
MUG-V 10B: High-efficiency Training Pipeline for Large Video Generation Models

Yongshun Zhang^{*}, Zhongyi Fan^{*}, Yonghang Zhang, Zhangzikang Li, Weifeng Chen,
Zhongwei Feng, Chaoyue Wang[†], Peng Hou[†], Anxiang Zeng[†]
LLM Team, Shopee Pte. Ltd.
{daniel.wang, peng.hou}@shopee.com zeng0118@e.ntu.edu.sg

Abstract

In recent years, large-scale generative models for visual content (spanning images, 3D scenes, and videos) have made remarkable progress. However, training large-scale video generation models remains particularly challenging and resource-intensive due to their inherent multimodality, the long sequences of visual tokens involved, and the complex spatiotemporal dependencies. To address these challenges, we systematically designed the training framework, optimizing four key components: (i) data processing, (ii) model architecture, (iii) training strategy, and (iv) infrastructure. These optimizations delivered significant efficiency gains and performance improvements across all stages of data processing, video compression, parameter scaling, curriculum-based pretraining, and alignment-focused post-training. Our resulting model, MUG-V 10B, achieves performance on par with recent state-of-the-art video generation models. Notably, it excels in e-commerce-oriented video generation tasks, outperforming leading open-source models in human evaluations.

1 Introduction

Generative artificial intelligence models have advanced rapidly in recent years. Scaling laws have been empirically validated in Transformer-based foundation models, yielding strong performance across multiple modalities, including text, vision, and speech, and their associated tasks. This rapid progress has given rise to a range of AI-generated content (AIGC) applications, such as chatbots, creative tools, and agents, which are already reshaping production workflows and everyday life. Progress spans both general-purpose and domain-specialized large language models; for example, the CompassLLM series documents advances in multilingual and domain-focused capabilities [1, 2, 3].

In this work, we focus on developing a training framework for diffusion transformers (DiT) and training a large-scale video generation model. Video generation is among the most challenging forms of visual content synthesis: relative to image generation, it must preserve static content fidelity while learning diverse motion dynamics; relative to 3D generation, it should not only implicitly capture object-level 3D structure but also model inter-object interactions and physical regularities [4, 5]. Meanwhile, training large-scale video generation models also demands addressing three core challenges: inherently multimodal conditioning, long visual-token sequences, and complex spatiotemporal patterns [6, 7, 8, 9, 10].

To address these challenges, we adopt the prevailing video generation paradigm, *i.e.*, latent diffusion transformers with flow matching objectives [11, 12], and systematically design and implement an

^{*} Equal contribution.

[†] Corresponding author.

The rest of the authors' email address: {first_name}.{last_name}@shopee.com.

end-to-end training framework spanning data processing, video compression, model pre-training, post-training, infrastructure, and application evaluation. Along this pipeline, we investigate (i) how to execute each stage with high efficiency and minimal resource cost, and (ii) how to validate recent techniques and introduce techniques that further improve generative quality. Our contributions are summarized as follows:

- **Scalable data processing pipeline.** We built a pipeline that filters and extracts high-quality video clips from large corpora and uses a fine-tuned vision–language model (VLM) to generate structured, high-quality captions for all clips, with emphasis on throughput and stage-wise accuracy.
- **High-ratio VideoVAE compression.** We trained a VideoVAE that achieves $8 \times 8 \times 8$ compression along time, height, and width. Combined with non-overlapping 2×2 patchification in the DiT, this yields $\approx 2048\times$ compression relative to pixel space. With targeted architecture and loss design, reconstruction quality remains comparable to state-of-the-art VAEs at this ratio.
- **Training-stable Transformer backbone.** We designed a 10-billion-parameter Diffusion Transformer (DiT) with a transformer-block configuration that trains stably, and introduced a new image/frame conditioning scheme that improves cross-frame consistency.
- **Multi-stage training strategy for better video generation.** The process comprises (i) small-model hyperparameter validation, (ii) curriculum-based pre-training after scaling parameters, (iii) annealed SFT with curated high-quality data, and (iv) preference optimization using human-labeled annotations—substantially reducing trial-and-error compute while steadily improving performance.
- **Efficient training infrastructure.** Built on Megatron-Core [13], our system combines data, tensor, and pipeline parallelism to fully utilize hardware’s compute and interconnect, avoid activation recomputation, and incorporates hand-written Triton kernels. On the system with 500 Nvidia H100 GPUs, it achieves near-linear scaling.

