

# WAN: OPEN AND ADVANCED LARGE-SCALE VIDEO GENERATIVE MODELS

**Wan Team, Alibaba Group**

## ABSTRACT

This report presents Wan, a comprehensive and open suite of video foundation models designed to push the boundaries of video generation. Built upon the mainstream diffusion transformer paradigm, Wan achieves significant advancements in generative capabilities through a series of innovations, including our novel spatio-temporal variational autoencoder (VAE), scalable pre-training strategies, large-scale data curation, and automated evaluation metrics. These contributions collectively enhance the model’s performance and versatility. Specifically, Wan is characterized by four key features: *Leading Performance*: The 14B model of Wan, trained on a vast dataset comprising billions of images and videos, demonstrates the scaling laws of video generation with respect to both data and model size. It consistently outperforms the existing open-source models as well as state-of-the-art commercial solutions across multiple internal and external benchmarks, demonstrating a clear and significant performance superiority. *Comprehensiveness*: Wan offers two capable models, *i.e.*, 1.3B and 14B parameters, for efficiency and effectiveness respectively. It also covers multiple downstream applications, including image-to-video, instruction-guided video editing, and personal video generation, encompassing up to eight tasks. Meanwhile, Wan is the first model that can generate visual text in both Chinese and English, significantly enhancing its practical value. *Consumer-Grade Efficiency*: The 1.3B model demonstrates exceptional resource efficiency, requiring only 8.19 GB VRAM, making it compatible with a wide range of consumer-grade GPUs. It also exhibits superior performance compared to larger open-source models, showcasing remarkable efficiency for text-to-video. *Openness*: We open-source the entire series of Wan, including source code and all models, with the goal of fostering the growth of the video generation community. This openness seeks to significantly expand the creative possibilities of video production in the industry and provide academia with high-quality video foundation models. In addition, we conduct extensive experimental analyses covering various aspects of the proposed Wan, presenting detailed results and insights. We believe these findings and conclusions will significantly advance video generation technology. All the code and models are available at <https://github.com/Wan-Video/Wan2.1>.

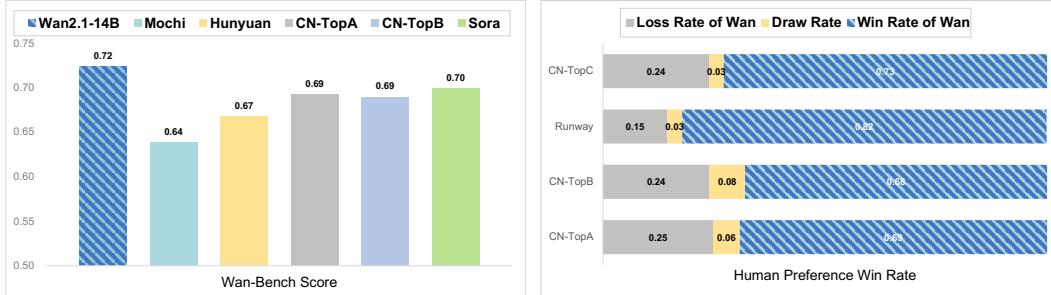


Figure 1: Comparison of Wan with state-of-the-art open-source and closed-source models. Following both benchmark and human evaluations, Wan consistently demonstrated superior results. Note that HunyuanVideo Kong et al. (2024) is tested using the open-source model.

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Related Work</b>	<b>3</b>
<b>3</b>	<b>Data Processing Pipeline</b>	<b>5</b>
3.1	Pre-training Data . . . . .	6
3.2	Post-training Data . . . . .	7
3.3	Dense Video Caption . . . . .	8
3.3.1	Open Source Dataset . . . . .	8
3.3.2	In-house Dataset . . . . .	8
3.3.3	Model Design . . . . .	9
3.3.4	Evaluation . . . . .	10
<b>4</b>	<b>Model Design and Acceleration</b>	<b>10</b>
4.1	Spatio-temporal Variational Autoencoder . . . . .	10
4.1.1	Model Design . . . . .	10
4.1.2	Training . . . . .	11
4.1.3	Efficient Inference . . . . .	11
4.1.4	Evaluation . . . . .	12
4.2	Model Training . . . . .	13
4.2.1	Video Diffusion Transformer . . . . .	14
4.2.2	Pre-training . . . . .	14
4.2.3	Post-training . . . . .	15
4.3	Model Scaling and Training Efficiency . . . . .	15
4.3.1	Workload Analysis . . . . .	15
4.3.2	Parallelism Strategy . . . . .	16
4.3.3	Memory Optimization . . . . .	17
4.3.4	Cluster Reliability . . . . .	17
4.4	Inference . . . . .	17
4.4.1	Parallel Strategy . . . . .	17
4.4.2	Diffusion Cache . . . . .	18
4.4.3	Quantization . . . . .	18
4.5	Prompt Alignment . . . . .	19
4.6	Benchmarks . . . . .	20
4.7	Evaluation . . . . .	22
4.7.1	Metrics and Results . . . . .	22
4.7.2	Ablation Study . . . . .	24
<b>5</b>	<b>Extended Applications</b>	<b>26</b>
5.1	Image-to-Video Generation . . . . .	26
5.1.1	Model Design . . . . .	26
5.1.2	Dataset . . . . .	27
5.1.3	Evaluation . . . . .	27
5.2	Unified Video Editing . . . . .	30
5.2.1	Model Design . . . . .	31
5.2.2	Datasets and Implementation . . . . .	34
5.2.3	Evaluation . . . . .	35
5.3	Text-to-Image Generation . . . . .	35
5.4	Video Personalization . . . . .	35
5.4.1	Model Design . . . . .	35
5.4.2	Dataset . . . . .	37
5.4.3	Evaluation . . . . .	37
5.5	Camera Motion Controllability . . . . .	38
5.6	Real-time Video Generation . . . . .	38
5.6.1	Method . . . . .	38
5.6.2	Streaming Video Generation . . . . .	40
5.6.3	Consistency Model Distillation . . . . .	44
5.7	Audio Generation . . . . .	44
5.7.1	Model Design . . . . .	45
5.7.2	Evaluation . . . . .	47
<b>6</b>	<b>Limitation and Conclusion</b>	<b>47</b>
<b>7</b>	<b>Contributors</b>	<b>49</b>

## 1 INTRODUCTION

Since the introduction of Sora (OpenAI, 2024) by OpenAI, video generation technology has attracted substantial attention from both industry and academia, leading to rapid advancements in the field. The emergence of models capable of generating videos that are on par with professionally crafted content has significantly improved the efficiency of content creation while simultaneously reducing the costs associated with video production. These rapid advancements in video generation technology have also been greatly attributed to the development of the open-source community. Notable projects like HunyuanVideo (Kong et al., 2024), Mochi (GenmoTeam, 2024), and CogVideoX (Yang et al., 2025b) have made their video foundation model codes and weights publicly available, gradually narrowing the gap between open-source models and commercial counterparts. However, it is essential to acknowledge the persistent gap between these outstanding open-source models and the latest closed-source models. This gap is primarily evident in three aspects: *Suboptimal Performance*: A notable performance gap remains, as the pace of development in commercial models far exceeds that of current open-source models, resulting in significantly superior capabilities. *Limited Capabilities*: Most foundational models are limited to general text-to-video (T2V) tasks, whereas the demands of video creation are multifaceted. Consequently, basic T2V models are insufficient to address these diverse requirements. *Insufficient Efficiency*: Despite their impressive performance and scale, these models often prove impractical for creative teams with limited computational resources, hindering their accessibility and usability. These challenges collectively impose constraints on the continued growth and innovation within the open-source community.

To address the aforementioned challenges, this report introduces and publicly releases a novel series of high-performance foundational video generation models, referred to as Wan, which sets a new benchmark in the field. The core design of Wan is inspired by the success of Diffusion Transformers (DiT) (Peebles & Xie, 2023) combined with Flow Matching (Lipman et al., 2022), a framework that has demonstrated the significant performance gains achievable through scaling in both text-to-image (T2I) (Esser et al., 2024) and text-to-video (T2V) (Kong et al., 2024) tasks. Within this architectural paradigm, cross-attention is employed to embed text conditions, while the model's design is meticulously optimized to ensure computational efficiency and precise text controllability. To further enhance the model's ability to capture complex dynamics, a full spatio-temporal attention mechanism is incorporated. Through extensive experimentation, the model is validated at scale, reaching 14 billion parameters. Subsequently, Wan has seen large-scale data comprising billions of images and videos, amounting to  $\mathcal{O}(1)$  trillions of tokens in total. This extensive training facilitates the emergence of the model's capabilities, allowing it to demonstrate robust performance across multiple dimensions, such as motion amplitude and quality, visual text generation, camera control, instruction adherence, and stylistic diversity. Building upon the powerful foundational model, we have expanded its capabilities to numerous downstream tasks, including image-to-video generation (I2V), instruction-guided video editing (V2V), zero-shot personalized customization, real-time video generation, and audio generation, among other critical applications. To minimize inference costs, we also introduce a 1.3B model alongside a 14B model for T2V and I2V, both of which support 480p resolution and greatly enhance inference efficiency. Remarkably, the 1.3B model requires only 8.19G of VRAM, allowing it to run on many consumer-grade GPUs, while its performance exceeds that of many larger open-source models.

Moreover, we will also publicly present the entire training process, including the large-scale data construction pipeline, video variational autoencoder (VAE), training strategies, acceleration techniques, and automated evaluation algorithms, to empower the community in developing specialized foundational video models. In addition, we will provide comprehensive design details and experimental results, offering insights into phenomena observed during the resource-intensive training of large generative models alongside our key findings and conclusions. We are confident that these contributions will play a pivotal role in accelerating the advancement of video generation technology.

## 2 RELATED WORK

Driven by advances in generative modeling, the landscape of large-scale video models has evolved significantly, particularly in diffusion-based frameworks. Our review focuses on two broad categories: closed-source models and contributions from the open-source community.



Figure 2: Samples generated by Wan. Wan can produce videos with large motions, high fidelity, and realistic details. It is also the first model to generate both Chinese and English text within videos. Additionally, Wan offers a range of features, including text-to-video, image-to-video, and video editing capabilities.

**Closed-source models.** Closed-source models are mainly those developed by major technology companies, which aims at high-quality, professional video generation due to the extensive resources invested. We organize the notable models released over the past year in chronological order. In February 2024, OpenAI introduced Sora ([OpenAI, 2024](#)) and marked a significant leap in AI-generated content. In June 2024, Kling ([Kuaishou, 2024.06](#)) by Kuaishou and Luma 1.0 ([LumaLabs, 2024.06](#)) by LUMA AI were opened up for public testing, offering powerful video generation capabilities. Meanwhile, Runway unveiled Gen-3 ([Runway, 2024.06](#)), building upon Gen-2 ([Runway, 2023](#)), to further elevate the standards in video creation. In July 2024, Shengshu AI released Vidu ([Bao et al., 2024](#)) equipped with self-designed U-ViT ([Bao et al., 2023](#)) architecture. In September 2024, both Kling and Luma undergone updates to version 1.5. During the same period, MiniMax introduced

Hailuo Video ([MiniMax, 2024.09](#)), delivering impressive visual results to the public. In October 2024, Pika Labs launched Pika 1.5 ([PikaLabs, 2024.10](#)), enabling users to customize visual and physical attributes in videos. In addition, Meta introduced Movie Gen ([Polyak et al.](#)), a series of video foundation models that detailed their training processes and applications. In December 2024, Kling advanced to version 1.6 and, concurrently, Google released Veo 2 ([DeepMind, 2024.12](#)) with an improved understanding of real-world physics and human movement nuances.

These developments underscore the intense global competition in the video generation sector. In this context, our open-source Wan has shown competitive or even superior performance over these commercial models across both internal and external benchmarks, leading in multiple attributes.

**Contributions from open-source community.** On the other hand, the open-source community has made substantial contributions not only to the holistic video generative models but also to the exploration of essential model components. Diffusion-based video generation models typically build on Stable Diffusion ([Rombach et al., 2022](#)) architecture. It mainly involves three critical modules: an autoencoder to map the original video into a compact latent space, a text encoder to extract text embeddings, and a neural network optimized via the diffusion model to learn the distribution of these video latents. For network structure, U-Net ([Ronneberger et al., 2015](#)), initially used in image generation ([Ho et al., 2020; Song et al., 2020](#)), is adapted for video generation by incorporating temporal dimensions. VDM ([Ho et al., 2022](#)) extends the 2D U-Net to a 3D version, while another paradigm ([Zhou et al., 2022; Wang et al., 2023a; Guo et al., 2024b](#)) introduces 1D temporal attention combined with 2D spatial attention block to reduce computation costs. Notably, Diffusion Transformers (DiT ([Peebles & Xie, 2023](#))), which rely solely on transformer blocks, are superior to U-Net in visual generation tasks ([Chen et al., 2023a](#)). This structure has also been transferred to video models ([Ma et al., 2024](#)), giving rise to two common variants: the original DiT ([Peebles & Xie, 2023; HaCohen et al., 2024](#)) that uses cross-attention for text embeddings and MM-DiT ([GenmoTeam, 2024; Kong et al., 2024](#)), where text embeddings are concatenated to visual embeddings for full-attention processing. Regarding autoencoders, while the early approach ([Rombach et al., 2022](#)) adopts standard VAE ([Kingma, 2013](#)), recent autoencoders like VQ-VAE ([Van Den Oord et al., 2017](#)) and VQGAN ([Esser et al., 2021](#)) improve model design for better reconstruction and compression. LTX-Video ([HaCohen et al., 2024](#)) modifies the VAE decoder to perform the final denoising step and convert latents into pixels, where the missing high-frequency details are generated during decoding. The text encoder is also critical for text-based video generation. Current powerful video generation models primarily utilize the T5 series ([Raffel et al., 2020](#)) as the main text encoder, often in conjunction with CLIP ([Radford et al., 2021](#)). In the case of HunyuanVideo ([Kong et al., 2024](#)), T5 is replaced with a Multimodal Large Language Model ([Liu et al., 2023c; Li et al., 2023](#)) to achieve more robust alignment between text embeddings and visual features.

By integrating these key modules with effective diffusion-based optimization techniques ([Ho et al., 2020; Song & Ermon, 2019; Lipman et al., 2022](#)), multiple promising open-source video generation models have emerged ([GenmoTeam, 2024; Kong et al., 2024; HaCohen et al., 2024; Zheng et al., 2024; Lin et al., 2024; Jin et al., 2024; Yang et al., 2025b](#)). In Wan, we have carefully designed or selected each critical component to ensure high-quality video synthesis. We provide detailed design information and ablation study, which can facilitate the design of future video generation models.

In addition, numerous studies have explored downstream tasks in video generation. These tasks include repainting ([AI, 2022; Zhou et al., 2023](#)), editing ([Meng et al., 2021; Brooks et al., 2023; Zhang et al., 2023a; Wang et al., 2023b; Wei et al., 2024b](#)), controllable generation ([Zhang et al., 2023b; Jiang et al., 2024; Wang et al., 2024c](#)), and frame reference generation ([Yang et al., 2025b; Guo et al., 2024a](#)), often by leveraging adapter-based and ControlNet-like architectures ([Zhang et al., 2023b; Chu et al., 2023](#)) to incorporate user-specified conditions. We also develop various downstream applications based on Wan, which have demonstrated excellent performance.

### 3 DATA PROCESSING PIPELINE

High-quality data is essential for training large generative models, and an automated data construction pipeline significantly enhances the efficiency of the training process. In developing our dataset, we prioritized three core principles: *high quality*, *high diversity*, and *substantial scale*. Following these principles, we curated a dataset comprising billions of videos and images. This section offers a detailed introduction to the data construction pipeline employed for Wan.

### 3.1 PRE-TRAINING DATA

We curated and deduplicated a candidate dataset sourced from both internal copyrighted sources and publicly accessible data. In the pre-training stage, our goal is to select high-quality and diverse data from this expansive yet noisy dataset to facilitate effective training. Throughout the data mining process, we designed a four-step data cleaning process, focusing on fundamental dimensions, visual quality, and motion quality. Subsequently, we will also highlight our data processing workflow for constructing visual text data.

**Fundamental dimensions.** The fundamental dimensions of our data filtering framework focus on the intrinsic attributes of the source video and image data, enabling efficient preliminary filtering out of all the unsuitable data. Specifically, our multidimensional filtering approach encompasses the following critical aspects: *Text detection*. A lightweight OCR detector is implemented to quantify text coverage ratios, effectively excluding videos and images with excessive textual elements to maintain visual clarity. *Aesthetic evaluation*: We use the widely adopted LAION-5B ([Schuhmann et al., 2021](#)) aesthetic classifier to perform an initial quality assessment of our images, quickly filtering out low-quality data. *NSFW Score*. Through our internal safety assessment model, we systematically evaluate and filter inappropriate content based on computed NSFW scores in all training data. *Watermark and logo detection*. We detect whether the video or images contain watermarks and logos, and crop these elements during training. *Black border detection*. Utilizing heuristic-based detection methods, we automatically crop extraneous black borders to maintain focus on content-rich regions. *Overexposure detection*. Our trained expert classifier evaluates and filters out data with abnormal tonal distributions, ensuring optimal visual quality in the training dataset. *Synthetic image detection*. Empirical evidence indicates that even minimal contamination (< 10%) by generated images can significantly degrade the performance of the model. Therefore, we train an expert classifier to filter out these “contaminating” images. *Blur detection*. An internally developed model assigns quantitative blur scores to training materials, enabling the systematic removal of visually indistinct content. *Duration and resolution*. We also enforce constraints where video duration must exceed 4 seconds, and resolution thresholds are applied at different training stages to filter out low-quality data. Through the implementation of these efficient preprocessing strategies, we successfully eliminated approximately 50% of the initial dataset. The retained high-quality data subsequently progresses to a more superior semantic-driven selection stage for further refinement.

**Visual quality.** This primarily involves selecting data from the candidate dataset that is of relatively high quality and meets pre-training standards. This subset of data must have good visual quality and its overall distribution should align with that of natural data. In this process, we divide the task into two parts: *clustering* and *scoring*. *Clustering* is used to split all the data into smaller subsets, allowing us to process each subset individually. The benefit of this approach is that it prevents the loss of small yet important data segments that might occur due to a long-tail distribution, thereby maintaining the original data distribution. Specifically, we divide the data into 100 clusters and then select a certain amount of data from each cluster for the next stage of data processing. *Scoring* is used to assign a quality score to each video or image, facilitating the selection and processing of data at different stages. Specifically, we select samples from each cluster for manual scoring (ranging from 1 to 5, with 1 being the worst and 5 being the best). We then train an expert assessment model using all the annotated data to score the entire dataset.

**Motion quality.** The goal of motion quality assessment is to select videos that are natural, complete, and with significant motion, while avoiding static or jittery movements. We classify video data into six distinct motion quality tiers. *Optimal motion*: This level represents videos with optimal attributes, such as significant motion layout, perspective, and amplitude, along with clean, smooth movement. *Medium-quality motion*: Videos in this category exhibit noticeable movements but may have minor issues like multiple subjects or partial occlusion. This data ensures motion diversity and can be used during pre-training to help the model better understand spatio-temporal relationships. *Static Videos*: This category primarily focuses on the videos, which consist largely of chat and interview-style videos. While these videos contain minimal motion information, they are of high quality. Therefore, we need to identify them separately and reduce their sampling ratio. *Camera-driven Motion*: Footage dominated by camera movement (*e.g.*, aerial shots) with minimal subject motion. Given their similarity to static images, these receive substantially lower sampling priority. *Low-quality Motion*: Videos exhibiting excessive subjects, severe occlusions, or unclear main subjects (*e.g.*, crowded street scenes). Such content is excluded due to its negative impact on training efficiency and motion

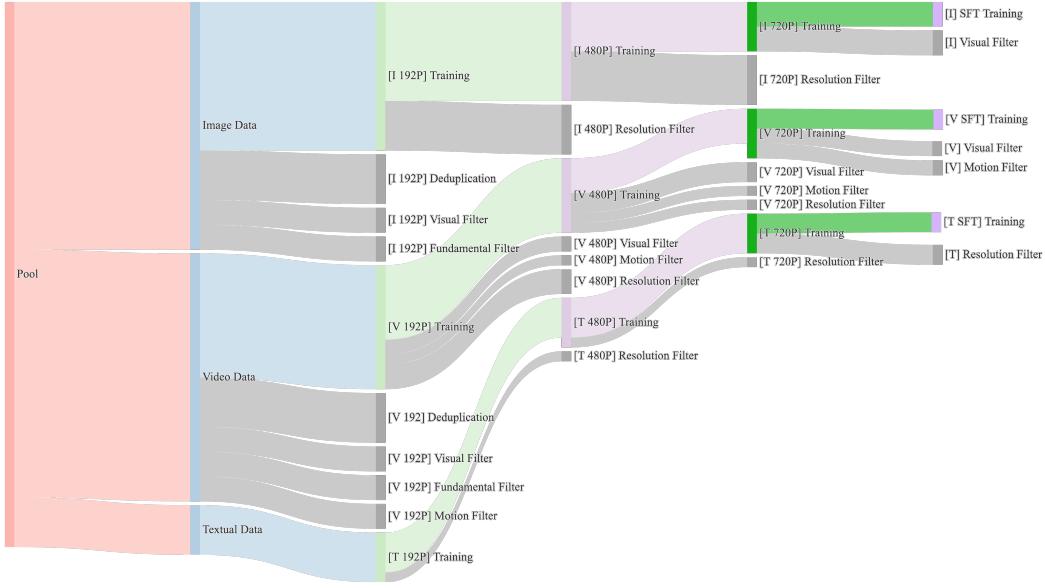


Figure 3: Data provisioning across different training phases. For each stage, we dynamically adjust the proportions of data related to motion, quality, and category based on data throughput.

generation quality. *Shaky camera footage*: Amateur recordings with pronounced camera shake, often causing motion blur and ambiguous foreground-background differentiation. This category is systematically excluded from training consideration.

**Visual text data.** In addition, we also introduce a novel data processing approach to enhance the visual text generation of *Wan*. This approach comprises two distinct processing branches designed to improve both the accuracy and the harmony of text rendering. On one hand, we synthesize hundreds of millions of text-containing images by rendering Chinese characters on a pure white background. On the other hand, we collect large amounts of text-containing images from vast real-world data. We employ multiple OCR models to accurately recognize both English and Chinese text from images and videos, and subsequently input these extracted text contents into the multimodal language model Qwen2-VL. This process generates natural descriptions of the images, ensuring that the descriptions incorporate as much precise text content as possible. We collect a mass of real-world image-text pairs by leveraging this processing pipeline. By integrating both synthetic and real data during the pre-training stage, our approach effectively generalizes to generate rare words in videos with accurate glyphs and high realism, demonstrating significant superiority in visual text generation.

### 3.2 POST-TRAINING DATA

The core objective of post-training is to improve both the visual fidelity and motion dynamics of generated videos through high-quality data. Our data processing pipeline in this stage employs distinct strategies for static and dynamic data: image data undergoes optimization for improved visual quality, while video data is specifically processed to refine motion quality.

