. *arXiv preprint arXiv:2309.13560*, 2023. [27](#)
- [101] Xuanlei Zhao, Xiaolong Jin, Kai Wang, and Yang You. Real-time video generation with pyramid attention broadcast. *arXiv preprint arXiv:2408.12588*, 2024. [13](#)
- [102] Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-sora: Democratizing efficient video production for all, March 2024. [6](#), [27](#)
- [103] Daquan Zhou, Weimin Wang, Hanshu Yan, Weiwei Lv, Yizhe Zhu, and Jiashi Feng. Mag-icvideo: Efficient video generation with latent diffusion models. *arXiv preprint arXiv:2310.11545*, 2023. [27](#)
- [104] Yuan Zhou, Qiuyue Wang, Yuxuan Cai, and Huan Yang. Allegro: Open the black box of commercial-level video generation model. *arXiv preprint arXiv:2410.15458*, 2024. [6](#), [27](#)