**Image processing.** From the pool of high-scoring image data, we perform an additional refinement process to identify top-quality samples that the images excel in aspects such as quality, composition and details. Specifically, we construct the curated dataset using two methods: *expert-based collection* and *manual collection*. The former method involves selecting the top 20% of images based on scores predicted by an expert model. For this subset, we also consider factors such as style and category to ensure diversity in data distribution. The latter approach involves manually collecting top-quality images from various categories and data sources while also filling in any missing concepts in our dataset to enhance the model's generalization capabilities. Through this process, we collected a total of millions of curated images.

**Video processing.** In this phase, we employ a strategy similar to the one used for images to collect top-quality videos. First, we filter out some top-ranked videos from the candidate datasets using the visual quality classifier. Then, based on the motion quality classifier, we select millions of videos featuring simple movements and millions of videos with complex movements. All video selections follow a strategy that emphasizes category balance and high diversity. Meanwhile, we select data from 12 major categories, including technology, animals, arts, humans, vehicles, and *etc*, in order to enhance the model’s generative capabilities for commonly used categories.

### 3.3 DENSE VIDEO CAPTION

While our dataset includes text descriptions from original webpages for many images and videos, they are often too simplistic to convey detailed visual content. DALL-E 3 (Betker et al., 2023) has demonstrated that the ability of visual generation models to follow prompts can be significantly enhanced by training on highly descriptive generated visual captions. Therefore, we develop an internal caption model to generate dense captions for each image and video of our dataset. To train this model, we incorporate both open-source vision-language datasets and additional data collected in-house.

#### 3.3.1 OPEN SOURCE DATASET

We collect widely used vision-language datasets for both images and videos. Our collection includes not only caption datasets but also visual Q&A datasets focused on visual content, such as actions, counting, and OCR. In some scenarios, we require the caption model to generate captions in a specific style or content based on user instructions. Therefore, we also collected pure text instruction data to enhance our model’s ability to follow instructions.

#### 3.3.2 IN-HOUSE DATASET

We collect an in-house dataset for various tasks to enhance the model’s capabilities in specific areas.

*Celebrities, landmarks, and movie characters.* To train our model to recognize celebrities, landmarks, and characters, we collect dataset comprising thousands of identities. First, we use large language models (LLMs) to collect names, then retrieve corresponding images from our image-text database using a CLIP-style model. We test various CLIP-style models and find that TEAM (Xie et al., 2022) excels in recognizing individuals, particularly Chinese celebrities. To further reduce noise in our dataset, we implemented keyword matching for the descriptions of the image-text pairs. An image is retained only if it is retrieved by the TEAM model and matches the specified keywords.

*Object counting.* To enhance our model’s visual counting ability, we compiled a counting dataset by retrieving images with text descriptions that include words like “one,” “two,” “three”, etc. These numbers provide coarse annotations for the images. We then use a LLM to extract (*category, count*) pairs from the corresponding text. Finally, we use Grounding DINO (Liu et al., 2023d) to count objects in those categories. An image is retained only if its text description matches the results from Grounding DINO (Liu et al., 2023d).

*OCR.* To improve our model’s capability in generating text descriptions, we collect an OCR-augmented image caption dataset. Initially, we used an off-the-shelf OCR detector to extract text from images. Then, we prompt our caption model to generate descriptions based on these OCR results. This approach ensures accurate text descriptions by using the extracted OCR text as prior knowledge. Currently, we include only English and Chinese texts.

*Camera angle and motion.* We observe that current multi-modal language models (MLLMs) struggle with predicting camera angles and motion, including state-of-the-art commercial models like GPT-4o and Google Gemini Pro. To enhance these predictions, we first annotate a set of videos for camera angle and motion. This dataset can be utilized in two ways: directly training our caption model used in the final training stage, or by training an expert model to annotate more video data, aiding our model during the early training phase. This dataset enhances the accuracy of our motion and angle descriptions, thereby improving the camera motion controllability of our video generation model.

*Fine-grained categories.* To enhance our model’s ability to recognize fine-grained categories such as animals, plants, and vehicles, we created a dataset containing several million images across these categories.

*Relational understanding.* We enhance our model’s relational understanding by gathering dataset that focus on spatial relationships, such as left, right, up, and down, through the incorporation of existing object detection datasets.

*Re-caption.* An important capability of a caption model is to leverage existing tags or brief captions to generate dense captions. To facilitate this, we curate a re-caption dataset that extends brief text tags or captions to detailed descriptions based on image content.

*Editing instruction caption.* We collect a dataset that describes the differences or transformations between two images. This type of caption is valuable for image editing tasks.

*Group image description.* We collect a dataset that provides descriptions for groups of images. Each caption begins with a description of the common features shared by the images, followed by individual descriptions of each image, using similar phrasing.

*Human-annotated image and video captions.* We gather human-annotated dense caption for both images and videos. These represent our highest-quality caption data and are used in the final stage of model training.

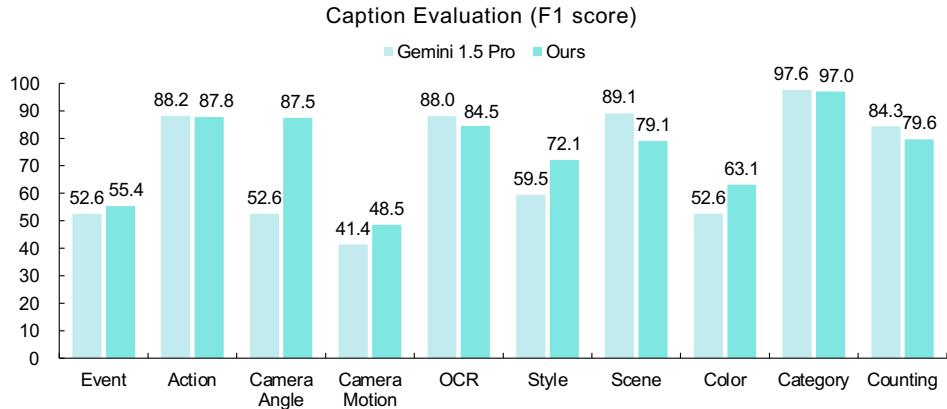


Figure 4: Our caption model achieves overall performance comparable to Google Gemini 1.5 Pro. In particular, it excels in areas such as events, camera angle, camera motion, style, and color.

### 3.3.3 MODEL DESIGN

**Architecture.** Our caption model adopts a LLaVA-style architecture (Liu et al., 2023c). We use a ViT encoder to extract visual embeddings of images and video frames. These embeddings are projected through a two-layer perception and then fed into the Qwen LLM (QwenTeam, 2024).

For the image input, we adopt dynamic high resolution as in LLaVA. An image can be divided into a maximum of seven patches. For each patch, we adaptively pool the vision embeddings to a  $12 \times 12$  grid representation to reduce computation.

For the video, we sample 3 frames per second, with an upper limit of 129 frames. To further reduce computation, we employ a slow-fast encoding strategy: maintaining the original resolution for every fourth frame and applying global average pooling to the embeddings of the remaining frames. Experiments on long video benchmarks indicate that the slow-fast mechanism enhances understanding with a limited number of visual tokens (*e.g.*, improving performance from 67.6% to 69.1% on VideoMME (Fu et al., 2024)) without subtitles.

**Training.** Following recent state-of-the-art methods, our training process is divided into three stages. In the first stage, we freeze the ViT and LLM, and train only the multi-layer perceptron to align the visual embeddings with the LLM input space, using a learning rate of 1e-3. In the second stage, all parameters are made trainable. In the final stage, we conduct end-to-end training on a small set of

high-quality data. For the last two stages, the learning rates are set to 1e-5 for the LLM and MLP, and 1e-6 for the ViT.

### 3.3.4 EVALUATION

Automatic evaluation is crucial for the development of our caption model, allowing us to identify the strengths and weaknesses of each model version. Following CAPability (Liu et al., 2025b), we develop an automatic caption evaluation pipeline. We focus on ten key visual dimensions in video generation: action, camera angle, camera motion, object category, object color, object counting, OCR, scene, style, and event. For each dimension, we randomly sampled 1,000 videos and their corresponding captions generated by our model. We also used Google Gemini 1.5 Pro to generate dense video captions for this dataset. We evaluate our captions and Gemini-generated captions by calculating the F1 metrics for each dimension following CAPability (Liu et al., 2025b). The results are illustrated in Fig. 4. Our caption performs better in video event, camera angle, camera motion, style, and object color, while Gemini excels in action, OCR, scene, object category, and object counting.

## 4 MODEL DESIGN AND ACCELERATION

### 4.1 SPATIO-TEMPORAL VARIATIONAL AUTOENCODER

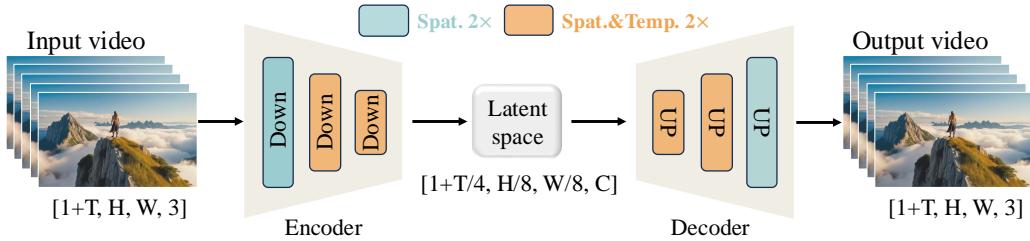


Figure 5: Our Wan-VAE Framework. Wan-VAE can compress the spatio-temporal dimension of a video by  $4 \times 8 \times 8$  times. The orange rectangles represent  $2 \times$  spatio-temporal compression, and the green rectangles represent  $2 \times$  spatial compression.

Variational Autoencoders (VAEs) (Wu et al., 2024; Blattmann et al., 2023; OpenAI, 2024) play a crucial role in learning compact latent representations from high-dimensional visual data (especially videos), facilitating scalable and efficient training of generative models, e.g., diffusion models. However, designing effective VAEs for video generation tasks faces several challenges. First, videos inherently possess both spatial and temporal dimensions, requiring the VAE to capture complex spatio-temporal dependencies. Second, the inherent high-dimensional nature of video (e.g., multiple frames with high-resolution pixels) increases memory consumption and computational costs, making it difficult to scale VAEs to long video sequences. Third, ensuring temporal causality (*i.e.*, future frames do not influence past frames) is critical for generating realistic and coherent video content, yet this constraint introduces additional architectural complexities.

To address these challenges, we propose a novel 3D causal VAE architecture (Wan-VAE) specifically designed for video generation. We combine multiple strategies (Wu et al., 2024) to improve spatio-temporal compression, reduce memory usage, and ensure temporal causality. These enhancements make Wan-VAE more efficient and scalable and better suited for integration with diffusion-based generative models such as DiT. Next, we introduce the network design, training details, efficient inference, and experimental comparison of our Wan-VAE.

#### 4.1.1 MODEL DESIGN

To achieve a bidirectional mapping between the high-dimensional pixel space and the low-dimensional latent space, we design a 3D causal VAE as illustrated in Fig. 5. Given an input video  $V \in \mathbb{R}^{(1+T) \times H \times W \times 3}$ , Wan-VAE compresses its spatio-temporal dimensions to  $[1 + T/4, H/8, W/8]$  while expanding the number of channels  $C$  to 16. Specifically, the first frame is only spatially

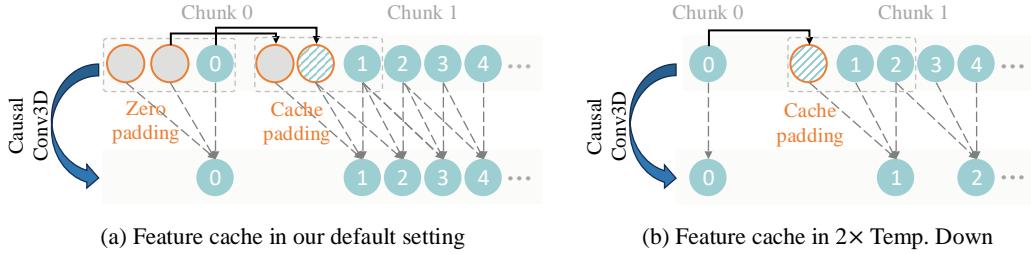


Figure 6: Our feature cache mechanism. (a) and (b) show how we use this mechanism in regular causal convolution and temporal downsampling, respectively.

compressed to better handle the image data, following MagViT-v2 (Yu et al., 2023). In terms of architecture, we replace all GroupNorm layers (Wu & He, 2018) with RMSNorm layers (Zhang & Sennrich, 2019) to preserve temporal causality. This modification enables the use of the feature cache mechanism (refer to Sec. 4.1.3), significantly improving the inference efficiency. Furthermore, we halve the input feature channel in the spatial upsampling layer, resulting in a 33% reduction in memory consumption during inference.

By carefully tuning the number of base channels further, Wan-VAE achieves a compact model size of only 127M parameters. This optimization reduces the encoding time and memory usage, thus benefiting the training of subsequent diffusion transformer models.

#### 4.1.2 TRAINING

We adopt a three-stage approach to train Wan-VAE. First, we construct a 2D image VAE with the same structure and train it on image data. Then, we inflate the well-trained 2D image VAE into a 3D causal Wan-VAE to provide an initial spatial compression prior, which drastically improves the training speed compared to training a video VAE from scratch. At this stage, Wan-VAE is trained on the low-resolution ( $128 \times 128$ ) and small frame number (5-frame) videos to accelerate convergence speed. The training losses include  $L1$  reconstruction loss, KL loss, and LPIPS perceptual loss, which are weighted by coefficients of 3, 3e-6, and 3, respectively. Finally, we fine-tune the model on high-quality videos with different resolutions and frame numbers, integrating the GAN loss (Esser et al., 2021) from a 3D discriminator.

#### 4.1.3 EFFICIENT INFERENCE

To efficiently support the encoding and decoding of arbitrarily long videos (Yang et al., 2025a; Yao et al., 2021), we implement a feature cache mechanism within the causal convolution module of Wan-VAE.

Specifically, the number of video sequence frames follows the  $1 + T$  input format, so we divide the video into  $1 + T/4$  chunks, which is consistent with the number of latent features. When processing input video sequences, the model employs a chunk-wise strategy where each encoding and decoding operation handles only the video chunk corresponding to a single latent representation. Based on the temporal compression ratio, the number of frames in each processing chunk is limited to at most 4, effectively preventing memory overflow.

To ensure temporal continuity between context chunks, our model strategically maintains frame-level feature caches from preceding chunks. These cached features are systematically integrated into the causal convolution computations of subsequent chunks. The specific process is illustrated in Fig. 6, which demonstrates two typical scenarios in our feature cache mechanism. In Fig. 6 (a), in our default setting, the causal convolution does not change the number of frames, and we need to maintain two cached features (convolution kernel size is 3) from historical frames. For the initial chunk, we apply zero padding with two dummy frames to initialize the cache buffer. Subsequent chunks systematically reuse the last two frames from the preceding chunk as cached features while discarding obsolete historical data. Fig. 6 (b) presents the scenario involving  $2 \times$  temporal downsampling (stride is equal to 2). Here, the scenario necessitates different cache management: we implement single-frame cache

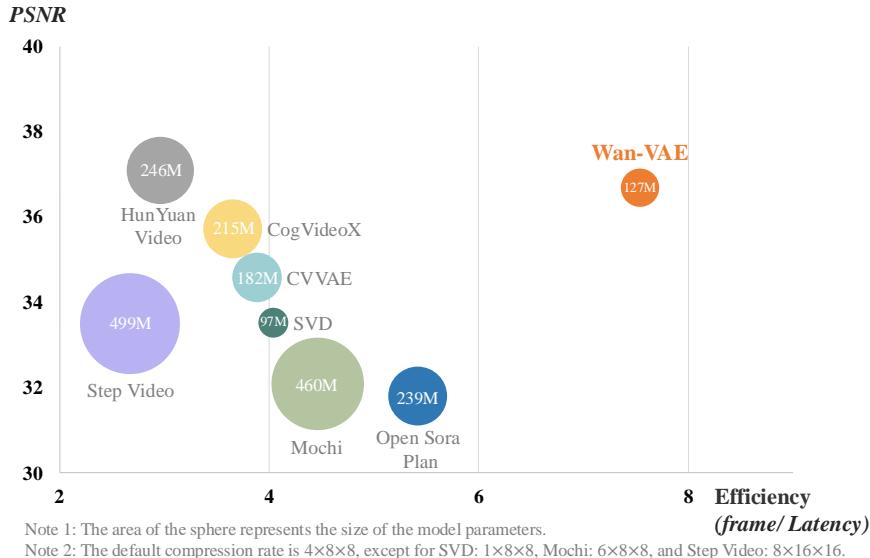


Figure 7: Comparison of video reconstruction performance at  $720 \times 720$  resolution and 25 frames.

padding exclusively for non-initial chunks to ensure dimensional consistency. This configuration guarantees that the output sequence length follows the exact downsampling ratio while maintaining causal relationships across the boundaries of the chunk.

This feature cache mechanism not only optimizes memory utilization but also preserves feature coherence across chunk boundaries, thereby supporting stable inference for infinite-length videos.

#### 4.1.4 EVALUATION

**Quantitative results.** This study conducts a comprehensive evaluation of the performance of various state-of-the-art (SOTA) video VAE models (Kong et al., 2024; Yang et al., 2025b) by analyzing their Peak Signal-to-Noise Ratio (PSNR) and processing efficiency (frame per unit latency (second)). Notably, for fair comparison, most models (Kong et al., 2024; Yang et al., 2025b) adopt the same compression rate and latent feature dimension as our approach, *i.e.*, a compression rate of  $4 \times 8 \times 8$  with a latent dimension of 16. Open Sora Plan (Lab & etc., 2024) has the same compression ratio as ours, but the latent dimension is 4. SVD (Blattmann et al., 2023) employs a compression rate of  $1 \times 8 \times 8$  with a latent dimension of 4, Step Video (Ma et al., 2025) employs a compression rate of  $8 \times 16 \times 16$  with a latent dimension of 64, and Mochi (GenmoTeam, 2024) uses a compression rate of  $6 \times 8 \times 8$  with a latent dimension of 12. The evaluation is performed on 200 videos. These videos consist of 25 frames, and the resolution is set to  $720 \times 720$ . In Fig. 7, the size of the circles is positively correlated with the number of model parameters.

Experimental results indicate that Wan-VAE exhibits highly competitive performance across both metrics, showcasing a compelling dual advantage of superior video quality and high processing efficiency. It is worth noting that under the same hardware environment, our VAE’s reconstruction speed is 2.5 times faster than the existing SOTA method (*i.e.*, HunYuan Video). This speed advantage will be further demonstrated at higher resolutions due to the small size design of our VAE model and the feature cache mechanism.

This advancement offers an efficient solution for video reconstruction tasks and video generation training. These findings not only validate the effectiveness of the proposed model design but also provide valuable insights for the future development of VAE technologies.

**Qualitative results.** To validate the video reconstruction performance of our method across diverse scenarios, Fig. 8 presents visual results for texture, face, text, and high-motion scenes. Compared to existing VAE models, Wan-VAE demonstrates superior performance in these contexts. In the texture scene shown in the first row, our method is capable of capturing details more accurately, such as the texture and direction of hair. In the facial scene depicted in the second row, Wan-VAE preserves facial

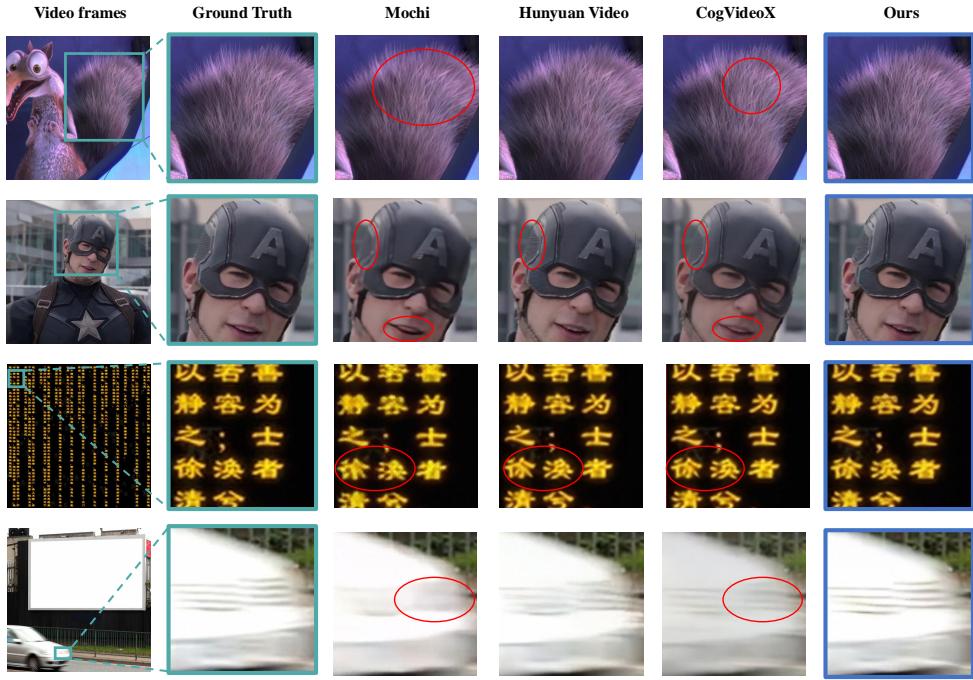


Figure 8: Visualization results of video reconstruction across different scenarios, including texture (first row), face (second row), text (third row), and high-motion (fourth row). The above videos have rich details but are not used in training.

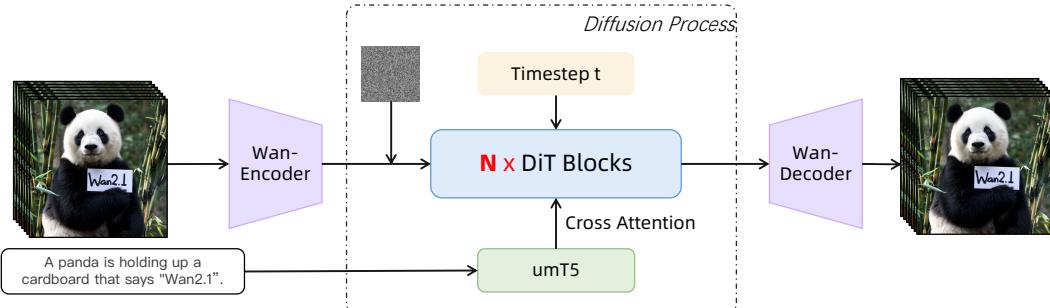


Figure 9: Architecture of the Wan.

features while reducing blurring and distortion around the lips. In the text scene illustrated in the third row, our method successfully restores textual content with clarity, mitigating issues of character distortion and loss. In the high-motion scene presented in the fourth row, Wan-VAE maintains the motion sharpness of video frames. These results indicate that our method offers significant advantages in handling a variety of complex scenarios.

#### 4.2 MODEL TRAINING

In this section, we will provide a detailed introduction to the architecture design of the foundational video model, specifically for the text-to-video task, along with the details of the pre-training and post-training phases.

In Fig. 9, we present the architecture of Wan, which is designed based on the current mainstream DiT (Peebles & Xie, 2023) architecture, which typically consists of three primary components: Wan-VAE, diffusion transformer, and text encoder. For a given video  $V \in \mathbb{R}^{(1+T) \times H \times W \times 3}$ , the

encoder of Wan-VAE encodes it from pixel space into the latent space  $x \in \mathbb{R}^{1+T/4 \times H/8 \times W/8}$ , which is then fed into the subsequent diffusion transformer structure.

#### 4.2.1 VIDEO DIFFUSION TRANSFORMER

**Diffusion transformer.** The diffusion transformer mainly consists of three components: a patchifying module, transformer blocks, and an unpatchifying module. Within each block, we focus on effectively modeling spatio-temporal contextual relationships and embedding text conditions alongside time steps. In the patchifying module, we use a 3D convolution with a kernel size of  $(1, 2, 2)$  and apply a flattening operation to convert  $x$  into a sequence of features with the shape of  $(B, L, D)$ , where  $B$  denotes the batch size,  $L = (1 + T/4) \times H/16 \times W/16$  represents the sequence length, and  $D$  indicates the latent dimension. Specifically, as illustrated in Fig. 10, we employ the cross-attention mechanism to embed the input text conditions, which can ensure the model’s ability to follow instructions even under long-context modeling. Additionally, we employ an MLP with a Linear layer and a SiLU ([Elfwing et al., 2018](#)) layer to process the input time embeddings and predict six modulation parameters individually. This MLP is shared across all transformer blocks, with each block learning a distinct set of biases. Through extensive experiments, we have demonstrated that this design can reduce 25% and reveal a significant performance improvement vs scale.

**Text encoder.** Wan’s architecture uses the umT5 (Chung et al., 2023) to encode input text. Through extensive experiments, we find that umT5 has several advantages in our framework: 1) It possesses strong multilingual encoding capabilities, allowing it to understand both Chinese and English effectively, as well as input visual text; 2) Under the same conditions, we find that umT5 outperforms other unidirectional attention mechanism LLMs in composition; 3) It exhibits superior convergence, with umT5 achieving faster convergence at the same parameter scale. Based on these findings, we ultimately chose umT5 as our text embedder.

### 4.2.2 PRE-TRAINING

We leverage the flow matching framework (Lipman et al., 2022; Esser et al., 2024) to model a unified denoising diffusion process across both image and video domains. We first conduct pre-training on low-resolution images, followed by multi-stage joint optimization of images and videos. The image-video joint training evolves using progressively upscaled data resolutions and extended temporal durations throughout the stages.

**Training objective.** Flow matching provides a theoretically grounded framework for learning continuous-time generative processes in diffusion models. It circumvents iterative velocity prediction, enabling stable training via ordinary differential equations (ODEs) while maintaining equivalence to maximum likelihood objectives. During training, given an image or video latent  $x_1$ , a random noise  $x_0 \sim \mathcal{N}(0, I)$ , and a timestep  $t \in [0, 1]$  sampled from a logit-normal distribution, an intermediate latent  $x_t$  is obtained as the training input. Following Rectified Flows (RFs) (Esser et al., 2024),  $x_t$  is defined as a linear interpolation between  $x_0$  and  $x_1$ , i.e.,

$$x_t = tx_1 + (1-t)x_0. \quad (1)$$

The ground truth velocity  $v_t$  is

$$v_t = \frac{dx_t}{dt} = x_1 - x_0. \quad (2)$$

The model is trained to predict the velocity, thus, the loss function can be formulated as the mean squared error (MSE) between the model output and  $v_t$ ,

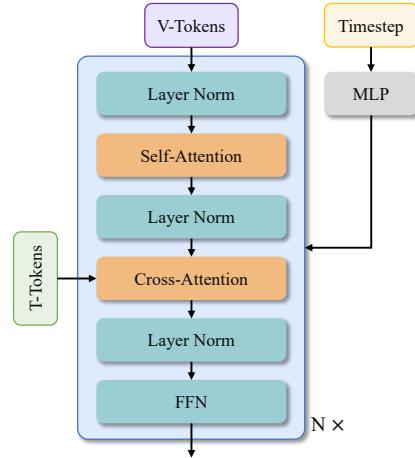


Figure 10: Transformer block of Wan.

$$\mathcal{L} = \mathbb{E}_{x_0, x_1, c_{txt}, t} \|u(x_t, c_{txt}, t; \theta) - v_t\|^2, \quad (3)$$

where  $c_{txt}$  is the umT5 text embedding sequence of 512 tokens long,  $\theta$  is the model weights, and  $u(x_t, c_{txt}, t; \theta)$  denotes the output velocity predicted by the model.

**Image pre-training.** Our experimental analysis reveals two critical challenges in direct joint training with high-resolution images and long-duration video sequences: (1) Extended sequence lengths (typically 81 frames for  $1280 \times 720$  video) substantially reduce training throughput. Under fixed GPU-hour budgets, this results in insufficient data throughput, thereby impeding model convergence rates. (2) Excessive GPU memory consumption forces suboptimal batch sizes, inducing training instability caused by spikes in gradient variance. To mitigate these issues, we initialize the 14B model training through low-resolution (256 px) text-to-image pre-training, enforcing cross-modal semantic-textual alignment and geometric structure fidelity before progressively introducing high-res video modalities.

**Image-video joint training.** Following large-scale 256 px text-to-image pre-training, we implement staged joint-training with image and video data through a resolution-progressive curriculum. The training protocol comprises three distinct stages differentiated by spatial resolution areas: (1) In the first stage, we conduct joint training with 256 px resolution images and 5-second video clips (192 px resolution, 16 fps). (2) In the second stage, we initiate spatial resolution scaling by upgrading both image and video resolutions to 480px while maintaining a fixed 5-second video duration. (3) In the final phase, we escalate spatial resolution to 720px for both images and 5-second video clips.

**Pre-training configurations.** We employ efficient training at bf16-mixed precision combined with the AdamW (Loshchilov & Hutter, 2017; Kingma & Ba, 2014) optimizer with a weight decay coefficient of  $1e^{-3}$ . The initial learning rate is set to  $1e^{-4}$ , and dynamically reduced triggered by plateaus of the FID and CLIP Score metrics.

#### 4.2.3 POST-TRAINING

In the post-training stage, we maintain the same model architecture and optimizer configuration from the pre-training stage, initializing the network with the pre-trained checkpoint. We conduct joint training at resolutions of 480px and 720px using the post-training video dataset detailed in Sec. 3.2.

### 4.3 MODEL SCALING AND TRAINING EFFICIENCY

#### 4.3.1 WORKLOAD ANALYSIS

In Wan, the computational costs associated with the text encoder and VAE encoder are lower than those of the DiT model, which accounts for more than 85% of the overall computation during training. For the DiT model, the main computational cost is given by the expression  $L(\alpha bsh^2 + \beta bs^2h)$ , where  $L$  denotes the number of DiT layers,  $b$  is the micro batch size,  $s$  indicates the sequence length, and  $h$  denotes the hidden dimension size. Here,  $\alpha$  corresponds to the cost of the linear layers and  $\beta$  to that of the attention layer, for the non-causal attention used in Wan,  $\beta$  is 4 for the forward pass and 8 for the backward pass. Unlike general LLM models, the sequence length  $s$  processed by Wan often reaches hundreds of thousands or more. Since the computational cost of attention increases quadratically with the number of tokens, while the cost associated with the linear layers increases only linearly, the attention mechanism gradually becomes the training bottleneck. In scenarios where the sequence length reaches 1 million, the computation time for attention can account for up to 95% of the end-to-end training time.

During the training process, only the DiT model is optimized while the text encoder and the VAE encoder remain frozen. Therefore, GPU memory usage is concentrated on training the DiT. The GPU memory usage for the DiT can be expressed as  $\gamma Lbsh$ , where  $\gamma$  depends on the implementation of the DiT layers,  $L$  denotes the number of DiT layers. Because video tokens, input prompts, and timesteps are taken as model input, the value of  $\gamma$  is generally larger than in ordinary LLM models. For example, while standard LLMs might have a  $\gamma$  of 34 (Korthikanti et al.), the DiT model's  $\gamma$  can exceed 60. Consequently, when the sequence length reaches 1 million tokens and the micro-batch size is 1, the total GPU memory usage for activations in a 14B DiT model can exceed 8 TB.

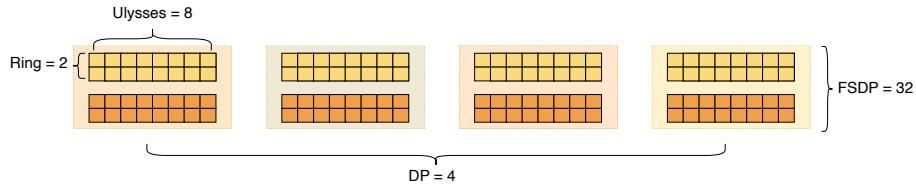


Figure 11: Illustration of DiT parallelism. Assuming a total of 128 GPUs, the innermost layer consists of a combination of Ulysses=8 and Ring=2. The outer layer employs FSDP=32, while the outermost layer utilizes DP=4. The global batch size is 8 times that of the micro-batch size.

In summary, the primary computational cost in Wan arises from the attention mechanism. While the computational cost increases quadratically with the sequence length  $s$ , the GPU memory usage scales only linearly with  $s$ . This provides a valuable reference for guiding subsequent optimization efforts.

#### 4.3.2 PARALLELISM STRATEGY

The Wan model mainly consists of three modules: VAE, text encoder, and DiT. Following the cache mechanism optimization outlined in Sec. 4.1.3, the VAE module exhibits minimal GPU memory usage and can seamlessly adopt Data Parallelism (DP). In contrast, the text encoder requires over 20 GB of GPU memory, necessitating the use of model weight sharding to conserve memory for the subsequent DiT module. To address this, we employ a Data Parallel (DP) approach combined with Fully Sharded Data Parallel (FSDP) (Zhao et al., 2023) strategy. The model parameters, gradients, and optimizer states of the DiT will exceed the capacity of a single GPU, with activation storage reaching approximately 8 TB in a scenario involving 1M tokens with a batch size of 1. Moreover, the DiT portion accounts for a significant proportion of the overall computational workload. Thus, we focus on developing efficient distributed strategies specifically for the DiT module.

**DiT parallel strategy.** In light of the DiT model’s modest size, and considering the optimization of overlapping computation and communication as well as maximizing the size of data parallelism (DP), we have selected FSDP as our parameter sharding strategy.

Regarding activations, the input shape of a DiT block is  $[b, s, h]$ , where the  $b$  dimension corresponds to data parallelism,  $s$  represents the sequence length, and  $h$  denotes the hidden dimension. Therefore, we need to examine sharding strategies for the  $s$  and  $h$  dimensions. Sharding along the  $s$  dimension is achieved through Context Parallelism (CP), with common methods including Ulysses (Jacobs et al., 2023) and Ring Attention (Liu et al., 2023a). Sharding along the  $h$  dimension primarily involves Megatron’s tensor parallelism (TP) (Shoeybi et al., 2019) combined with sequence parallelism (SP) (Korthikanti et al.), which shards the hidden dimension of the activations by splitting the weights. Since the communication overhead of CP is smaller than that of (TP + SP), we propose using CP to accelerate the batch-level computations while reducing the activation memory footprint on each GPU. We have designed a two-dimensional (2D) CP that combines the characteristics of Ulysses and Ring Attention, similar to USP (Fang & Zhao, 2024). In this design, the outer layer employs Ring Attention, and the inner layer leverages Ulysses. This combination mitigates the slow cross-machine communication inherent in Ulysses and addresses the need for large block sizes after sharding in Ring Attention, thereby maximizing the overlap between outer-layer communication and inner-layer computation. In scenarios with a sequence length of 256K and 16 GPUs across 2 machines, the communication overhead of 2D Context Parallelism is reduced from over 10% with Ulysses to below 1%.

The FSDP group and CP group intersect within the FSDP group, the DP size equals the FSDP size divided by the CP size. After satisfying memory and single-batch latency requirements, we use DP for the scaling. Fig. 11 illustrates the distributed scheme for the DiT module. For example, in a configuration with 128 GPUs, the CP size is 16, with Ulysses set to 8 and Ring Attention to 2; FSDP is set to 32, corresponding to a batch size of 2b; DP is set to 4, resulting in a global batch size of 8b.

**Distributed strategy switching for different modules.** During training, different modules utilize the same resources. Specifically, we apply DP to the VAE and Text Encoder, while the DiT module employs a combination of DP and CP. Therefore, when the outputs of the VAE and Text Encoder are fed into the DiT module during training, it is necessary to switch distributed strategies to avoid

resource wastage. Specifically, CP requires that devices within the CP group read the same batch of data. Yet, if devices in the VAE and Text Encoder sections also read the same data, this leads to redundant computations. To eliminate this redundancy, devices within the CP group initially read different data. Then, before CP, we perform a loop traversal of size equal to the CP size, sequentially broadcasting the data read by different devices within the CP group to ensure that the inputs within CP are identical. In this way, the time proportion of VAE and Text Encoder within a single model iteration is reduced to  $1/CP$ , thereby enhancing overall performance.

#### 4.3.3 MEMORY OPTIMIZATION

As introduced, the computational cost in Wan grows quadratically with sequence length  $s$ , whereas GPU memory usage scales linearly with  $s$ . This means that in long sequence scenarios, computation time can eventually exceed the PCIe transfer time required for offloading activations. Specifically, the PCIe transfer time for offloading the activation of one DiT layer can be overlapped with the computation of just 1 to 3 DiT layers. Compared to gradient checkpointing (GC) strategies (Chen et al., 2016), activation offloading (Rhu et al., 2016) that allows computation overlap provides an effective way to reduce GPU memory usage without sacrificing end-to-end performance. Therefore, we prioritize activation offloading to reduce GPU memory. Since CPU memory is also prone to exhaustion in long sequence scenarios, we combine offloading with GC strategies, prioritizing gradient checkpointing for layers with high GPU memory-to-computation ratios.

#### 4.3.4 CLUSTER RELIABILITY

By leveraging Alibaba Cloud’s advanced intelligent scheduling, slow machine detection, and robust self-healing capabilities within the training cluster, high stability is ensured. Hardware issues are identified during the job startup phase, ensuring that only healthy nodes are allocated to training tasks. Throughout training, any faulty nodes are promptly isolated and repaired, with tasks automatically restarted and training seamlessly resumed. This efficient orchestration effectively combines reliability with high performance, ensuring overall system stability.

### 4.4 INFERENCE

The primary objective of inference optimization is to minimize the latency of video generation. Unlike training, the inference process involves multiple sampling steps, typically around 50. Accordingly, we employ techniques such as quantization and distributed computing to reduce the time required for each individual step. Additionally, we leverage the attention similarities between steps to decrease the overall computational load. Furthermore, for classifier-free guidance (CFG) (Ho & Salimans, 2022), we exploit its inherent similarities to further minimize computational requirements.

#### 4.4.1 PARALLEL STRATEGY

As discussed in Sec. 4.3.2, we utilize Context Parallel to reduce the latency of generating a single video when scaling to multiple GPUs. Additionally, for large models like Wan 14B, we adopt model sharding to mitigate GPU memory constraints. *Model Sharding Strategy*: Given the long sequence lengths during inference, FSDP incurs less communication overhead compared to TP and allows for computation overlap. Therefore, we adopt the FSDP method for model sharding, consistent with our training approach. *Context Parallel Strategy*: We employ the same 2D Context Parallelism as that in the training stage, with RingAttention serving as the outer loop and Ulysses operating as the inner loop. Equipped with 2D Context Parallel and FSDP parallel strategy, DiT achieves nearly linear speedup on the Wan 14B model, as shown in the Fig. 12.

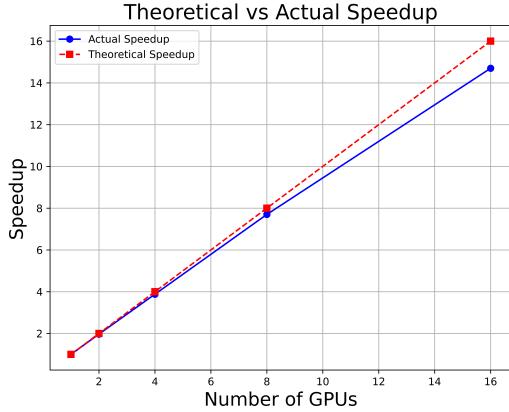


Figure 12: Scaling inference via multiple GPUs.

#### 4.4.2 DIFFUSION CACHE

We conduct a thorough analysis of the Wan model and identify the following characteristics in its inference process:

- Attention similarity: Within the same DiT block, attention outputs across different sampling steps exhibit significant similarity.
- CFG similarity: In the later stages of sampling, there is a notable similarity between conditional and unconditional DiT outputs.

These findings are also highlighted in recent works, such as DiTFastAttn (Yuan et al., 2024b) and FasterCache (Lv et al., 2024). We leverage these characteristics to implement diffusion caching to reduce the computational workload. In practice, we tailor the diffusion cache to the Wan model to ensure lossless performance. Specifically, we use the validation set to select certain steps for caching:

- Attention cache: For the attention component, we perform an attention forward pass every few steps and cache the results, reusing the cached results for the other steps.
- CFG cache: For the unconditional part, we perform a DiT forward pass every few steps and reuse the conditional results for the other steps. Additionally, we apply residual compensation methods similar to FasterCache to prevent the degradation of fine details.

After applying diffusion caching to the Wan 14B text-to-video model, we have improved the inference performance by 1.62 $\times$ .

#### 4.4.3 QUANTIZATION

**FP8 GEMM.** We found that using FP8 precision for the GEMM operations, combined with per-tensor quantization on the weights and per-token quantization on the activations in the sampling steps, incurs minimal performance loss. Thus we apply FP8 quantization to all GEMM operations in the DiT block. The FP8 GEMM delivers twice the performance of BF16 GEMM and achieves a 1.13 $\times$  speedup in the DiT module.

**8-Bit FlashAttention.** Although FlashAttention3 (Shah et al., 2025) achieves high performance, its native FP8 implementation suffers from significant quality degradation in video generation, as observed in our experiments. Conversely, SageAttention (Zhang et al., 2024) employs a mixed precision of int8 and fp16 to reduce precision loss, but it does not provide specific adaptations for Hopper GPUs. As of this report’s publication, subsequent optimizations to SageAttention have demonstrated improved Hopper compatibility. Future work will explore the integration of these performance enhancements.

We present our optimizations applied to FA3-FP8 implementation, focusing on both numerical stability and computational efficiency.

**Accuracy.** To mitigate the numerical errors in attention observed in Wan under FA3-FP8 quantization, we apply two primary techniques:

- Mixed 8-Bit optimization. The native implementation of FA3 uses FP8 (E4M3) for all input tensors ( $Q$ ,  $K$ ,  $V$ ). Based on both our analysis of the data distribution and the empirical evidence from the SageAttention, we adopt a hybrid quantization strategy: Int8 for  $S = QK^T$ , FP8 for  $O = PV$ .
- FP32 accumulation for cross-block reduction. The native FP8 WGMMA implementation employs 14-bit accumulators, making it prone to overflow during long-sequence processing. To address this limitation, our approach leverages FP8 WGMMA for intra-block  $PV$  reduction while implementing cross-block accumulation through CUDA cores with Float32 registers, a strategy inspired by DeepSeek-V3’s (Liu et al., 2024a) FP8 GEMM methodology.

**Performance.** To maintain kernel performance under mixed INT8-FP8 quantization while enhancing the accumulation precision of  $O = PV$ , we employ two techniques:

- Fuse Float32 accumulation with intra-warpgroup pipelining. FlashAttention3 employs an intra-warpgroup pipeline that overlaps WGMMA with softmax and scaling. To mitigate the

User prompts	Rewritten results
A Viking warrior wields a great axe with both hands, battling a mammoth at dusk, amidst a snowy landscape with snowflakes swirling in the air.	An epic battle scene, unfolds as a tall and muscular Viking warrior wields a heavy great axe with both hands, facing off against a massive mammoth. The warrior is clad in leather armor and a horned helmet, with prominent muscles and a fierce, determined expression. The mammoth is covered in long hair, with sharp tusks, and roars angrily. It is dusk, and the snowy landscape is filled with swirling snowflakes, creating an intense and dramatic atmosphere. The backdrop features a barren ice field with the faint outlines of distant mountains. The use of cool-toned lighting emphasizes strength and bravery. The scene is captured in a dynamic close-up shot from a high-angle perspective.
The camera follows a motor-boat chasing dolphins in the sea.	In a documentary photography style, the camera follows a motor-boat chasing a pod of dolphins leaping out of the vast ocean. On the motorboat, there is a driver wearing a life jacket and a safety helmet, with a focused and excited expression. The dolphins are sleek and smooth-skinned, occasionally leaping out of the water and diving back in with agility. The sky is bright blue, the sun is shining, and the sea is shimmering with light. A few white clouds dot the distant sky. The scene is dynamic and full of energy, captured from a medium shot in a tracking perspective.
The tiny Potato King, wearing a majestic crown, sits on the throne as potato subjects pay homage to it.	In a surrealist style, the tiny Potato King wears a magnificent gold crown and sits on a towering throne. Its skin has a natural earthy yellow tone with subtle bumpy textures. The potato subjects are lined up on either side, bowing their heads in homage to the king. The background features the grand interior of a palace, with gold and red decorations that appear luxurious and solemn. A beam of light shines down from above, creating a sacred atmosphere. The scene is captured in a close-up shot from a high-angle perspective.

Table 1: Examples of user prompts and their rewritten results. These examples are in Chinese originally and have been translated into English for the purpose of presentation.

performance penalty introduced by Float32 *PV* accumulation across blocks, we fuse the Float32 accumulation with *rescale\_o* operation.

- Block size tuning. Due to the additional register pressure introduced by Float32 *PV* accumulation register across blocks, we adjust the block size to reduce register spilling. After optimization, our 8-bit FlashAttention achieves 95% MFU on NVIDIA H20 GPUs.

Consequently, with 8-bit FlashAttention, we boost the inference efficiency by more than 1.27×.

#### 4.5 PROMPT ALIGNMENT

Prompt alignment is designed to boost the effectiveness of model inference by aligning the user’s input prompt with the format and style of captions used during training. We employ two primary strategies in this process. First, we augment each video with a variety of captions, thereby increasing diversity. Second, we rewrite user prompts to align them with the distribution of video captions from the training stage. In the following, we detail these strategies.

To accommodate diverse user inputs, each image or video is paired with multiple captions reflecting various styles and lengths. For example, we use captions of differing lengths (e.g., long, medium, and short) and styles (e.g., formal, informal, and poetic) to create a varied set of sample-caption pairs for our training data. These captions establish different mapping relations between text and video, effectively covering the range of prompts that users might provide.

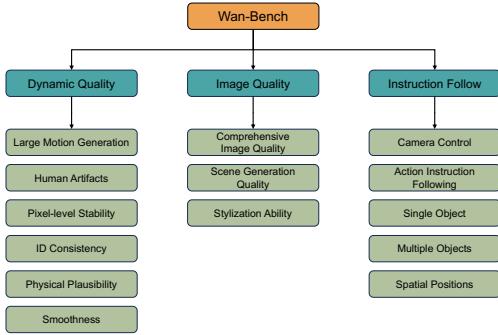


Figure 13: The dimensions covered in Wan-Bench.

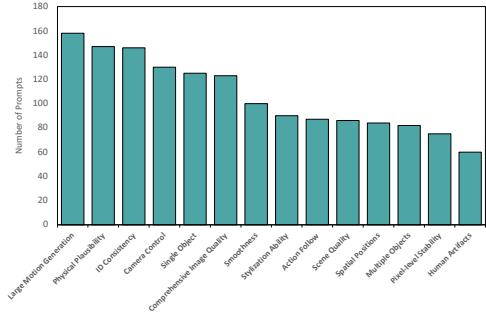


Figure 14: Distribution of the number of prompts across various dimensions of Wan-Bench.

Despite the diversity of captions, a distribution mismatch often exists between the dataset captions and user input prompts. Typically, users prefer concise prompts consisting of a few simple words, which are significantly shorter than the training captions. This discrepancy can adversely affect the quality of the generated videos. To address this issue, we rewrite prompts by utilizing a large language model (LLM) to refine user prompts. Our primary objective is to align the distribution of these refined prompts with the distribution of the training captions, thereby achieving improved inference performance. The LLM is guided by the following principles when rewriting user prompts:

1. We instruct the LLM to add details to prompts without altering their original meanings, enhancing the completeness and visual appeal of the generated scenes.
2. The rewritten prompts should incorporate natural motion attributes, where we add appropriate actions for the subject based on its category to ensure smoother and more fluent motion in the generated videos.
3. We recommend structuring the rewritten prompts similarly to the post-training captions, beginning with the video style, followed by an abstract of the content, and concluding with a detailed description. This method helps align prompts with the distribution of high-quality video captions.

We evaluate the effectiveness of LLM-assisted prompt rewriting on our benchmark, and several results are presented in Tab. 1. Our findings suggest that LLMs with strong instruction-following capabilities can generate suitable prompts that enhance video generation. To balance speed and performance, we use Qwen2.5-Plus as our rewriting model.

#### 4.6 BENCHMARKS

Existing video generation evaluation methods, such as Fréchet Video Distance (FVD) ([Unterthiner et al., 2018](#)) and Fréchet Inception Distance (FID) ([Heusel et al., 2017](#)), lack alignment with human perception. We propose an automated, comprehensive, and human-aligned Wan-Bench for evaluating video generation models.

Wan-Bench is composed of three core dimensions: dynamic quality, image quality, and instruction following with 14 fine-grained metrics. We design specific scoring algorithm for each dimension, using traditional detectors for the simple tasks (*e.g.*, object detection) and multi-modal Large Language Models (MLLMs) for the complex tasks.

**Dynamic quality.** Dynamic quality aims to evaluate the quality and stability of the model in non-static scenarios.

*Large motion generation.* To measure the upper limit of motion generation, prompts are designed to encourage significant motions. We calculate the optical flow of the generated video using RAFT ([Teed & Deng, 2021](#)) and assess the motion scores by the magnitude of the normalized optical flow.

*Human artifacts.* Existing evaluation methods lack the ability to detect AI-generated artifacts. Thus, we train a YOLOv3-based model [Redmon & Farhadi \(2018\)](#) on 20,000 AI-generated images annotated

by humans to identify the locations of artifacts. The artifact scores take into account the predicted likelihood, the bounding boxes, and the corresponding duration.

*Physical plausibility & smoothness.* To evaluate physical plausibility, we design prompts to generate physics-related movements (*e.g.*, ball bouncing, object interactions, water flow) and employ Qwen2-VL (Bai et al., 2023), leveraging its extensive real-world physics knowledge. Through a video question-answer model, we detect violations of physical laws, such as object clipping, unrealistic collisions, or gravity-defying movements. To evaluate smoothness, prompts are crafted to include complex motions, with Qwen2-VL identifying artifacts to assess motion fluidity.

*Pixel-level stability.* Pixel-level stability is used to determine if noise frequently appears in the video. We employ optical flow to identify static regions and calculate frame-wise differences within those areas. For stable videos, these differences within static regions are relatively minimal.

*ID consistency.* Three kinds of sub-dimensions are considered: human consistency, animal consistency, and object consistency covering different collections of prompts. Frame-level DINO (Caron et al., 2021) features are extracted to compare the frame-wise similarity as the consistency score.

**Image quality.** The purpose of the image quality dimension is to evaluate the visual quality and aesthetic perception.

*Comprehensive image quality.* Measuring the visual quality requires an evaluation of the fidelity and aesthetics. For fidelity, we employ MANIQA (Yang et al., 2022), which effectively detects blur and artifacts, ensuring precise clarity assessment. For aesthetics, we utilize two established models: the LAION-based aesthetic predictor (LAION-AI, 2022) and MUSIQ (Ke et al., 2021). We compute the final image quality score by averaging these three evaluators.

*Scene generation quality.* We evaluate two key aspects: frame-wise scene consistency and scene-text alignment. For consistency, we measure CLIP similarity (Radford et al., 2021) across consecutive frames to ensure temporal coherence. For alignment, we compute CLIP similarity between frames and their corresponding text to verify semantic accuracy. We compute the quality score by weighted averaging these metrics.

*Stylization.* We employ Qwen2-VL to evaluate the artistic video generation capability through frame-level question answering.

**Instruction following.** Instruction following assesses the model’s ability to comprehend and execute textual instructions.

*Single object & multiple objects & spatial positions.* We employ Qwen2-VL to predict object categories, counts, and spatial relationships within videos. The evaluation score represents the average number of frames that accurately correspond to the given prompts.

*Camera control.* We evaluate five camera movement styles: panning, lifting, zooming in/out, aerial shots, and tracking shots by 130 tailored prompts. For panning, lifting, and zooming, we analyze optical flow via RAFT to assess movement style. For complex aerial and tracking shots, we employ Qwen2-VL to infer camera movement through video question-answering.

*Action instruction following.* We categorize motions into human (*e.g.*, running), animal (*e.g.*, crawling), and object movements (*e.g.*, flying). As evaluating these motions requires prior knowledge of their execution, we employ Qwen2-VL by providing it with keyframes. We query the model for alignment of the action, completion of the action, and presence of artifacts to comprehensively evaluate alignment between text and video.

**Human feedback guided weighting strategy.** We utilize human feedback to assign a comprehensive video quality score, considering that each dimension has a different impact on user preferences rather than simply averaging them. We collect over 5,000 pairwise comparisons of videos generated by different models including Wan. Users evaluate pairs based on a given text, indicating their preference and assigning approximate scores. The dimension preference weighting is derived from the Pearson correlation between model-generated scores and human-rated scores, serving as a weighting factor in the final score calculation.

Wan-Bench Dimension	CNTopB	Hunyuan	Mochi	CNTopA	Sora	Wan 1.3B	Wan 14B
Large Motion Generation	0.405	0.413	0.420	0.284	<b>0.482</b>	<u>0.468</u>	0.415
Human Artifacts	0.712	0.734	0.622	<b>0.833</b>	<u>0.786</u>	0.707	0.691
Pixel-level Stability	0.977	<b>0.983</b>	<u>0.981</u>	0.974	0.952	0.976	0.972
ID Consistency	<u>0.940</u>	0.935	0.930	0.936	0.925	0.938	<b>0.946</b>
Physical Plausibility	0.836	0.898	0.728	0.759	<u>0.933</u>	0.912	<b>0.939</b>
Smoothness	0.765	0.890	0.530	0.880	<b>0.930</b>	0.790	<u>0.910</u>
Comprehensive Image Quality	0.621	0.605	0.530	<b>0.668</b>	<u>0.665</u>	0.596	0.640
Scene Generation Quality	0.369	0.373	0.368	<u>0.386</u>	<b>0.388</b>	0.385	<u>0.386</u>
Stylization Ability	<b>0.623</b>	0.386	0.403	0.346	<u>0.606</u>	0.430	0.328
Single Object Accuracy	<b>0.987</b>	0.912	0.949	0.942	0.932	0.930	<u>0.952</u>
Multiple Object Accuracy	0.840	0.850	0.693	<u>0.880</u>	<b>0.882</b>	0.859	0.860
Spatial Position Accuracy	<u>0.518</u>	0.464	0.512	0.434	0.458	0.476	<b>0.590</b>
Camera Control	0.465	0.406	<b>0.605</b>	<u>0.529</u>	0.380	0.483	0.527
Action Instruction Following	<b>0.917</b>	0.735	<u>0.907</u>	0.783	0.721	0.844	0.860
<b>Weighted Score</b>	0.690	0.673	0.639	0.693	<u>0.700</u>	0.689	<b>0.724</b>

Table 2: Performance comparison of commercial and open-source models using Wan-Bench.

## 4.7 EVALUATION

### 4.7.1 METRICS AND RESULTS

**Baselines and metrics.** As of the time of writing, there are numerous comparative competitors in the market, including commercial models such as Kling (Kuaishou, 2024.06), Hailuo (MiniMax, 2024.09), Sora (OpenAI, 2024), Runway (Runway (2024.06) and Vidu (ShengShu-AI, 2024.07), as well as open-source models like Mochi (GenmoTeam, 2024), CogVideoX (Yang et al., 2025b) and Hunyuan (Kong et al., 2024). In this section, we compare our model with the commercial and open-source models that have been evaluated through benchmarks and received relatively positive user feedback.

**Quantitative results.** We collected 1,035 samples for each candidate model using Wan-Bench prompts and conducted fair evaluations by scoring the models with Wan-Bench. The evaluation focused on three key metrics: dynamic quality, image quality, and instruction-following accuracy. We rank the performance of the different generation models by calculating total scores weighted according to human preferences (Sec. 4.6). The results in Table 2 indicate that Wan outperforms existing commercial and open-source models.

**Qualitative results.** As illustrated in the Fig. 15, Wan effectively generates diverse, high-quality videos directly from textual descriptions. Wan excels at synthesizing dynamic scenes involving large-scale, complex movements, as well as scenarios reflecting physical interactions. Moreover, it adeptly handles various artistic styles and consistently produces cinematic-quality visuals. Additionally, Wan demonstrates robust multilingual text-generation capabilities, integrating Chinese and English texts into animations and giving rise to movie-level textual effects.

**Human evaluation.** We design over 700 evaluation tasks, which are annotated by more than 20 individuals and assessed by four key dimensions: alignment, image quality, dynamic quality, and overall quality. The results in Table 3 demonstrate that in the T2V task, the Wan 14B model consistently excels across all visual quality dimensions, indicating its superior performance.

**Wan in public leaderboard.** Wan has demonstrated state-of-the-art performance on the VBench leaderboard (Huang et al., 2023), a widely used evaluation approach for video generation task. The VBench employs a multidimensional assessment architecture that decomposes video quality evaluation into 16 distinct, human-aligned dimensions spanning aesthetic quality, motion smoothness, and semantic coherence.

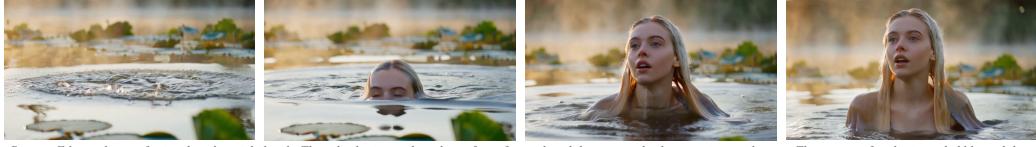
Our analysis focuses on two model variants: the 14B parameter Wan 14B and the Wan 1.3B variant (shown in Table 4). The Wan 14B model achieves a benchmark-leading performance with an aggregate score of 86.22%, including 86.67% in visual quality and 84.44% in semantic consistency,



*Prompt: Retro 80s Monster Horror Comedy Movie Scene: Color film, children's bedroom bathed in soft, warm light. Plush monsters of various sizes and colors are having a chaotic party, jumping on the bed, dancing to upbeat music, and throwing confetti. The walls are adorned with posters of classic 80s movies, and the room is filled with the playful laughter of children.*



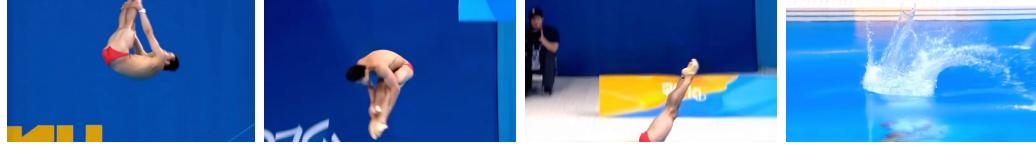
*Prompt: A sepia-toned vintage photograph depicting a whimsical bicycle race featuring several dogs wearing goggles and tiny cycling outfits. The canine racers, with determined expressions and blurred motion, pedal miniature bicycles on a dusty road. Spectators in period clothing line the sides, adding to the nostalgic atmosphere. Slightly grainy and blurred, mimicking old photos, with soft side lighting enhancing the warm tones and rustic charm of the scene. 'Bicycle Race' captures this unique moment in a medium shot, focusing on both the racers and the lively crowd.*



*Prompt: Film quality, professional quality, rich details. The video begins to show the surface of a pond, and the camera slowly zooms in to a close-up. The water surface begins to bubble, and then a blonde woman is seen coming out of the lotus pond soaked all over, showing the subtle changes in her facial expression, creating a dreamy atmosphere.*



*Prompt: Sports photography full of dynamism, several motorcycles fiercely compete on the loess flying track, their wheels rolling up the dust in the sky. The motorcyclist is wearing professional racing clothes. The camera uses a high-speed shutter to capture moments, follows from the side and rear, and finally freezes in a close-up of a motorcycle, showcasing its exquisite body lines and powerful mechanical beauty, creating a tense and exciting racing atmosphere. Close up dynamic perspective, perfectly presenting the visual impact of speed and power.*



*Prompt: A professional male diver performs an elegant diving maneuver from a high platform. Full-body side view captures him wearing bright red swim trunks in an upside-down posture with arms fully extended and legs straight and pressed together. The camera pans downward as he dives into the water below, creating a dramatic splash with perfect entry form.*



*Prompt: A weightless young man, with soft features and an expression of serene astonishment, is slowly drifting above a sun-drenched field of swaying grass. Filmed in a retro cinematic style with warm golden tones and slight grain, side light accentuates the texture of his tousled hair as the wind gently brushes through it. The wide-angle lens captures the expansive landscape, enhancing the quiet, surreal levitation scene filled with calm wonder and gentle absurdity, defying reality's expectations.*



*Prompt: A distinctive father-son duo rides bicycles through city streets, clad in eye-catching attire - the father in a vibrant red suit and the son in neat school uniform. Their striking feature: giant yellow balloons replacing their heads, each meticulously painted with celebratory Chinese character "囍" in bold black ink, gently swaying in the wind as they pedal through the urban landscape.*



*Prompt: A retro 70s-style title sequence for a fictional action movie. Hand-drawn, stylized text "WAN" appears dynamically on screen, overlaid on fast-paced clips of car chases, explosions, and daring stunts. The text is bold, gritty, and slightly distorted, reflecting the 70s action movie aesthetic. A montage of high-octane scenes with a retro film grain effect, featuring warm, vintage colors. The sequences are bathed in golden hour light, enhancing the nostalgic feel.*

Figure 15: Results of the Wan-T2V. Our model excels at generating complex motions, creative transitions, cinematic-quality videos, and accurately produces both Chinese and English text.

	CN-TopA	CN-TopB	CN-TopC	Runway	All Rounds
Visual Quality	30.6%	15.9%	27.8%	48.1%	5710
Motion Quality	16.1%	9.7%	14.9%	40.3%	5785
Matching	46.0%	57.9%	56.7%	69.1%	5578
Overall Ranking	44.0%	44.0%	48.9%	67.6%	5560

\* The table displays the proportion of instances in which the Wan 14B model outperformed other models in pairwise comparisons, relative to the total number of comparisons.

Table 3: Win rate gap of T2V.

significantly outperforming competitors like OpenAI’s Sora and MiniMax’s Hailuo. The efficient Wan 1.3B variant also remains competitive, scoring 83.96% and surpassing commercial models such as HunyuanVideo (Kong et al., 2024) and Kling 1.0 (Kuaishou, 2024.06), as well as the open-sourced CogVideoX1.5-5B (Yang et al., 2025b).

Model Name	Quality Score	Semantic Score	Total Score
MiniMax-Video-01 (MiniMax, 2024.09)	84.85%	77.65%	83.41%
Hunyuan (Open-Source Version) (Kong et al., 2024)	85.09%	75.82%	83.24%
Gen-3 (2024-07) (Runway, 2024.06)	84.11%	75.17%	82.32%
CogVideoX1.5-5B (5s SAT prompt-optimized) (Yang et al., 2025b)	82.78%	79.76%	82.17%
Kling (2024-07 high-performance mode) (Kuaishou, 2024.06)	83.39%	75.68%	81.85%
Sora (OpenAI, 2024)	85.51%	79.35%	84.28%
Wan 1.3B	84.92%	80.10%	83.96%
Wan 14B (2025-02-24)	<b>86.67%</b>	<b>84.44%</b>	<b>86.22%</b>

Table 4: Model performance scores on Vbennch.

#### 4.7.2 ABLATION STUDY

We provide ablation studies of key modules within our model design to offer insights into the overall architecture’s contributions. Experiments are conducted on 1.3B version for rapid evaluation.

**Ablation on adaptive normalization layers.** Given the substantial parameter load of adaptive normalization layers (Perez et al., 2018) (adaLN) in DiT (Peebles & Xie, 2023), we explore whether focusing on parameter volume in adaLN or increasing the depth of the network layers is more effective. To adjust the parameter volume, we follow the setup from PixArt (Chen et al., 2023a) to study whether to share adaLN or not. In the non-shared configuration, each block’s scale and shift parameters are predicted by a block-specific MLP that takes as input the time embedding. Instead, the shared AdaLN configuration (referred to as AdaLN-single in PixArt) computes a single set of global parameters, which involves predicting scale and shift values, in the first block, which are then shared across all blocks. This sharing significantly reduces the number of parameters.

We evaluate four configurations: (i) Full-shared-adaLN-1.3B: This is our original model, where adaLN is fully shared across all 30 attention blocks. (ii) Half-shared-adaLN-1.5B: AdaLN is shared in the first 15 attention blocks while not for the remaining blocks. This results in a 1.5B model with an increase of 0.2B parameters. (iii) Full-shared-adaLN-1.5B (extended): AdaLN is shared across all attention blocks, but the model depth is increased to 35 layers, keeping the 1.5B parameter count. (iv) Non-shared-AdaLN-1.7B: AdaLN is not shared across any blocks, and the model depth remains at 30 layers, resulting in a 1.7B model. We maintain other parameters unchanged and train the models from scratch to perform text-to-image task for 200,000 steps (i.e., the first stage of our text-to-video training), with the global batch size of 1536. For comparison, we measure performance using the L2 loss between generated images and real images in the latent space during the training stage, with smaller training loss indicating better convergence.

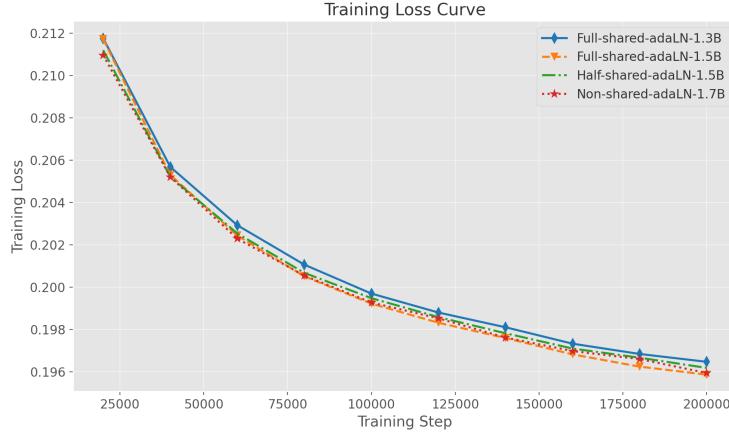


Figure 16: Training loss curve with different configurations.

As presented in Fig. 16, the Full-shared-AdaLN-1.3B model, with fewer parameters, results in a slightly higher training loss compared to others. When comparing Half-shared-AdaLN-1.5B and Full-shared-AdaLN-1.5B, which have the same parameter count, Full-shared-AdaLN-1.5B consistently achieves the lowest training loss. This suggests that focusing on model depth rather than the parameters in AdaLN leads to better performance. Additionally, the Non-shared-AdaLN-1.7B model, despite having more parameters, does not outperform the Full-shared-AdaLN-1.5B and further supports this idea. Therefore, we adopt the fully shared AdaLN design to effectively reduce parameter count while preserving performance.

**Ablation on text encoder.** We select three text encoders that can handle bilingual inputs: umT5 (Chung et al., 2023) (5.3B), Qwen2.5-7B-Instruct (QwenTeam, 2024), and GLM-4-9B (TeamGLM et al., 2024). The T5 series has been a popular choice in video generation models (GenmoTeam, 2024; Yang et al., 2025b), while the latter two encoders excel in language understanding, especially among all LLM models under 10B parameters. For this ablation, we keep other settings unchanged and train a text-to-image task with a global size of 1536. Text embeddings are derived from the second-to-last layers of Qwen2.5-7B-Instruct and GLM-4-9B. We measure the training loss as shown in Fig. 17.

Compared to other powerful LLM-based encoders, umT5 demonstrates superior text embedding performance. Notably, Hunyuan-Video (Kong et al., 2024) highlights that decoder-only LLMs use causal attention whereas umT5 adopts bidirectional attention, making it suitable for diffusion models. Following HunyuanVideo’s strategy, we incorporate a bidirectional token refiner as an adapter layer. We apply this setup to both Qwen2.5-7B-Instruct and GLM-4-9B, but the training losses and visualizations still show umT5 performing most favorably. Further, we compare umT5 with a pre-trained Multimodal Large Language Model, i.e., Qwen-VL-7B-Instruct (Bai et al., 2023). As shown in Table 6, using the second-to-last layer’s features from Qwen-VL results in comparable generation performance to umT5, but with a larger model size.

Model	VAE	VAE-D
10k steps	<b>42.60</b>	44.21
15k steps	<b>40.55</b>	41.16

Table 5: FID scores ( $\downarrow$ ) of VAE and VAE-D.

Models	umT5	Qwen-VL-7B (last layer)	Qwen-VL-7B (second last layer)
FID( $\downarrow$ )	43.01	43.72	<b>42.91</b>

Table 6: FID scores for different text encoders. The “last layer” and “second last layer” refer to the text features extracted from the last and second last layers of the encoder, respectively.

**Ablation on autoencoder.** In addition to our VAE, we design a variant called VAE-D, where the reconstruction loss is replaced by a diffusion loss. We adopt both the pre-trained VAE and VAE-D on the text-to-image generation task for 150,000 training steps until achieving convergence. We measure

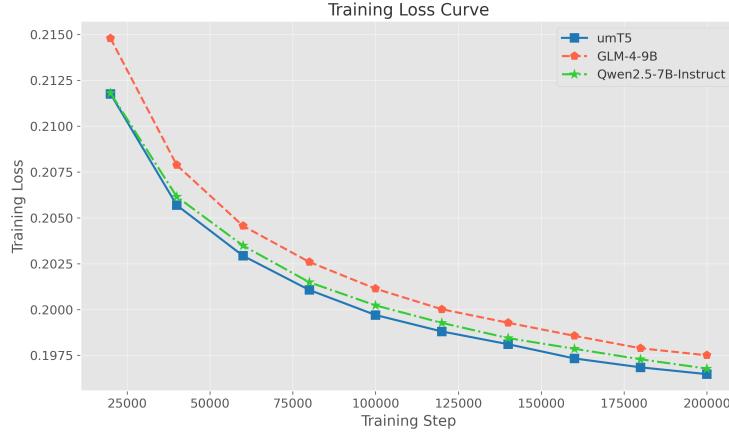


Figure 17: Training loss curve with different text encoders.

the FID scores at both 100,000 and 150,000 steps. As shown in Table 5, VAE model consistently achieves lower FID compared to VAE-D.

## 5 EXTENDED APPLICATIONS

### 5.1 IMAGE-TO-VIDEO GENERATION

The image-to-video (I2V) generation task focuses on synthesizing a dynamic video sequence from a static input image guided by a textual prompt. This approach significantly enhances the controllability of video generation by anchoring the output to a specific initial frame, thereby garnering substantial interest within the research community. Existing approaches, such as I2VGen-XL (Zhang et al., 2023c), SVD (Blattmann et al., 2023), and CogVideo (Yang et al., 2025), extend T2V frameworks to I2V by channel-wise concatenation of the conditional latent representation with the noise latent. Building upon this paradigm, our Wan-I2V model adopts a similar strategy, capitalizing on the robust prior knowledge embedded in our foundational T2V model to achieve high-quality I2V generation.

#### 5.1.1 MODEL DESIGN

We introduce an additional condition image as the first frame to control video synthesizing. Specifically, we concatenate the condition image  $I \in R^{C \times 1 \times H \times W}$  with zero-filled frames along the temporal axis, and these guidance frames  $I_c \in R^{C \times T \times H \times W}$  are compressed by Wan-VAE into condition latent  $z_c \in R^{c \times t \times h \times w}$  ( $c = 16$  denotes the latent channel,  $t = 1 + (T - 1)/4$ ,  $h = H/8$ ,  $w = W/8$ ). Additionally, we introduce a binary mask  $M \in \{0, 1\}^{1 \times T \times h \times w}$  where 1 indicates the preserved frame and 0 denotes the frames to be generated. The spatial size of mask  $M$  is consistent with condition latent  $z_c$ , but  $M$  shares the same temporal length with the target video, which is then rearranged as shape of  $s \times t \times h \times w$  where  $s$  is temporal stride of Wan-VAE. The noise latent  $z_t$ , condition latent  $z_c$ , and rearranged mask  $m$  are concatenated along the channel axis and then passed through the DiT model of Wan. Since the input of the I2V DiT model has more channels than T2V (*i.e.*,  $2 \times c + s$  v.s.  $c$ ), an additional projection layer is employed, which is initialized with zero values. Moreover, we utilize the image encoder of CLIP (Radford et al., 2021) to extract feature representations from the condition image, and the extracted features are projected by a three-layer multi-layer perceptron (MLP), which plays a role as global context. The produced global context is then injected into the DiT model via decoupled cross-attention.

In practice, the aforementioned approach as illustrated in Fig. 18 is extensible to other controllable generation tasks, such as first-last frame transformation and video continuation. The proposed masking mechanism explicitly delineates the input conditions and specifies the frames to be generated. To optimize the efficacy of the masking strategy, we incorporate multiple tasks, *i.e.*, image-to-video generation, video continuation, first-last frame transformation, and random frame interpolation, into our unified framework. Specifically, we adopt the following training paradigm for the I2V model.

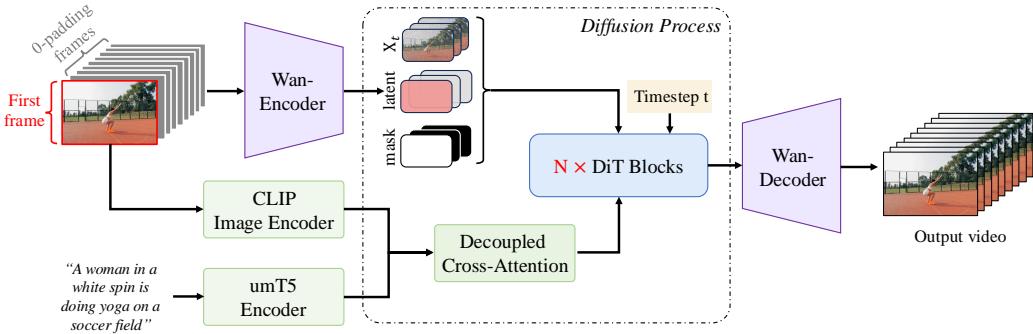


Figure 18: Wan-I2V model framework. The mask mechanism enables the model to selectively focus on input frames such that the proposed framework demonstrates compatibility with video continuation and first-last frame transformation tasks.

During the joint training phase, we conduct unified pre-training of our mask-guided model across these diverse tasks. This phase facilitates the model’s ability to discern which positions should be preserved and which should be generated, depending on the input content. Subsequently, in the fine-tuning stage, we concentrate on refining the model’s performance on each individual task.

### 5.1.2 DATASET

During the joint training phase, we utilize the same training dataset employed for T2V pertaining. This phase equips the model with the foundational capability to generate motion conditioned on reference images. In the fine-tuning phase, we curate task-specific datasets tailored to the unique characteristics of each individual task.

**Image-to-video dataset.** In the early experiments, we found that when the first frame in the training data significantly differed from the video content, the model struggled to learn stable image-driven video generation. From this perspective, a smaller difference between the first frame and the video content in the training data is beneficial for model training. We calculate the difference between the first frame and the video content based on SigLIP (Zhai et al., 2023) features. Specifically, we compute the cosine similarity between the features of the first frame and the mean of the features of the remaining frames, and then we retain only the videos with a similarity greater than a predefined threshold.

**Video continuation dataset.** Similarly, our findings also indicate that training on temporally consistent videos significantly improves the performance of video continuation tasks. To quantify the consistency between the initial and final segments of a video (*i.e.*, the first 1.5 seconds and the last 3.5 seconds), we compute the cosine similarity of their respective SigLIP features, followed by using the computed similarity scores to filter and select the required training dataset.

**First-last frame transformation dataset.** Unlike the I2V task, the community places greater emphasis on the smooth transition between the initial and final frames in the mage transition task. To this end, we increase the proportion of data samples with significant transition between the first and last frames in the training dataset.

### 5.1.3 EVALUATION

The training of the I2V model, along with first-last frame transformation and video continuation, is structured into two stages. In the initial pre-training stage, we employ the same dataset as that used in the T2V task. Given the slightly larger number of frames involved in this stage, the image encoder branch is excluded, which is motivated by the need to maintain a strong alignment between textual input and visual output, ensuring that the model retains its sensitivity to nuanced textual semantics. Upon the stabilization of the pre-training stage, the model has already manifested a variety of capabilities, including I2V generation, video continuation, and so on. This demonstrates the efficacy of the pre-training process in establishing a robust foundation for both spatial and temporal understanding.



*Retro Hong Kong-style photography focuses on a fashionable young man wearing a vibrant yellow velvet jacket paired with a black turtleneck sweater. He slowly removes a pair of stylish sunglasses, revealing a mysterious gaze. Gently cradling a gray kitten in his arms, his tender embrace creates a cute contrast to his rugged appearance. The background features a pastel-pink wall, casting a dreamy atmosphere. Wearing black leather shoes, he stands at the center of the frame, commanding attention. The scene is a close-up portrait, with the subject facing the camera at a side angle, exuding his unique personal charm.*



*A video frame in an aesthetic and dreamy style features a Caucasian woman with long golden hair gently swaying in the wind. She stands sideways beneath a dazzling starry sky. Adorned with delicate golden earrings and a necklace, she wears a flowing silver dress that shimmers brilliantly, its hem softly swaying in the breeze like the brightest star in the night sky. The background is a deep cosmic universe dotted with countless stars, crossed by the luminous Milky Way, creating an aura of mystery and romance. Her gaze shifts from being lowered to softly lifting with tenderness as she looks directly at the camera, her lips curving into a faint smile. Her expression is serene yet filled with anticipation. In a medium shot, her smooth motion transitions from a side profile to facing the camera head-on, capturing the tranquil beauty of the fleeting moment.*



*A surreal digital illustration features a relaxed white cat wearing pink wireless headphones, immersed in its own musical world. Its pointy ears gently press against a pair of pink sunglasses, adding a touch of fashion. The cat is skillfully blowing a transparent bubblegum bubble with its front paw, the bubble swaying lightly in the air as it reaches its maximum size. The background is a soft, pure pink, creating a dreamlike atmosphere. The cat's eyes are slightly open, exuding an expression of contentment and enjoyment, as if the entire world has turned sweet. A close-up shot focuses on the moment the cat blows the bubble, capturing the serenity and playfulness of the scene. In a near view, from an eye-level perspective, the details of the cat's facial expressions and movements are emphasized.*



*Dynamic surrealist digital art depicts three faces of different colors slowly opening their eyes and offering a warm smile to the camera. The face on the left is pink with striking red lips, the one in the middle is white with long, shiny black hair, and the one on the right is yellow, adorned with vibrant orange lips. All three start with closed eyes in serene stillness, then gently open them, revealing soft, affectionate smiles as if forming a magical connection with the viewer. The background features soft, light tones that enhance the facial features, creating a dreamy and mysterious atmosphere. Close-up shots capture the subtle changes in expression, utilizing smooth tracking technology to make every moment come alive. A near view showcases the multi-faceted close-up, accompanied by meaningful eye contact.*



*A summer ocean scenery video features an elegant woman wearing a light beige sleeveless top, seated in a retro wooden cabin. Behind her lies a shimmering azure sea and distant misty emerald islands. Her wavy golden hair flows freely in the wind, resting gently on her shoulders. Sunlight streams through the porthole, illuminating her warm smile as she faces the camera sideways, her elbow resting lightly on the seat in a relaxed and charming posture. Outside the window, a clear sky dotted with a few leisurely white clouds creates a serene and carefree atmosphere. The high-definition video captures every ray of sunlight and every detail of the waves, showcasing the romantic allure of summer sailing. A close-up shot of the woman facing the camera sideways is paired with expansive seascapes filmed using a wide-angle lens.*



*A 3D-rendered New Year's celebration scene shows a crowd with their backs to the camera, gazing up at fireworks blooming in the night sky. The massive "2025" digits, composed of dazzling fireworks, illuminate the entire sky. People are dressed in various winter outfits, with different postures—some have their arms crossed over their chests, while others raise their phones to capture the moment. In the background, the silhouette of the city is visible, with skyscrapers and trees faintly outlined. At the instant the fireworks explode, the crowd falls into an unusual stillness, collectively welcoming the arrival of the new year. Long shot, viewed from a low-angle perspective.*

**Figure 19:** Videos generated by Wan-I2V model. The results demonstrate that our I2V model effectively captures and replicates the dynamics of real-world scenes.



A digital illustration depicts a fencer in a mask, dressed in a sleek, white fencing uniform, holding a silver foil with a serious expression and sharp, focused eyes that reveal a strong fighting spirit. The deep black ground is illuminated only by faint light outlining its silhouette, heightening the sense of tension. The fencer steps forward with its left foot, bends its right knee slightly, and assumes a standard foiling stance, as if poised to launch a swift attack in the next moment. The air is thick with concentration and the aura of a duel. Rendered with intricate brushstrokes and high-contrast colors, the artwork creates a highly dynamic and tense visual effect. A closeup full-body shot of the character is shown from a side view.



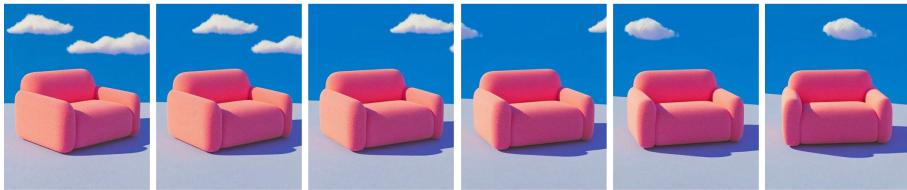
A dreamlike garden scene in digital art style features an orange-and-blue butterfly gracefully fluttering in the air, surrounded by several translucent pink butterflies. The vibrant pink flowers in full bloom have delicate petals, golden stamens, and fresh green leaves. The background of a blue sky filled with shimmering light creates a fairy-tale-like atmosphere. In a close-up shot with the camera slightly tilted upward, the interaction between the butterflies and the flowers is captured, filling the entire frame with life and energy.



The video is presented in an American TV cartoon style, featuring an adult dressed in an all-white, high-quality fluffy rabbit costume. The outfit is intricately detailed, including a 3D rabbit ear hat and a fluffy tail, creating a soft and adorable atmosphere. The model stands in front of a simple gray backdrop with their hands casually tucked into the pockets, exuding a relaxed and carefree attitude. The costume perfectly replicates the form of a rabbit, delivering a warm, whimsical visual effect. In a medium shot, the model slowly walks toward the light steps, as if stepping into a fairytale world. The video uses smooth tracking shots to capture every lively moment.



A dynamic video scene captures a passionate street artist immersed in playing the violin against a pure white background. He wears a wide-brimmed straw hat, its brim gently shading his bright eyes, and dresses in a simple white shirt paired with retro brown trousers. A saxophone hangs stylishly at his waist, adding a touch of effortless charm. The artist's fingers move nimbly across the strings, fully absorbed in the music; his face radiating a deep love and dedication for his craft. The surrounding blank space stretches infinitely, as if the entire world is immersed in his melody, tranquil and pure. In a medium shot, the camera captures the performer from the front, focusing on the moment of complete devotion.



A modern art piece in 3D rendering style features a pink single-seat sofa chair placed on a smooth floor. The design of the sofa chair is minimalist, with a subtle granular texture on its surface. The background is a clear blue sky dotted with a few white clouds, creating an overall bright and fresh color palette. With no people in the scene, the sofa chair stands out prominently. The shot is taken in a medium view, with a slight rotation of the camera emphasizing the details and texture of the sofa chair.



An illustration in a fairy-tale style depicts a little girl riding on the back of a giant owl in flight. The girl wears a green dress, her hair tied into twin pigtails, and her expression is one of excitement and focus. The owl spreads its broad wings, its bright eyes seemingly guiding the way ahead. The background features mountains and rivers under a starry sky, with moonlight casting its glow over the land and luminous flowers scattered around. The scene is filled with an atmosphere of dreams and adventure. Close-up, viewed from a low-angle perspective.

Figure 20: Videos generated by Wan-I2V model. The results indicate that our I2V model can effectively animate diverse types of images into highly realistic and dynamic videos.

	CN-TopA	CN-TopB	CN-TopC	CN-TopD	All Rounds
Visual Quality	29.2%	60.8%	24.6%	55.6%	896
Motion Quality	21.7%	21.7%	32.5%	67.0%	890
Matching	-4.2%	35.0%	51.7%	72.2%	896
Overall Ranking	10.8%	47.5%	50.8%	81.6%	890

Table 7: Win rate gap of I2V models. The values in the table represent the proportion of instances in which the Wan-I2V model was preferred in pairwise comparisons against other models, relative to the total number of comparisons conducted.

Our empirical observations reveal that relying solely on frame-level information is inadequate for achieving optimal performance in specific tasks, such as image-to-video generation, video continuation, and first-last frame transformation.

This insight indicates that while the pre-trained model offers a foundational framework for frame-based video generation, it lacks the requisite contextual and semantic depth to fully capture the temporal consistency of dynamic video sequences, particularly when the number of conditioning frames is highly limited.

To address this limitation, during the SFT stage, we incorporated an image encoder to extract image features through decoupled cross-attention (Ye et al., 2023) like (Guo et al., 2024a), providing global contextual information to enhance the model’s capacities. We conducted experiments based on two pre-trained checkpoints at 480p and 720p resolutions, covering the training of image-to-video generation, video continuation, and first-last frame transformation tasks.

Similarly to human evaluation for our T2V model, we benchmark our I2V model against state-of-the-art techniques to provide a comprehensive comparison focusing on key aspects such as visual quality, motion quality, and matching. The results shown as Table 7 demonstrate that our model performs favorably across all dimensions evaluated. Fig. 19 and Fig. 20 present additional visualizations of our I2V model, offering a clear illustration of its superior performance in animating images into high-quality videos.

## 5.2 UNIFIED VIDEO EDITING

The foundational pre-trained models for text-to-image and text-to-video generation have facilitated the expansion of various downstream tasks and applications, including repainting (AI, 2022; Zhou et al., 2023), editing (Meng et al., 2021; Brooks et al., 2023; Zhang et al., 2023a; Wang et al., 2023b; Wei et al., 2024b), controllable generation (Zhang et al., 2023b; Jiang et al., 2024; Wang et al., 2024c), customized generation (Chen et al., 2023b; Wei et al., 2025), frame reference generation (Yang et al., 2025b; Guo et al., 2024a), efficient generation (Wang et al., 2023c; Yuan et al., 2024a; Liu et al., 2024b), and ID-referenced video synthesis (Pan et al., 2024; Yuan et al., 2025; Liu et al., 2025a). To enhance task flexibility and minimize the overhead associated with deploying multiple models, researchers have increasingly focused on developing unified model architectures (Tan et al., 2024; Chen et al., 2024), such as ACE (Han et al., 2025; Mao et al., 2025) and OmniGen (Xiao et al., 2024). These unified architectures aim to integrate diverse tasks within a single image model, thereby facilitating the creation of various application workflows while maintaining ease of use. In the domain of video, the collaborative transformations in both temporal and spatial dimensions suggest that leveraging a unified model can unlock limitless possibilities for video creation.

In our previous work, VACE (Jiang et al., 2025), we introduced a unified framework for controllable video generation and editing, addressing tasks such as reference-to-video generation, video-to-video editing, and masked video-to-video editing. Building upon the pre-trained model of Wan, we integrate multiple modalities, including images and videos for editing, references, and masks, into the Video Condition Unit (VCU) to support diverse input formats. VACE employs a concept decoupling strategy that enables the model to identify which aspects should be retained and which should be modified,

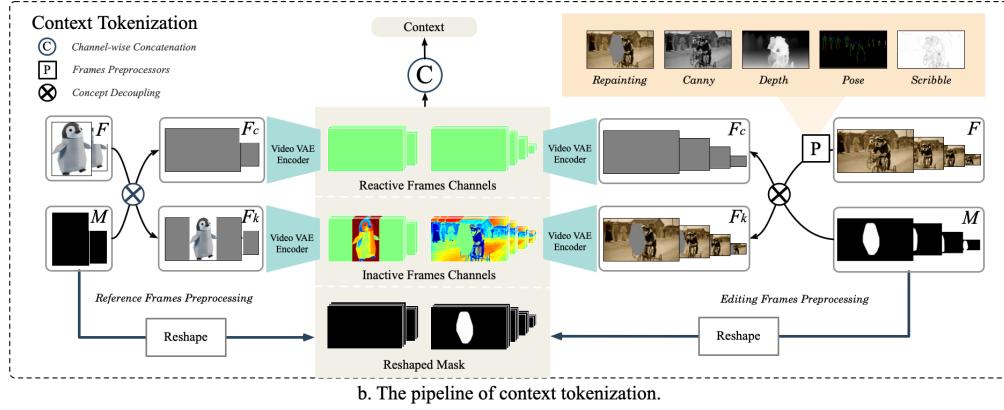
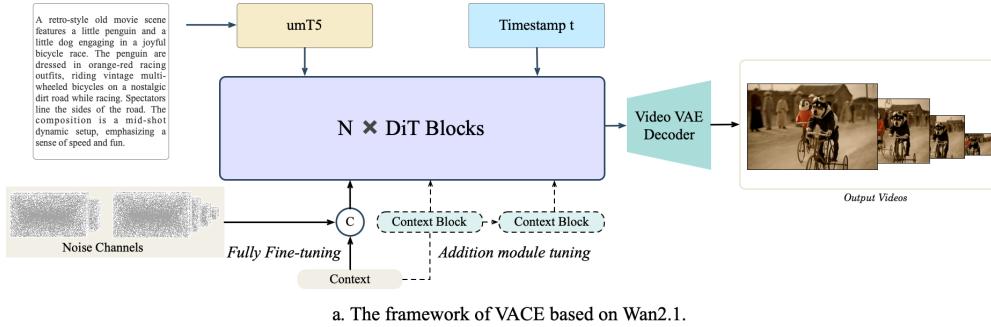


Figure 21: Unified controllable generation and editing framework. Frames and masks are tokenized through Concept Decoupling, Context Latent Encode, and Context Embedder. We propose two training strategies for Wan: Fully Fine-tuning and Context Adapter Tuning.

thereby distinctly separating visual modality information in editing and reference tasks. We offer two training modes. The first mode involves training with the VCU as input, fully fine-tuning the entire Wan model. The second mode utilizes a pluggable Context Adapter structure in a Res-Tuning (Jiang et al., 2023) manner, allowing support for controllable and editing tasks without modifying the base model’s weights but with an increased overall model scale. Leveraging the robust video generation capabilities of Wan, this innovative framework demonstrates significant competitiveness in both quantitative and qualitative evaluations. It facilitates the compositional expansion of fundamental tasks, enabling scenarios such as long video re-rendering. Consequently, the framework offers a versatile and efficient solution for video synthesis, opening new avenues for user-driven video content creation and editing.

### 5.2.1 MODEL DESIGN

The Video Condition Unit (VCU), proposed in VACE, serves as an input paradigm that unifies diverse input conditions into textual inputs, frame sequences, and mask sequences. It can be formally represented as follows:

$$V = [T; F; M], \quad (4)$$

where  $T$  is a text prompt, while  $F$  and  $M$  are sequences of context video frames  $\{u_1, u_2, \dots, u_n\}$  and masks  $\{m_1, m_2, \dots, m_n\}$  respectively. Here,  $u$  is in RGB space, normalized to  $[-1, 1]$  and  $m$  is binary, in which “1”s and “0”s symbolize where to edit or not.  $F$  and  $M$  are aligned both in spatial size  $h \times w$  and temporal size  $n$ . Under this paradigm, we employ the previously described Wan-VAE to tokenize the input video frames and mask information into context tokens. As illustrated in Fig. 21(a), these context tokens are then combined with noisy video tokens to fine-tune the Wan model. Additionally, a Context Adapter Tuning strategy is proposed, which allows context tokens to pass through Context Blocks and be reintegrated into the original DiT blocks. This approach facilitates the seamless integration of contextual information, enhancing the model’s ability to handle diverse editing and generation tasks without altering the base model’s weights.



**TASK-OUTPAINTING:** A large spaceship is flying through space, with a smaller ship visible in the background. Suddenly, the larger ship explodes in a massive fireball, sending debris flying in all directions. The explosion is intense and bright, with flames and smoke billowing out from the wreckage. The smaller ship remains unharmed and continues to fly away from the scene. ...



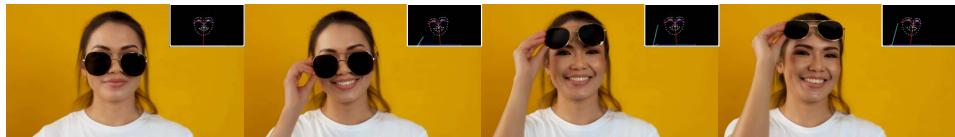
**TASK-INPAINTING:** A person is painting on a canvas outdoors, using a palette with various colors of paint. The person is wearing a dark blue jacket and a matching beret, and is seated on a wooden chair. The canvas depicts a landscape with a body of water and mountains in the background. The person is carefully applying green paint to the canvas, adding details to the scene. ...



**TASK-EXTENSION:** A butterfly with black and orange wings approaches a hanging, brownish seed pod on a branch. The butterfly lands on the seed pod, causing it to sway slightly, then quickly flies away. The background is a blurred green, suggesting a forest or garden setting. The lighting is bright and natural, indicating daytime. The camera remains stationary, ...



**TASK-DEPTH:** A pig dressed as a chef is standing in a kitchen, holding a frying pan with a blue flame underneath. The pig is wearing a white chef's hat and apron, and it is stirring shredded yellow food in the pan. The background shows a modern kitchen with stainless steel appliances, a sink, and various kitchen utensils and containers on the counter. The lighting is bright and natural, ...



**TASK-POSE:** A young woman with curly hair and wearing a white shirt is standing against a yellow background. She is smiling and looking at the camera while holding her sunglasses up to her forehead with her right hand. The woman has dark skin, and her hair is styled in loose curls that fall around her shoulders. She is wearing large, round sunglasses with a gold frame and red lenses. ...

Figure 22: The visualization results of the proposed VACE.

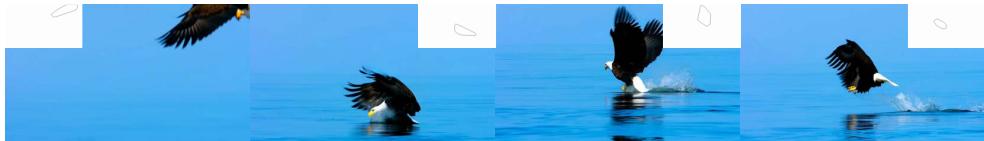
Before tokenization, pixel-level context frames and masks are preprocessed using a concept decoupling strategy and then encoded into the latent space. As shown in Fig. 21 (b), the concept decoupling strategy explicitly separates the data of different modalities and distributions based on masks into two frame sequences of identical shape:  $F_c = F \times M$  and  $F_k = F \times (1 - M)$ , where  $F_c$  refers to reactive frames containing all pixels to be modified, while  $F_k$  denotes inactive frames that retain all pixels to be preserved, named inactive frames. This mechanism ensures clear task definitions and guarantees the model's convergence across different tasks.  $F_c, F_k$  are processed by Wan-VAE and mapped into the same latent space of  $X$ , maintaining their spatio-temporal consistency. To avoid



**TASK-GRAY:** A young girl with long, curly hair is lying on a bed of lilac flowers and sheer fabric. She is wearing a pink dress with intricate lace details. The girl reaches up to touch the flowers above her, smiling and looking content. The background is filled with more lilacs and green leaves, creating a dreamy atmosphere. The camera angle is from above, capturing the girl and the surrounding flowers in a soft ...



**TASK-SCRIBBLE:** A person wearing a light blue shirt is gently petting a tabby cat lying on a white table. The cat, adorned with a white garment featuring cartoon characters, appears relaxed and content as the person strokes its head and body. The background is plain and light-colored, keeping the focus on the interaction between the person and the cat. The camera remains stationary, ...



**TASK-LAYOUT:** An eagle is flying over a calm blue ocean under a clear sky. The eagle, with its brown and white feathers and yellow beak, descends towards the water, its wings spread wide. As it approaches the surface, it dives into the water, creating a splash, and emerges with a fish in its talons. The eagle then takes off again, flying away from the camera with the fish clutched tightly. ...



**TASK-OBJECT:** A vibrantly colored Chinese lion dance costume stands prominently against a rich red background, exuding traditional cultural significance and festivity. The costume features elaborate details, with yellow fur adorning its edges and a complex pattern of green, red, and gold accents. Adornments such as large, expressive eyes, a wide mouth with a toothy grin, ...



**TASK-FACE:** A man is sitting at a table, playing chess. He is holding a chess piece in his right hand and appears to be contemplating his next move. The man has curly hair and a beard, and he is wearing a black sweater over a white collared shirt. The chessboard is in front of him, with several pieces still on the board. There are also a few bottles on the table. The background is plain and neutral. ...

Figure 23: The visualization results of the proposed VACE.

any mishmash of images and videos, reference images are separately encoded by the Wan-VAE encoder and concatenated back along the temporal dimension, while the corresponding parts need to

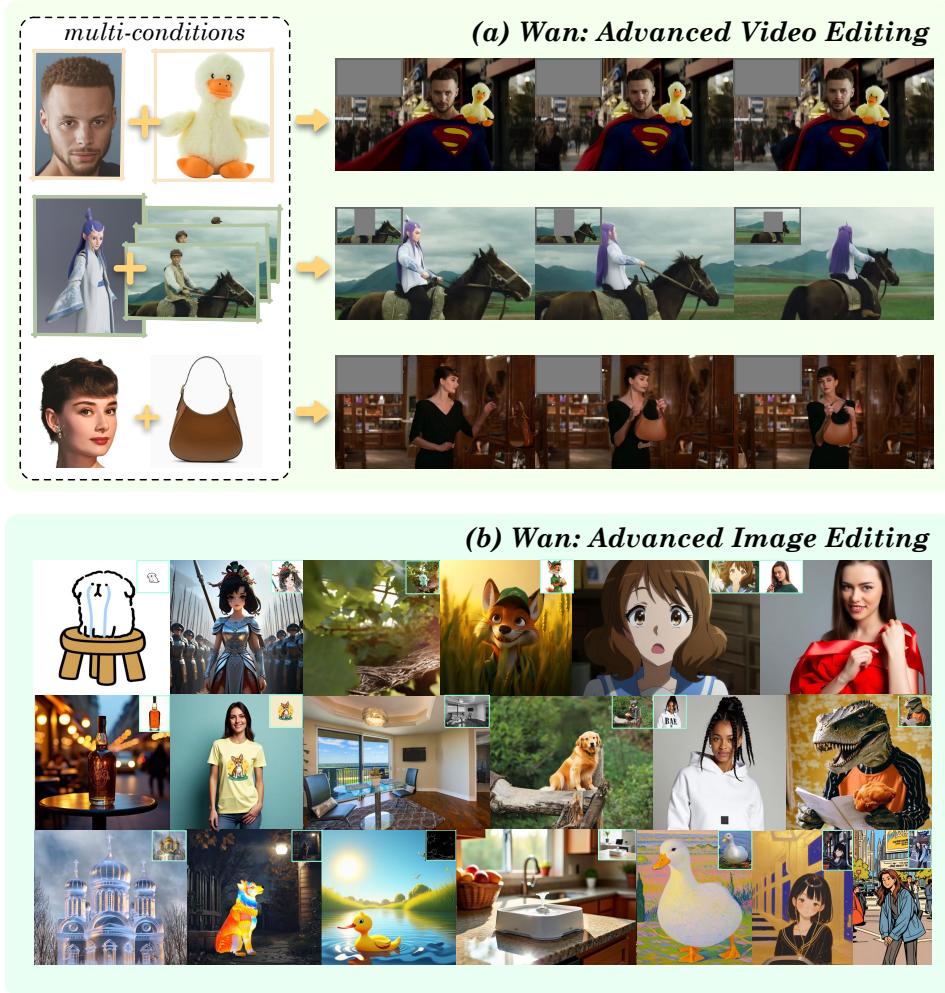


Figure 24: The visualization results of advanced video/image editing.

be removed during decoding.  $M$  is directly reshaped and interpolated. After that,  $F_c$ ,  $F_k$ , and  $M$  are mapped into latent spaces and spatio-temporal aligned with  $X$  in the shape of  $n' \times h' \times w'$ .

### 5.2.2 DATASETS AND IMPLEMENTATION

**Data construction.** To develop an all-in-one model, the diversity and complexity of data construction must increase. For controllable video generation and editing tasks, input modalities are expanded to include target videos, source videos, local masks, references, and more. Efficient and rapid data acquisition for various tasks necessitates maintaining video quality while performing instance-level analysis and understanding. We begin by performing shot slicing on the video data and initially filtering based on resolution, aesthetic score, and motion amplitude. The first frame is then labeled using RAM (Zhang et al., 2023d) and processed with Grounding DINO (Liu et al., 2023d) for detection, enabling secondary filtering of videos with target areas that are either too small or too large. Additionally, we utilize the propagation operation of SAM2 (Ravi et al., 2025) for video segmentation to obtain instance-level information throughout the video. Based on the segmentation results, we further filter instances temporally by calculating the effective frame ratio according to a mask area threshold. The construction for different tasks must be tailored to each task’s specific characteristics. For detailed information, refer to the work (Jiang et al., 2025).

**Implementation details.** We train the controllable and editing model, pre-trained on Wan-T2V-14B, to support resolutions up to 720p through a multi-stage training process. Initially, we focus on foundational tasks such as inpainting and extension to complement the pre-trained text-to-video

models. This phase incorporates masks and develops contextual generation capabilities in both spatial and temporal dimensions. Subsequently, we expand the task by transitioning from single-input to multiple-input reference frames and from individual to composite tasks. Finally, we enhance the model’s quality by fine-tuning it with higher-quality data and longer sequences.

### 5.2.3 EVALUATION

In Fig. 22 and Fig. 23, we present the results of the Wan single model across various tasks. It is evident that the model achieves a high level of performance in video quality and temporal consistency. Additionally, we demonstrate new applications resulting from the combination of different generative capabilities, as illustrated in Fig. 24 (a). Our model performs effectively on these tasks, showcasing significant potential for capability expansion. The unified controllable generation and editing framework proposed by VACE is also applicable to image generation and editing, as shown in Fig. 24 (b).

## 5.3 TEXT-TO-IMAGE GENERATION

Wan is jointly trained on both image and video datasets, equipping it with not only advanced video generation capabilities but also exceptional image synthesis performance. This dual-training strategy facilitates cross-modal knowledge transfer, creating a highly versatile framework that achieves powerful results in both domains. In practice, our model has been trained on an image dataset nearly ten times larger than its video counterpart, allowing for synergistic benefits between image and video generation tasks. This extensive training regimen has led to outstanding performance in various aspects of image synthesis. As demonstrated in Fig. 25, Wan generates high-fidelity images across diverse categories, encompassing artistic text-based visuals, photorealistic portraits, imaginative creative designs, and professional-grade product photography.

## 5.4 VIDEO PERSONALIZATION

Video personalization aims to generate videos that maintain consistent identity with the user-provided reference (Wei et al., 2024a; Li et al., 2024; Yuan et al., 2025). In this section, we integrate advanced personalization techniques into the video generation pipeline, achieving state-of-the-art performance. The subsequent discussion provides a comprehensive elaboration of our approach, encompassing the model architecture, personalization data, and experimental validation.

### 5.4.1 MODEL DESIGN

The core techniques behind video personalization tackle two primary challenges: (1) the acquisition of high-fidelity personalized identities and (2) the seamless integration of these identity features into the video generation pipeline.

We start from our Wan-T2V foundation model by first obtaining personalized identity information. Existing video personalization approaches usually rely on ID extractors (*e.g.*, ArcFace (Deng et al., 2019)) or versatile visual extractors (*e.g.*, CLIP (Radford et al., 2021)) to obtain identity information. While they make promising progress, their performance is bottlenecked by the limitations of the feature extractors. For instance, ID extractors mainly focus on features for face recognition but may not capture other critical visual cues such as scars or stickers. Moreover, they are susceptible to low-resolution faces, illumination variations, and partial occlusions (such as glasses, masks, or hair). On the other hand, versatile visual extractors are not specifically tailored to the face domain and tend to focus on coarse-grained semantic information. Therefore, we choose not to rely on any feature extractor to avoid information loss, and our generation process is directly conditioned on the input face image in the latent space of Wan-VAE.

There are many ways to inject the personalized identity information into the video generation process. In our experiments, we found that cross-attention operation is suited to interact with dense representations obtained from identity extractors, while self-attention is more appropriate for modeling data in the same latent space. Fig. 26 illustrates the design of our video personalization approach, which follows a self-attention paradigm. Specifically, in the latent space, we first extend  $K$  additional frames prepended to the given video using the segmented face images. These face images are extracted from the paired video using face detection and segmentation, and the facial landmarks



Figure 25: Image samples generated from Wan model.



Figure 26: The core design of the video personalization approach in the latent space of Wan-VAE.

are further aligned to a black canvas that has the same size as the video frame. Then, along the channel dimension of this extended video, we concatenate the face images with all-ones masks for the first  $K$  frames, and blank images with all-zeros masks for the remaining frames. These concatenated signals serve as the conditions. Finally, the diffusion process is performed on this temporally extended video, conditioned on the channel-wise condition signals in an inpainting fashion. Note that no other modifications are applied to our Wan-T2V model.

During training, we randomly drop a portion of face images in the  $K$  extended frames, so as to support 0 to  $K$  reference face(s) video generation. In the inference stage, starting from the pure noise of the extended video, we concatenate the user-provided face images (no more than  $K$ ) to the extended frames in the channel dimension and set the corresponding masks as all-ones. The goal is to reconstruct the face images in the first  $K$  frames, and synthesize a new personalized video in the subsequent frames that preserves the provided identity.

#### 5.4.2 DATASET

We curate a collection of personalization data from our T2V foundation model’s training dataset through careful filtering and automatic data synthesizing. We first filter out about  $\mathcal{O}(100)$ M videos using our internal human classifiers and perform face detection at 1 FPS on these videos. A video shall be discarded if more than one face is detected in any frame or if over 10% of the frames have no face detected. Then we calculate ArcFace similarities between consecutive frames and discard videos with low similarities. Afterwards, we perform face segmentation to remove the distracting backgrounds, and perform face landmark detection so as to facilitate canvas alignment during our training. Note that we do not filter out faces with the small areas, because we find that such videos usually contain full-body human figures. Finally, we construct about  $\mathcal{O}(10)$ M personalized videos, where each video is accompanied by 5 segmented faces in average.

We also resort to automatic data synthesizing to improve facial diversity. Particularly, we randomly select  $\mathcal{O}(1)$ M personalized videos from above, and adopt Instant-ID (Wang et al., 2024b) to synthesize diverse faces for each video. Notably, we build a text template with over 100 prompts, including anime, line art, cinematic, Minecraft, and so on. Every time we take a random prompt from this template, as well as a random human pose estimated from the rest of the videos, to serve as inputs to Instant-ID. We further measure the ArcFace similarity on the generated faces and filter out those with lower similarities. In total, we gather about  $\mathcal{O}(1)$ M videos with synthesized faces, which greatly improve the diversity of styles, poses, illuminations, and occlusions on our personalization dataset.

	Wan	CN-TopA	CN-TopB	CN-TopC
Arcface Similarity	0.5526	0.5655	0.5197	0.4998

Table 8: Comparison of other leading video personalization methods.

#### 5.4.3 EVALUATION

We present our video personalization results in Fig. 27, where the left part represents the input identities and the right part represents the synthesized personalized videos. All the testing images are

randomly picked from Pexels<sup>1</sup>. We also evaluate our model on an unseen evaluation set and measure the ArcFace similarity between input faces and output personalized videos by sampling faces at 1 FPS from videos. The final similarity scores are averaged from these sampled faces. The corresponding results are illustrated in Table 8, where our approach demonstrates competitive video personalization performance compared to other leading commercial and closed-source Chinese competitors.

## 5.5 CAMERA MOTION CONTROLLABILITY

The camera motion control module is designed to accurately match the motion and viewpoint of the video by leveraging camera trajectory. Specifically, we utilize the extrinsic parameters  $[R, t] \in \mathbb{R}^{3 \times 4}$  and intrinsic parameters  $K_f \in \mathbb{R}^{3 \times 3}$  for each frame. Our approach consists of two main components to effectively inject camera motion conditions: camera pose encoder and camera pose adapter, as illustrated in Fig. 28.

**Camera pose encoder.** First, for each pixel in each video frame, we use Plücker coordinates to transform the given extrinsic and intrinsic parameters into a sequence of fine-grained positions  $P \in \mathbb{R}^{6 \times F \times H \times W}$ . To adjust the compression resolution of video latent features, we apply the PixelUnshuffle operation in Pytorch (Paszke et al., 2019) to reduce the spatial resolution of  $P$  while increasing the number of channels. Finally, we encode the output using a series of convolutional modules aligned with the number of DiT blocks to extract multi-level camera motion features.

**Camera pose adapter.** To integrate camera motion features into video latent features, we adopt an adaptive normalization layer. Specifically, we transform the input camera motion feature sequence into scaling factors  $\gamma_i$  and shifting parameters  $\beta_i$  using two zero-initialized convolution layers. Then,  $\gamma_i$  and  $\beta_i$  are incorporated into each DiT block through a simple linear projection defined by the formula  $f_i = (\gamma_i + 1) * f_{i-1} + \beta_i$ , where  $f_i$  represents the video latent features at layer  $i$ .

Regarding the training data, we utilize an advanced camera pose estimation algorithm named VGG-SfM (Wang et al., 2024a) to extract camera trajectories from our training videos that exhibit significant camera motion. This process yields approximately  $\mathcal{O}(1)$  thousand video segments. We train our module within our text-to-video generation framework using the Adam optimizer. Fig. 29 presents various example results of videos generated under camera motion guidance.

## 5.6 REAL-TIME VIDEO GENERATION

Current video generative methods often require significant computational resources and considerable time to produce even short video clips. Systems relying on high-end hardware remain slow, often taking several minutes to generate just a few seconds of video. While the visual quality of these generated videos has seen substantial improvements in recent years, the extended processing times present a major bottleneck in practical applications and iterative design workflows. In fields where rapid prototyping and real-time adjustments are crucial—such as interactive entertainment, virtual reality, and video production—these long generation times limit creative flexibility and slow down the process of refining visual content.

Moreover, the slow speed of current methods limits their use in dynamic environments where real-time feedback is essential. In contexts like live-streaming or gaming, there's an increasing demand for systems that can instantly render synthetic scenes and characters in response to user actions or changing inputs. Any delay in generating high-quality video clips can result in a suboptimal user experience, reducing immersion and interactivity. By focusing on real-time generation, Wan seeks to enhance both the performance and applicability of video generation techniques, broadening their potential in fast-paced, interactive environments. Fig. 30 and 31 provide such cases of real-time long video stream generation. We then introduce how to achieve this using our Wan models.

### 5.6.1 METHOD

To construct the real-time generation pipeline, we build upon a previously pre-trained version of the Wan model. This design choice offers two primary benefits. First, starting from a well-trained model significantly accelerates convergence and improves training stability, as the model already possesses a strong initialization. Second, the pre-trained model has already captured valuable knowledge about

---

<sup>1</sup>[www.pexels.com](http://www.pexels.com)



Figure 27: Visualization of our video personalization approach.

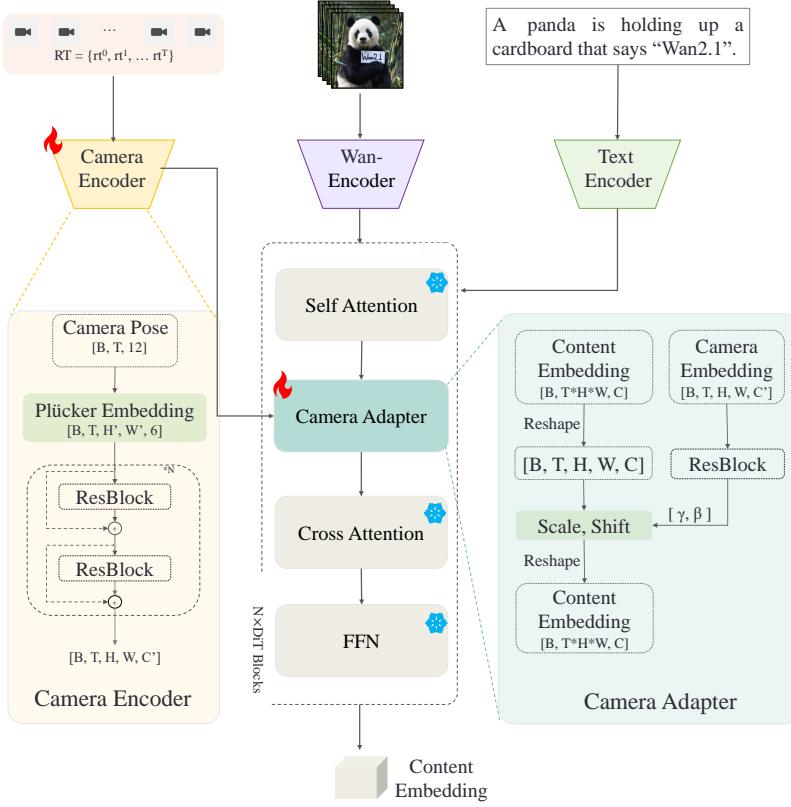


Figure 28: Framework of video generation guided by camera motion.

motion patterns and continuous temporal dynamics, which can be directly inherited by the new pipeline to enhance the smoothness and coherence of generated content. Therefore, we adopt the pre-trained `Wan` model as the foundation for our real-time pipeline <sup>2</sup>.

The pre-trained `Wan` model is designed to generate fixed-length video clips, typically 5 to 10 seconds long. To transform this into a real-time streaming model capable of generating continuous video streams with no predefined length (potentially infinite), two key modifications are required:

- **Streaming pipeline adaptation.** We replace the static generation process, which produces a fixed-length sequence of video tokens in a single pass, with a streaming mechanism that incrementally generates video tokens. In this streaming setup, video tokens are processed through a denoising queue — each time the oldest token is generated and dequeued, a new token is added to the bottom of the queue, enabling continuous generation without length constraints.
- **Real-time acceleration.** In addition to enabling streaming, we optimize the generation speed to ensure the system meets real-time performance requirements, *i.e.*, the pipeline must generate new frames fast enough to keep up with real-time playback.

In the following sections, we will describe each of these modifications in detail, including their design rationale, technical implementation, and empirical performance.

### 5.6.2 STREAMING VIDEO GENERATION

Traditional Diffusion Transformer (DiT) (Peebles & Xie, 2023) models are constrained in their ability to generate videos longer than a few seconds, even when employing significant spatial and

<sup>2</sup>Part of this section is introduced in a previously released tech report *The Matrix* (Feng et al., 2024).

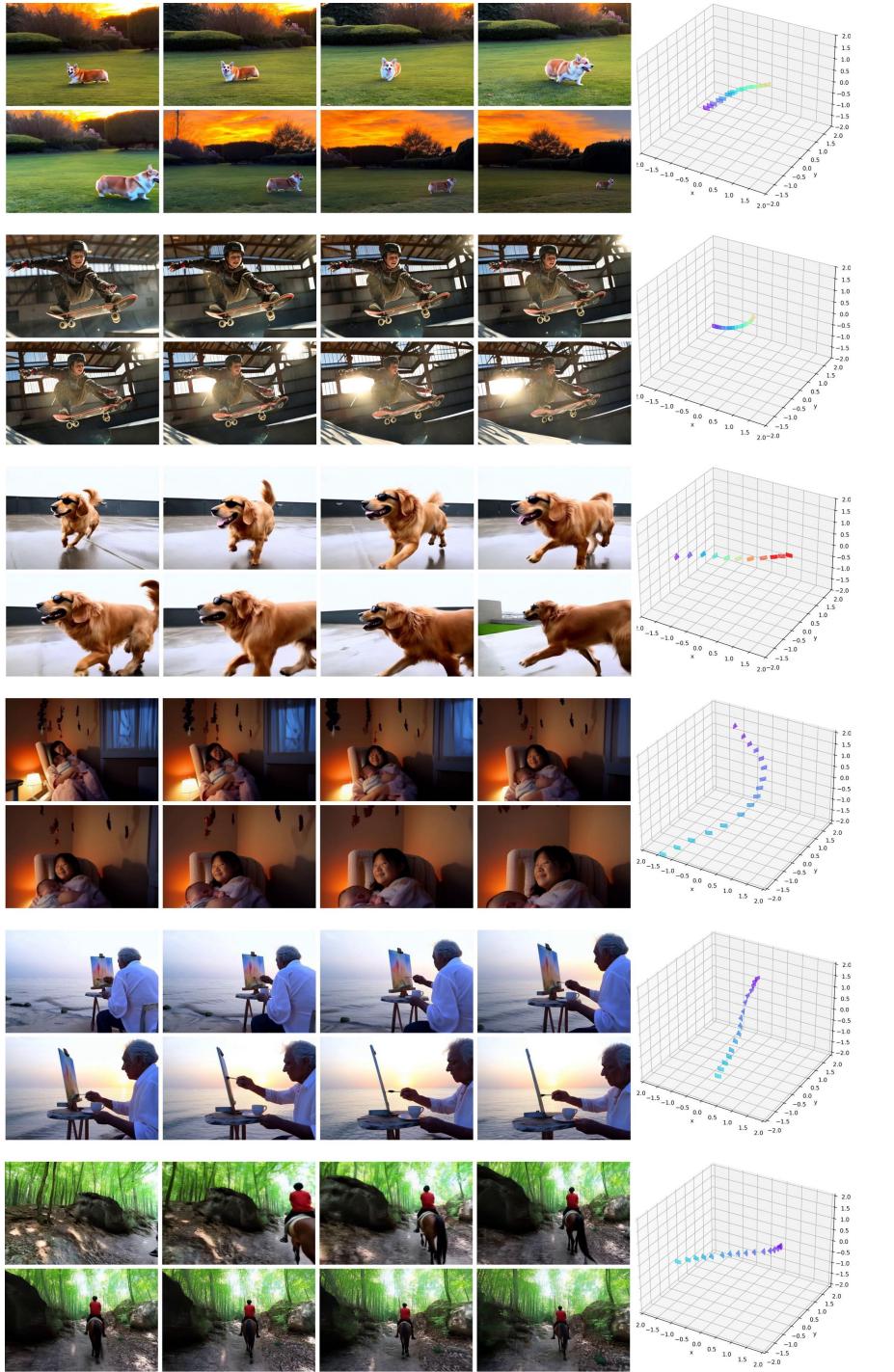


Figure 29: Generated video results guided by camera motion.

temporal compression through Variational Autoencoders (VAEs) (Kingma, 2013). This limitation stems primarily from the computational and memory-intensive nature of attention mechanisms over extended temporal sequences. To overcome this challenge, we introduce Streamer, a novel approach that leverages a sliding temporal window to manage temporal dependencies efficiently, enabling the generation of long or even infinite-length videos in a streaming fashion.



Figure 30: Generated 15 minutes video with 8 A100 GPUs in real-time 8 FPS.

**Shift window denoise process models.** The key innovation of Streamer lies in its assumption that temporal dependencies are confined within a limited time window. By focusing attention computations within this window, Streamer significantly reduces computational overhead while maintaining video continuity. Specifically, Streamer processes video tokens in a queue, denoising them over multiple



Figure 31: Generated videos with a single RTX 4090 GPU, int8 and TensorRT quantization in real-time 20 FPS.

noise levels within the window. After a fixed number of denoising steps, the leftmost token (with the lowest noise level) is dequeued and cached, while a new token with Gaussian noise is appended to the rightmost position. This sliding mechanism ensures that the window maintains a consistent length while facilitating the generation of continuous video frames.

**Training and inference.** Streamer is fine-tuned from a pre-trained DiT model. During training, a sequence of  $2w$  video tokens is sampled, where  $w$  is the size of the sliding window (typically set to  $w = T$ , the number of diffusion solver steps). The first  $w$  tokens are used solely for ‘warming up’ the model and do not contribute to the loss computation. The loss is calculated only on the last  $w$  tokens, ensuring that the model learns to generate coherent video frames within the window. At inference time, the same warmup strategy is employed: the first  $w$  tokens are discarded, and the generated video begins from the  $(w + 1)$ -th token.

**Ensuring continuity during training and inference.** To maintain smooth transitions between consecutive windows, cached tokens are re-introduced into the token queue at a noise level of 0. This allows previously generated tokens to participate in the denoising process of subsequent windows, ensuring temporal consistency and continuity in the generated video.

Streamer represents a significant advancement in video generation by adapting DiT models for streaming applications. Its sliding temporal window approach addresses the limitations of traditional DiT models, enabling the efficient generation of infinite-length videos while maintaining high quality and temporal coherence. This method opens new possibilities for real-time video generation and long-form content creation. We summarize its advantages as follows:

- Infinite-Length Video Generation: By leveraging a sliding window mechanism, Streamer can generate videos of arbitrary length without incurring prohibitive computational costs;
- Efficient Attention Computation: Limiting attention computations to a confined temporal window reduces memory and processing demands;

- Seamless Continuity: The caching and re-introduction of tokens ensure smooth transitions between windows, preserving video quality over extended durations.

### 5.6.3 CONSISTENCY MODEL DISTILLATION

After successfully extending the DiT model to Streamer for infinite-length video generation, the next critical step is to address the challenge of achieving real-time rendering of the simulated world. While Streamer enables the generation of long or even infinite videos by efficiently managing temporal dependencies within a sliding window, the computational demands of the diffusion process still pose a bottleneck for real-time applications. To overcome this, we propose a novel integration of Streamer with Consistency Models, a cutting-edge approach for accelerating diffusion-based generative processes. Specifically, we leverage the Latent Consistency Model (LCM ([Luo et al., 2023](#))) and its video version (VideoLCM ([Wang et al., 2023c](#))), which distill the original diffusion process and class-free guidance into a highly efficient four-step consistency model. This distillation process significantly reduces the number of sampling steps required for high-quality generation while maintaining the temporal coherence and streaming capabilities inherent to Streamer.

The integration of LCM with Streamer is designed to preserve the denoising window mechanism, ensuring that the sliding temporal window continues to manage dependencies effectively. During training, we optimize this combined framework to balance efficiency and quality. The result is a  $10 - 20 \times$  acceleration in inference speed, enabling the model to achieve a rendering rate of 8 - 16 FPS. This dramatic improvement in performance makes the system viable for real-time applications, such as interactive simulations, live video synthesis, and dynamic content generation in virtual environments. By combining the strengths of Streamer—its ability to generate infinite-length videos with temporal coherence—and LCM—its efficiency in accelerating the diffusion process—we bridge the gap between high-quality video generation and real-time responsiveness. This advancement not only enhances the practicality of diffusion models for streaming applications but also opens up new possibilities for immersive, interactive, and real-time media creation.

**Quantization for running in customer-level devices.** While the generating speed has been accelerated to real-time levels, deploying the model on consumer-level devices remains challenging due to the high computational and memory requirements, even for high-end GPUs like the NVIDIA 4090. To address this, we introduce quantization techniques to optimize the model for efficient deployment. Specifically, we employ two distinct quantization strategies: int8 quantization ([torchao maintainers & contributors, 2024](#)) for attention layers and linear heads, and TensorRT quantization ([NVIDIA, 2023](#)) for the overall model. The int8 quantization approach significantly reduces memory consumption by converting weights and activations to 8-bit integers, while maintaining the overall generation quality. However, this method provides minimal acceleration in generating speed, as it primarily focuses on memory efficiency rather than computational optimization. On the other hand, TensorRT quantization offers a more comprehensive solution, enabling substantial acceleration in generation speed. This technique allows the model to achieve real-time performance of 8 FPS even on a single 4090 GPU, making it feasible for consumer-level devices. However, TensorRT quantization introduces a modest increase in network error, which can be observed even when its built-in error-checking mechanisms are enabled. This error manifests as slight deviations in output quality, such as minor artifacts or inconsistencies in the generated video. In some cases, it may also lead to measurable instability, such as flickering or temporal incoherence. Despite these trade-offs, the combination of int8 and TensorRT quantization provides a balanced approach to optimizing the model for real-time, consumer-level deployment, ensuring both efficiency and practicality while maintaining acceptable generation quality. By carefully tuning the quantization parameters and leveraging TensorRT’s error-checking features, we mitigate these issues to the greatest extent possible, enabling robust and efficient performance on consumer-grade hardware.

## 5.7 AUDIO GENERATION

The primary objective of audio generation in this work is to produce synchronized soundtracks for video clips, formulated as a video-to-audio (V2A) generation framework. The generated soundtracks consist of ambient sound and background music, explicitly excluding speech or vocal elements. In contrast to text-to-audio models ([Liu et al., 2023b; 2024c](#)), the ambient sound generated by video-to-audio models must be temporally aligned with the visual content of the video, while the

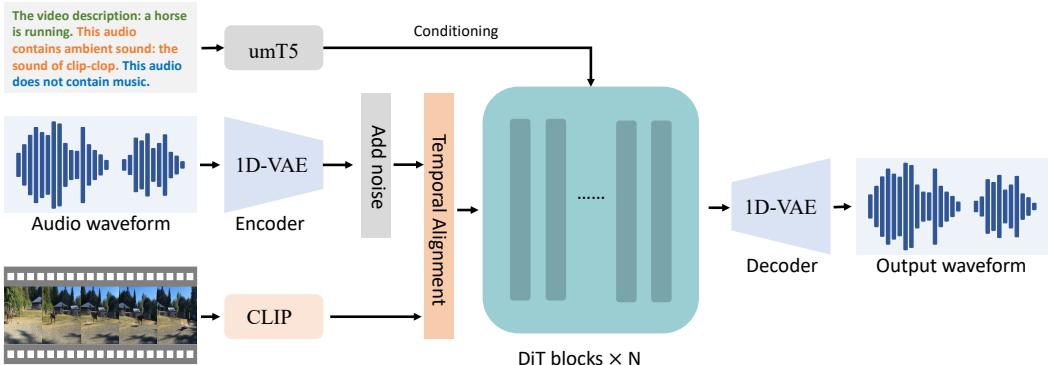


Figure 32: Framework of video-to-audio generation. Our model processes both a video chunk and its associated textual description as dual inputs to synthesize high-quality, semantically coherent audio.

accompanying music should accurately reflect the emotional tone and contextual setting of the video, as would naturally be expected. Furthermore, to enhance user control over sound design, our framework integrates both video and textual prompts, enabling users to specify on-screen or off-screen sounds and define the style and presence of background music.

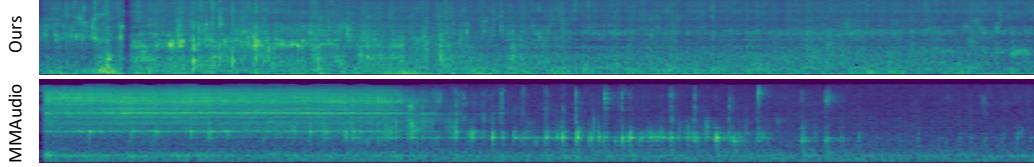
#### 5.7.1 MODEL DESIGN

Consistent with our video generation framework, our video-to-audio model also utilizes a diffusion transformer (DiT) (Peebles & Xie, 2023) based on the flow-matching (Lipman et al., 2022) to model the denoising diffusion process in the audio domain. The detailed pipeline of our V2A model is illustrated in Fig. 32.

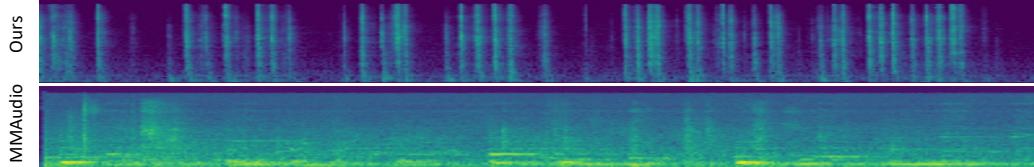
**Audio autoencoder.** A widely adopted audio compression approach involves transforming raw waveforms into mel-spectrograms, followed by encoding them into  $H^a \times W^a \times C^a$  latent features using an image-based variational autoencoder, as demonstrated in (Liu et al., 2023b; 2024c). However, in the context of a diffusion transformer backbone, these latent features must be partitioned into patches and reshaped into a tensor of size  $(H^a \cdot W^a) \times C^a$ . This patching-and-reshaping process can disrupt the temporal alignment with the corresponding video content, posing a significant challenge for synchronization. Considering this limitation, we train a compression model that operates directly on raw waveforms. Our 1D-VAE generates latent features of size  $T^a \times C^a$ , where  $T^a$  represents the sequence length along the time axis, preserving explicit temporal information essential for accurate synchronization.

**Video and text encoder.** To achieve frame-accurate synchronization between synthesized audio and visual sequences, we establish temporal coherence through multimodal feature fusion. Our architecture first extracts frame-wise visual embeddings using a CLIP Radford et al. (2021) model, then performs temporal rate adaptation through feature replication to match the audio feature’s sampling rate, following (Polyak et al.). The synchronized visual and acoustic features undergo dimension transformation via linear projection layers to align with DiT’s latent space, followed by element-wise summation for multimodal fusion. For linguistic understanding across languages, we leverage the pre-trained umT5 model (Chung et al., 2023) as our text encoder, capitalizing on its cross-lingual representation capabilities through frozen parameters.

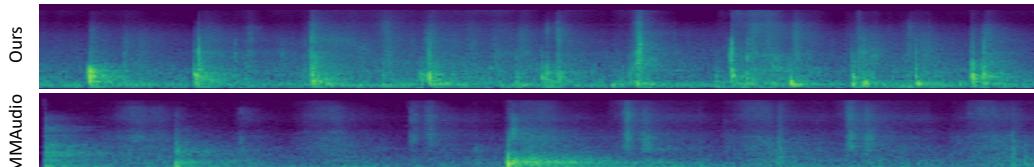
**Data.** The training data for our V2A model is derived from the video generation dataset through a rigorous filtering process. We systematically remove videos lacking soundtracks or containing speech/vocal music, resulting in a refined subset of  $\mathcal{O}(1)$  thousand hours. To enhance multimodal representation, we employ a comprehensive captioning strategy: dense video descriptions are complemented with audio-specific captions generated by Qwen2-audio (Chu et al., 2024). These audio captions are categorized into ambient sounds and musical compositions, with the latter characterized by style, rhythm, melody, and instrumentation. The final structured caption integrates three components: (1) dense video description, (2) ambient sound characterization, and (3) background music analysis, providing a unified multimodal representation for training.



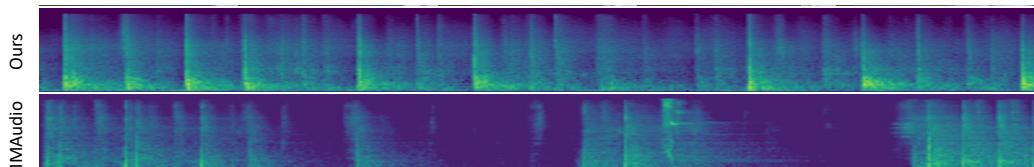
A person holds a coffee pot above a cup and slowly pours the steaming coffee into it.



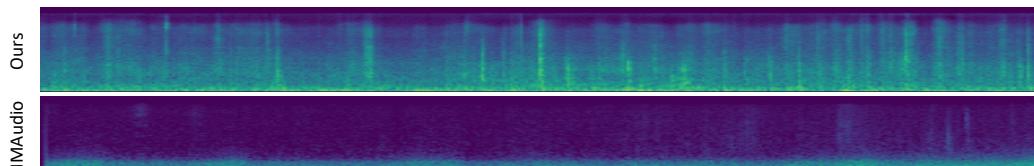
A person sits alone in a crowded café, typing on a laptop, with people chatting softly and moving quietly in the background.



A nurse walked quietly down a dimly lit hospital corridor, passing by row upon row of empty rooms.



A solitary horse gallops across the vast plain, its thundering hooves pounding the earth and sending sharp snorts of wind through the air. Its mane and tail flutter in the breeze.



A person is using an ice axe to tap on a large chunk of ice on a smooth surface, exposing its crystalline structure.

Figure 33: Audio generation samples of MMAudio and ours.

**Implementation details.** Our V2A architecture generates high-fidelity stereo audio with a maximum duration of 12 seconds at a 44.1 kHz sampling rate. The model processes multimodal inputs through dedicated encoders: umT5-XXL produces 4096-dimensional text embeddings, while the CLIP extracts 1024-dimensional visual embeddings per frame. The audio and video features are projected into a unified 1536-dimensional latent space within the DiT backbone. For temporal alignment, we downsample input videos and yield precisely 48 frames for 12-second clips, ensuring frame-accurate synchronization between visual and audio modalities. The system processes latent sequences of length 256 in one batch. To enhance the model’s capability of generating audio solely from visual cues, we implement a random masking strategy during training where ambient sound and music captions are selectively omitted with a predefined probability. This conditioning mechanism forces the model to establish robust cross-modal associations between visual content and corresponding audio patterns, while maintaining the flexibility to utilize textual descriptions when available.

#### 5.7.2 EVALUATION

We provide qualitative evaluations of audio generation in Fig. 33, where we compare our method with the recently released open-sourced approach MMAudio (Cheng et al., 2024). This comparative analysis is particularly meaningful as MMAudio represents the most similar and competitive baseline to our method, demonstrating impressive performance in audio generation tasks. The videos are generated by our text-to-video model with a duration of 5 seconds.

Our V2A model demonstrates superior performance in several key aspects of audio generation, as evidenced by the comparative results. Specifically, the model exhibits enhanced long-term consistency, particularly noticeable in the ‘pouring coffee’ scenario (the first case in Fig. 33). In the second case of ‘typing’, our approach generates significantly cleaner audio output compared to the noisier results produced by MMAudio. Furthermore, our method excels in synthesizing rhythmic sound patterns, as demonstrated in the ‘walking’, ‘boxing’, and ‘clip-clop’ examples (the last three cases in Fig. 33), where it achieves more natural and coherent audio generation.

**Limitations.** Our method exhibits limitations in generating human vocal sounds, including but not limited to laughter, crying, and speech. This limitation is primarily attributed to our data preparation process, where speech-related vocal data is deliberately excluded from the training dataset. The MMAudio model’s ability to produce random speech-like sounds stems from its retention of speech-related data in its training corpus. In our future work, we plan to incorporate speech generation capabilities to address this limitation.

## 6 LIMITATION AND CONCLUSION

**Limitation.** This study presents Wan, a foundational video generation model that has achieved notable advancements across multiple benchmarks. Specifically, significant improvements have been observed in motion amplitude (*e.g.*, sports, dance) and instruction-following capabilities. Nevertheless, several limitations persist, which should be addressed to realize more potential of Wan. First, preserving fine-grained details in scenarios involving large motion remains a challenge for our method, as well as a broader issue within the field of video generation. Substantial efforts are required to improve the fidelity of videos with large-scale motion dynamics. Second, the computational cost associated with large-scale models remains prohibitive. Inference on a 14B model currently requires approximately 30 minutes on a single head-end GPU without additional optimization. To democratize video generation as a universally accessible AI tool, further research into efficiency and scalability is essential. Finally, there is still a lack of domain-specific expertise. As a foundational video model, we are committed to enhancing Wan’s general capabilities, yet performance in specific localized scenarios, such as education and medicine, may be insufficient. We aim to address this limitation by open-sourcing our latest models and fostering community-driven development across diverse specialized domains.

**Conclusion.** In this report, we publicly release Wan, our latest video model, which establishes a new benchmark for video generation. We provide a comprehensive overview of the architectural design of our Wan-VAE and DiT models, including detailed insights into their training approach, data curation processes, evaluation methods, and empirical results. Furthermore, we meticulously analyze our data preprocessing pipeline and the strategic adaptations implemented to optimize model training. This holistic approach is designed to drive advancements in the domain of video generation. We also extensively investigate downstream applications, such as image-to-video generation, video

editing, and personalized video generation, to demonstrate the versatility and practical utility of Wan across diverse scenarios. In addition to open-sourcing the 14B model, we explore the feasibility of leveraging smaller-scale models for efficient video generation. Notably, our 1.3B model not only achieves competitive performance compared to larger counterparts but also enables seamless inference on consumer-grade GPUs, significantly enhancing accessibility and practicality for content creators. Looking ahead, we plan to focus on scaling both data and model architectures to tackle the most pressing challenges in video generation. Our ongoing efforts aim to provide the research community with more robust and versatile tools for video creation, fostering innovation and broader adoption in this rapidly evolving field.

## 7 CONTRIBUTORS

Authors are listed **alphabetically by the first name.**

Ang Wang	Pandeng Li	Xin Xu
Baole Ai	Pingyu Wu	Yan Kou
Bin Wen	Ruihang Chu	Yangyu Lv
Chaojie Mao	Ruili Feng	Yifei Li
Chen-Wei Xie	Shiwei Zhang	Yijing Liu
Di Chen	Siyang Sun	Yiming Wang
Feiwu Yu	Tao Fang	Yingya Zhang
Haiming Zhao	Tianxing Wang	Yitong Huang
Jianxiao Yang	Tianyi Gui	Yong Li
Jianyuan Zeng	Tingyu Weng	You Wu
Jiayu Wang	Tong Shen	Yu Liu
Jingfeng Zhang	Wei Lin	Yulin Pan
Jingren Zhou	Wei Wang <sup>~1</sup>	Yun Zheng
Jinkai Wang	Wei Wang <sup>~2</sup>	Yuntao Hong
Jixuan Chen	Wenmeng Zhou	Yupeng Shi
Kai Zhu	Wente Wang	Yutong Feng
Kang Zhao	Wenting Shen	Zeyinzi Jiang
Keyu Yan	Wenyuan Yu	Zhen Han
Lianghua Huang	Xianzhong Shi	Zhi-Fan Wu
Mengyang Feng	Xiaoming Huang	Ziyu Liu
Ningyi Zhang		

## REFERENCES

- Runway AI. Stable Diffusion Inpainting Model Card, <https://huggingface.co/runwayml/stable-diffusion-inpainting>, 2022.
- Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-VL: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 2023.
- 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023.
- Fan Bao, Chendong Xiang, Gang Yue, Guande He, Hongzhou Zhu, Kaiwen Zheng, Min Zhao, Shilong Liu, Yaole Wang, and Jun Zhu. Vidu: a highly consistent, dynamic and skilled text-to-video generator with diffusion models. *arXiv preprint arXiv:2405.04233*, 2024.
- 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 (2023). URL <https://cdn.openai.com/papers/dall-e-3.pdf>, 2023.
- Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.
- Tim Brooks, Aleksander Holynski, and Alexei A. Efros. InstructPix2Pix: Learning To Follow Image Editing Instructions. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 18392–18402, 2023.
- Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv:2104.14294*, 2021.
- Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, et al. Pixart- $\alpha$ : Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*, 2023a.
- Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*, 2016.
- Xi Chen, Lianghua Huang, Yu Liu, Yujun Shen, Deli Zhao, and Hengshuang Zhao. AnyDoor: Zero-shot Object-level Image Customization. *arXiv preprint arXiv:2307.09481*, 2023b.
- Xi Chen, Zhifei Zhang, He Zhang, Yuqian Zhou, Soo Ye Kim, Qing Liu, Yijun Li, Jianming Zhang, Nanxuan Zhao, Yilin Wang, Hui Ding, Zhe Lin, and Hengshuang Zhao. UniReal: Universal Image Generation and Editing via Learning Real-world Dynamics. *arXiv preprint arXiv:2412.07774*, 2024.

Ho Kei Cheng, Masato Ishii, Akio Hayakawa, Takashi Shibuya, Alexander Schwing, and Yuki Mitsufuji. Taming multimodal joint training for high-quality video-to-audio synthesis. *arXiv preprint arXiv:2412.15322*, 2024.

Ruihang Chu, Enze Xie, Shentong Mo, Zhenguo Li, Matthias Nießner, Chi-Wing Fu, and Jiaya Jia. Diffcomplete: Diffusion-based generative 3d shape completion. *Adv. Neural Inform. Process. Syst.*, 2023.

Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, et al. Qwen2-audio technical report. *arXiv preprint arXiv:2407.10759*, 2024.

Hyung Won Chung, Noah Constant, Xavier Garcia, Adam Roberts, Yi Tay, Sharan Narang, and Orhan Firat. Unimax: Fairer and more effective language sampling for large-scale multilingual pretraining. *arXiv preprint arXiv:2304.09151*, 2023.

Google DeepMind. Veo 2. <https://deepmind.google/technologies/veo/veo-2/>, 2024.12.

Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 4690–4699, 2019.

Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks*, 2018.

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 12873–12883, 2021.

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 *Int. Conf. Mach. Learn.*, 2024.

Jiarui Fang and Shangchun Zhao. Usp: A unified sequence parallelism approach for long context generative ai. *arXiv preprint arXiv:2405.07719*, 2024.

Ruili Feng, Han Zhang, Zhantao Yang, Jie Xiao, Zhilei Shu, Zhiheng Liu, Andy Zheng, Yukun Huang, Yu Liu, and Hongyang Zhang. The matrix: Infinite-horizon world generation with real-time moving control. *arXiv preprint arXiv:2412.03568*, 2024.

Chaoyou Fu, Yuhai Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *arXiv preprint arXiv:2405.21075*, 2024.

GenmoTeam. Mochi 1. <https://github.com/genmoai/models>, 2024.

Xun Guo, Mingwu Zheng, Liang Hou, Yuan Gao, Yufan Deng, Pengfei Wan, Di Zhang, Yufan Liu, Weiming Hu, Zhengjun Zha, Haibin Huang, and Chongyang Ma. I2V-Adapter: A General Image-to-Video Adapter for Diffusion Models. In *SIGGRAPH*, pp. 1–12, 2024a.

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. *International Conference on Learning Representations*, 2024b.

Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, Poriya Panet, Sapir Weissbuch, Victor Kulikov, Yaki Bitterman, Zeev Melumian, and Ofir Bibi. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024.

Zhen Han, Zeyinzi Jiang, Yulin Pan, Jingfeng Zhang, Chaojie Mao, Chenwei Xie, Yu Liu, and Jingren Zhou. ACE: All-round Creator and Editor Following Instructions via Diffusion Transformer. In *Int. Conf. Learn. Represent.*, 2025.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Adv. Neural Inform. Process. Syst.*, 2017.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Adv. Neural Inform. Process. Syst.*, 33:6840–6851, 2020.

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

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, and Nattapol Chanpaisit. Vbench: Comprehensive benchmark suite for video generative models. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023.

Sam Ade Jacobs, Masahiro Tanaka, Chengming Zhang, Minjia Zhang, Shuaiwen Leon Song, Samyam Rajbhandari, and Yuxiong He. Deepspeed ulysses: System optimizations for enabling training of extreme long sequence transformer models. *arXiv preprint arXiv:2309.14509*, 2023.

Zeyinzi Jiang, Chaojie Mao, Ziyuan Huang, Ao Ma, Yiliang Lv, Yujun Shen, Deli Zhao, and Jingren Zhou. Res-Tuning: A Flexible and Efficient Tuning Paradigm via Unbinding Tuner from Backbone. In *Adv. Neural Inform. Process. Syst.*, 2023.

Zeyinzi Jiang, Chaojie Mao, Yulin Pan, Zhen Han, and Jingfeng Zhang. SCEdit: Efficient and Controllable Image Diffusion Generation via Skip Connection Editing. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 8995–9004, 2024.

- Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. Vace: All-in-one video creation and editing. *arXiv preprint arXiv:2503.07598*, 2025.
- Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. *arXiv preprint arXiv:2410.05954*, 2024.
- Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. Musiq: Multi-scale image quality transformer. *arXiv preprint arXiv:2108.05997*, 2021.
- Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuandvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.
- Vijay Korthikanti, Jared Casper, Sangkug Lym, Lawrence McAfee, Michael Anderesch, Mohammad Shoeybi, and Bryan Catanzaro. Reducing activation recomputation in large transformer models, 2022. *arXiv preprint arXiv:2205.05198*.
- Kuaishou. Kling ai. <https://klingai.kuaishou.com/>, 2024.06.
- PKU-Yuan Lab and Tuzhan AI etc. Open-sora-plan, April 2024.
- LAION-AI. aesthetic-predictor, 2022. URL <https://github.com/LAION-AI/aesthetic-predictor>.
- Hengjia Li, Haonan Qiu, Shiwei Zhang, Xiang Wang, Yujie Wei, Zekun Li, Yingya Zhang, Boxi Wu, and Deng Cai. PersonalVideo: High ID-Fidelity Video Customization without Dynamic and Semantic Degradation. *arXiv preprint arXiv:2411.17048*, 2024.
- Yanwei Li, Yuechen Zhang, Chengyao Wang, Zhisheng Zhong, Yixin Chen, Ruihang Chu, Shaoteng Liu, and Jiaya Jia. Mini-gemini: Mining the potential of multi-modality vision language models. *arXiv preprint arXiv:2403.18814*, 2023.
- Bin Lin, Yunyang Ge, Xinhua Cheng, Zongjian Li, Bin Zhu, Shaodong Wang, Xianyi He, Yang Ye, Shanghai Yuan, Liuhan Chen, et al. Open-sora plan: Open-source large video generation model. *arXiv preprint arXiv:2412.00131*, 2024.
- Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.
- Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024a.

- Feng Liu, Shiwei Zhang, Xiaofeng Wang, Yujie Wei, Haonan Qiu, Yuzhong Zhao, Yingya Zhang, Qixiang Ye, and Fang Wan. Timestep embedding tells: It's time to cache for video diffusion model. *arXiv preprint arXiv:2411.19108*, 2024b.
- Hao Liu, Matei Zaharia, and Pieter Abbeel. Ring attention with blockwise transformers for near-infinite context. *arXiv preprint arXiv:2310.01889*, 2023a.
- Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D Plumbley. Audioldm: Text-to-audio generation with latent diffusion models. *arXiv preprint arXiv:2301.12503*, 2023b.
- Haohe Liu, Yi Yuan, Xubo Liu, Xinhao Mei, Qiuqiang Kong, Qiao Tian, Yuping Wang, Wenwu Wang, Yuxuan Wang, and Mark D Plumbley. Audioldm 2: Learning holistic audio generation with self-supervised pretraining. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024c.
- Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023c.
- Lijie Liu, Tianxiang Ma, Bingchuan Li, Zhuowei Chen, Jiawei Liu, Qian He, and Xinglong Wu. Phantom: Subject-consistent video generation via cross-modal alignment. *arXiv preprint arXiv:2502.11079*, 2025a.
- Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection. *arXiv preprint arXiv:2303.05499*, 2023d.
- Zhihang Liu, Chen-Wei Xie, Bin Wen, Feiwu Yu, Jixuan Chen, Boqiang Zhang, Nianzu Yang, Pandeng Li, Yun Zheng, and Hongtao Xie. What is a good caption? a comprehensive visual caption benchmark for evaluating both correctness and coverage of mllms. *arXiv preprint arXiv:2502.14914*, 2025b.
- Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- LumaLabs. Dream machine. <https://lumalabs.ai/dream-machine>, 2024.06.
- Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023.
- Zhengyao Lv, Chenyang Si, Junhao Song, Zhenyu Yang, Yu Qiao, Ziwei Liu, and Kwan-Yee K. Wong. Fastercache: Training-free video diffusion model acceleration with high quality. *arXiv preprint arXiv:2410.19355*, 2024.
- Guoqing Ma, Haoyang Huang, Kun Yan, Liangyu Chen, Nan Duan, Shengming Yin, Changyi Wan, Ranchen Ming, Xiaoniu Song, Xing Chen, et al. Step-video-t2v technical report: The practice, challenges, and future of video foundation model. *arXiv preprint arXiv:2502.10248*, 2025.

Xin Ma, Yaohui Wang, Gengyun Jia, Xinyuan Chen, Ziwei Liu, Yuan-Fang Li, Cun-jian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *arXiv preprint arXiv:2401.03048*, 2024.

Chaojie Mao, Jingfeng Zhang, Yulin Pan, Zeyinzi Jiang, Zhen Han, Yu Liu, and Jingren Zhou. ACE++: Instruction-Based Image Creation and Editing via Context-Aware Content Filling. *arXiv preprint arXiv:2501.02487*, 2025.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations. In *Int. Conf. Learn. Represent.*, 2021.

MiniMax. Hailuo ai. <https://hailuoai.com/video>, 2024.09.

NVIDIA. Nvidia tensorrt: Programmable inference accelerator, 2023. Version 8.6.

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

Yulin Pan, Chaojie Mao, Zeyinzi Jiang, Zhen Han, and Jingfeng Zhang. Locate, Assign, Refine: Taming Customized Image Inpainting with Text-Subject Guidance. *arXiv preprint arXiv:2403.19534*, 2024.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural Inform. Process. Syst.*, 32, 2019.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Int. Conf. Comput. Vis.*, pp. 4195–4205, 2023.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Assoc. Adv. Artif. Intell.*, volume 32, 2018.

PikaLabs. Pika 1.5. <https://pika.art/>, 2024.10.

Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, David Yan, Dhruv Choudhary, Dingkang Wang, Geet Sethi, Guan Pang, Haoyu Ma, Ishan Misra, Ji Hou, Jialiang Wang, Kiran Jagadeesh, Kunpeng Li, Luxin Zhang, Mannat Singh, Mary Williamson, Matt Le, Matthew Yu, Mitesh Kumar Singh, Peizhao Zhang, Peter Vajda, Quentin Duval, Rohit Girdhar, Roshan Sumbaly, Sai Saketh Rambhatla, Sam Tsai, Samaneh Azadi, Samyak Datta, Sanyuan Chen, Sean Bell, Sharadh Ramaswamy, Shelly Sheynin, Siddharth Bhattacharya, Simran Motwani, Tao Xu, Tianhe Li, Tingbo Hou, Wei-Ning Hsu, Xi Yin, Xiaoliang Dai, Yaniv Taigman, Yaqiao Luo, Yen-Cheng Liu, Yi-Chiao Wu, Yue Zhao, Yuval Kirstain, Zecheng He, Zijian He, Albert Pumarola, Ali Thabet, Artsiom Sanakoyeu, Arun Mallya, Baishan Guo, Boris Araya, Breena Kerr, Carleigh Wood, Ce Liu, Cen Peng, Dmitry Vengertsev, Edgar Schonfeld, Elliot Blanchard, Felix Juefei-Xu, Fraylie

Nord, Jeff Liang, John Hoffman, Jonas Kohler, Kaolin Fire, Karthik Sivakumar, Lawrence Chen, Licheng Yu, Luya Gao, Markos Georgopoulos, Rashel Moritz, Sara K. Sampson, Shikai Li, Simone Parmeggiani, Steve Fine, Tara Fowler, Vladan Petrovic, and Yuming Du. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*.

QwenTeam. Qwen2.5: A party of foundation models, September 2024. URL <https://qwenlm.github.io/blog/qwen2.5/>.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Int. Conf. Mach. Learn.*, pp. 8748–8763. PMLR, 2021.

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. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment Anything in Images and Videos. In *Int. Conf. Learn. Represent.*, 2025.

Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

Minsoo Rhu, Natalia Gimelshein, Jason Clemons, Arslan Zulfiqar, and Stephen W. Keckler. vdnn: virtualized deep neural networks for scalable, memory-efficient neural network design. In *The 49th Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO-49. IEEE Press, 2016.

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

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5–9, 2015, proceedings, part III 18*, pp. 234–241. Springer, 2015.

Runway. Gen-2: Generate novel videos with text, images or video clips. <https://runwayml.com/research/gen-2>, 2023.

Runway. Gen-3. <https://runwayml.com/>, 2024.06.

Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021.

Jay Shah, Ganesh Bikshandi, Ying Zhang, Vijay Thakkar, Pradeep Ramani, and Tri Dao. Flashattention-3: Fast and accurate attention with asynchrony and low-precision. *Adv. Neural Inform. Process. Syst.*, 37:68658–68685, 2025.

ShengShu-AI. Vidu. <https://www.vidu.studio/>, 2024.07.

Mohammad Shoeybi, Mostafa 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.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv:2010.02502*, October 2020.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Adv. Neural Inform. Process. Syst.*, 32, 2019.

Zhenxiong Tan, Songhua Liu, Xingyi Yang, Qiaochu Xue, and Xinchao Wang. OminiControl: Minimal and Universal Control for Diffusion Transformer. *arXiv preprint arXiv:2411.15098*, 2024.

TeamGLM, 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.

Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow (extended abstract). In *Assoc. Adv. Artif. Intell.*, 2021.

torchao maintainers and contributors. torchao: Pytorch native quantization and sparsity for training and inference, 2024.

Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphael Marinier, and Sylvain Gelly. Towards accurate generative models of video: A new metric and challenges, 2018.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Adv. Neural Inform. Process. Syst.*, 30, 2017.

Jianyuan Wang, Nikita Karaev, Christian Rupprecht, and David Novotny. VGGSfM: Visual geometry grounded deep structure from motion. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2024a. Highlight.

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

- Qixun Wang, Xu Bai, Haofan Wang, Zekui Qin, and Anthony Chen. Instantid: Zero-shot identity-preserving generation in seconds. *arXiv preprint arXiv:2401.07519*, 2024b.
- Xiang Wang, Hangjie Yuan, Shiwei Zhang, Dayou Chen, Jiuniu Wang, Yingya Zhang, Yujun Shen, Deli Zhao, and Jingren Zhou. VideoComposer: Compositional video synthesis with motion controllability. *Adv. Neural Inform. Process. Syst.*, 36:7594–7611, 2023b.
- Xiang Wang, Shiwei Zhang, Han Zhang, Yu Liu, Yingya Zhang, Changxin Gao, and Nong Sang. VideoLCM: Video latent consistency model. *arXiv preprint arXiv:2312.09109*, 2023c.
- Xiang Wang, Shiwei Zhang, Changxin Gao, Jiayu Wang, Xiaoqiang Zhou, Yingya Zhang, Luxin Yan, and Nong Sang. Unianimate: Taming unified video diffusion models for consistent human image animation. *arXiv preprint arXiv:2406.01188*, 2024c.
- Yujie Wei, Shiwei Zhang, Zhiwu Qing, Hangjie Yuan, Zhiheng Liu, Yu Liu, Yingya Zhang, Jingren Zhou, and Hongming Shan. DreamVideo: Composing your dream videos with customized subject and motion. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 6537–6549, 2024a.
- Yujie Wei, Shiwei Zhang, Hangjie Yuan, Xiang Wang, Haonan Qiu, Rui Zhao, Yutong Feng, Feng Liu, Zhizhong Huang, Jiaxin Ye, Yingya Zhang, and Hongming Shan. DreamVideo-2: Zero-Shot Subject-Driven Video Customization with Precise Motion Control. *arXiv preprint arXiv:2410.13830*, 2024b.
- Yujie Wei, Shiwei Zhang, Hangjie Yuan, Biao Gong, Longxiang Tang, Xiang Wang, Haonan Qiu, Hengjia Li, Shuai Tan, Yingya Zhang, and Hongming Shan. DreamRelation: Relation-Centric Video Customization. *arXiv preprint arXiv:2503.07602*, 2025.
- Pingyu Wu, Kai Zhu, Yu Liu, Liming Zhao, Wei Zhai, Yang Cao, and Zheng-Jun Zha. Improved video vae for latent video diffusion model. *arXiv preprint arXiv:2411.06449*, 2024.
- Yuxin Wu and Kaiming He. Group normalization. In *Eur. Conf. Comput. Vis.*, pp. 3–19, 2018.
- Shitao Xiao, Yueze Wang, Junjie Zhou, Huaying Yuan, Xingrun Xing, Ruiran Yan, Chaofan Li, Shuting Wang, Tiejun Huang, and Zheng Liu. OmniGen: Unified Image Generation. *arXiv preprint arXiv:2409.11340*, 2024.
- Chen-Wei Xie, Jianmin Wu, Yun Zheng, Pan Pan, and Xian-Sheng Hua. Token embeddings alignment for cross-modal retrieval. In *ACM Int. Conf. Multimedia*, 2022.
- Nianzu Yang, Pandeng Li, Liming Zhao, Yang Li, Chen-Wei Xie, Yehui Tang, Xudong Lu, Zhihang Liu, Yun Zheng, Yu Liu, and Junchi Yan. Rethinking video tokenization: A conditioned diffusion-based approach. *arXiv preprint arXiv:2503.03708*, 2025a.

Sidi Yang, Tianhe Wu, Shuwei Shi, Shanshan Lao Yuan Gong, Mingdeng Cao, Jiahao Wang, and Yujiu Yang. Maniqa: Multi-dimension attention network for no-reference image quality assessment. *arXiv e-prints*, 2022.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Xiaotao Gu, Yuxuan Zhang, Weihan Wang, Yean Cheng, Ting Liu, Bin Xu, Yuxiao Dong, and Jie Tang. CogVideoX: Text-to-Video Diffusion Models with An Expert Transformer. In *Int. Conf. Learn. Represent.*, 2025b.

Zhuoyuan Yao, Di Wu, Xiong Wang, Binbin Zhang, Fan Yu, Chao Yang, Zhendong Peng, Xiaoyu Chen, Lei Xie, and Xin Lei. Wenet: Production oriented streaming and non-streaming end-to-end speech recognition toolkit. In *InterSpeech*, 2021.

Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023.

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.

Hangjie Yuan, Shiwei Zhang, Xiang Wang, Yujie Wei, Tao Feng, Yining Pan, Yingya Zhang, Ziwei Liu, Samuel Albanie, and Dong Ni. InstructVideo: Instructing video diffusion models with human feedback. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 6463–6474, 2024a.

Shanghai Yuan, Jinfa Huang, Xianyi He, Yunyan Ge, Yujun Shi, Liuhan Chen, Jiebo Luo, and Li Yuan. Identity-Preserving Text-to-Video Generation by Frequency Decomposition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2025.

Zhihang Yuan, Hanling Zhang, Lu Pu, Xuefei Ning, Linfeng Zhang, Tianchen Zhao, Shengen Yan, Guohao Dai, and Yu Wang. DiTFastattn: Attention compression for diffusion transformer models. In *Adv. Neural Inform. Process. Syst.*, 2024b.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Int. Conf. Comput. Vis.*, pp. 11975–11986, 2023.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Adv. Neural Inform. Process. Syst.*, 32, 2019.

Jintao Zhang, Haofeng Huang, Penglle Zhang, Jun Zhu, Jianfei Chen, et al. Sageattention: Accurate 8-bit attention for plug-and-play inference acceleration. *arXiv preprint arXiv:2410.02367*, 2024.

Kai Zhang, Lingbo Mo, Wenhui Chen, Huan Sun, and Yu Su. MagicBrush: A Manually Annotated Dataset for Instruction-Guided Image Editing. In *Adv. Neural Inform. Process. Syst.*, 2023a.

- Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Int. Conf. Comput. Vis.*, pp. 3836–3847, 2023b.
- Shiwei Zhang, Jiayu Wang, Yingya Zhang, Kang Zhao, Hangjie Yuan, Zhiwu Qin, Xiang Wang, Deli Zhao, and Jingren Zhou. I2vgen-xl: High-quality image-to-video synthesis via cascaded diffusion models. *arXiv preprint arXiv:2311.04145*, 2023c.
- Youcai Zhang, Xinyu Huang, Jinyu Ma, Zhaoyang Li, Zhaochuan Luo, Yanchun Xie, Yuzhuo Qin, Tong Luo, Yaqian Li, Shilong Liu, Yandong Guo, and Lei Zhang. Recognize Anything: A Strong Image Tagging Model. *arXiv preprint arXiv:2306.03514*, 2023d.
- Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, Alban Desmaison, Can Balioglu, Pritam Damania, Bernard Nguyen, Geeta Chauhan, Yuchen Hao, Ajit Mathews, and Shen Li. Pytorch fsdp: Experiences on scaling fully sharded data parallel, 2023.
- 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.
- Daquan Zhou, Weimin Wang, Hanshu Yan, Weiwei Lv, Yizhe Zhu, and Jiashi Feng. Magicvideo: Efficient video generation with latent diffusion models. *arXiv preprint arXiv:2211.11018*, 2022.
- Shangchen Zhou, Chongyi Li, Kelvin C. K. Chan, and Chen Change Loy. ProPainter: Improving Propagation and Transformer for Video Inpainting. In *Int. Conf. Comput. Vis.*, pp. 10477–10486, 2023.