2 Data Processing

Video–text pairs constitute the primary training corpus for large video generation models because they (i) are relatively inexpensive to acquire at scale, and (ii) jointly capture visual dynamics and their semantic descriptions. Unfortunately, most publicly available videos contain abrupt scene changes, static segments, duplicated content or heavy post-processing that make them unsuitable for direct use [14, 15, 16, 17]. We therefore built a scalable video processing pipeline that filters and captions raw footage to yield a large, diverse, and high-quality set of video clip–caption pairs. A relatively small, top-quality subset is further selected for post-training.

2.1 Scalable Video Processing Pipeline

We aggregate raw videos from both public and internal sources. Each video first undergoes a video-level screening for licensing, privacy compliance, prohibited content, and diversity of scenes, subjects, and motion. Only videos passing this gate enter the fine-grained pipeline below.

Video splitting. Accurately isolating semantically coherent segments is critical because current captioners struggle with clips that contain multiple scene transitions. We employ PySceneDetect [18] and Color-Struct SVM (CSS) method from [17] in tandem: PySceneDetect handles most cuts, while CSS complements it on identifying scene transitions such as gradual fades. Confidence thresholds are tuned according to data sources to maximise true-positive splits.

Visual-quality filtering. To guarantee sharp, aesthetically pleasing, and temporally coherent clips, we apply four-stage filtering:

- *Sharpness test.* The Laplacian-variance metric from OpenCV [19] highlights edge energy, frames with a variance $\in [200, 2000]$ are retained, otherwise the entire clip is rejected.
- *Aesthetic score.* A LAION-style aesthetic predictor discards clips scoring < 4.5 [20, 21].
- *Motion amplitude.* Optical flow magnitude is estimated with RAFT [22]. We sample three evenly spaced frame pairs (start, middle, end), average their flow magnitudes, and drop clips that are nearly static (< 1) or overly dynamic (> 20).

- *Multimodal LLM filter.* A proprietary model fine-tuned on 24k labelled videos is employed to identify clips with heavy post-processing (text overlays, large borders, special-effects), speed-altered footage, and camera shake.

Caption generation. High-fidelity captions drive both prompt-following and training convergence. We first finetune a Qwen-VL-72B [23, 24] captioner on public datasets plus internally labelled clips, optimising for descriptions that cover objects, appearance, motion, and background context. The capability is then distilled into a Qwen-VL-7B model [23, 24], striking a balance between accuracy and inference throughput for large-scale captioning.

Data balancing and deduplication. To control distributional bias and eliminate duplicates, we parse captions with a large language model that extracts key entities (subjects, actions, scenes). These tags form a lightweight ontology used to (i) stratify sampling so under-represented categories receive adequate weight and (ii) identify near-duplicate clips for removal. The resulting dataset offers a balanced mix of content while maximising diversity.

2.2 Human-Labelled Post-training Data

Pre-training equips the model with basic text–video alignment, motion priors, and a grasp of physical dynamics. Nevertheless, two problems persist: (i) Limited generation quality, *e.g.*, low aesthetics, motion discontinuities; (ii) Physical errors, *e.g.*, impossible trajectories, inconsistent details [25, 26, 27]. To address these issues we curate a human-verified post-training corpus that serves two complementary purposes: (i) refining the model on the highest-quality real videos and (ii) labeling preference signals that directly target remaining failure cases.

2.2.1 High-quality Clip Labeling

Score-based filtering. From the full pre-training set we retain the top $\approx 10\%$ of clips ranked by the composite score described in Section 2.1.

Distribution re-balancing. Unlike the pre-training stage, we intentionally up-weight human-centric clips (people, complex body motion, human–object interactions). Our empirical finding is that rigid-object dynamics are comparatively easy to learn, whereas articulated human motion remains a major bottleneck yet dominates real user queries.

Manual quality labeling. Automated filters still meet failure modes such as subtle scene splices (scene transition problem) or mild video compression artefacts. Human annotators therefore review each candidate clip on three axes: (i) Motion continuity (no jump cuts or speed ramps); (ii) Content stability (no scene changes, dissolves, or stitched footage); (iii) Visual fidelity (clarity, absence of heavy post-processing). Clips failing any criterion are discarded. The resulting subset forms the supervised post-training data, offering uniformly high visual and temporal consistency.

2.2.2 Preference Optimisation Data Labeling

Even with real-video data training, the performance of generative model can plateau before generating reasonable videos. We thus collect human preference annotations on the model’s own outputs:

Pairwise comparison labeling. Annotators compare two generated videos for overall aesthetics, motion smoothness, and severity of visual errors. The preferred video receives a positive label; the other receives a negative one.

Absolute correctness labeling. Independently, each clip is checked for (i) semantic match to the prompt, (ii) preservation of the main subject throughout the sequence, and (iii) presence of any physical or rendering errors. These evaluations yield binary pass or fail labels.

This dual annotation scheme powers the preference-learning stage (detailed in Section 4.2.3), enabling iterative improvement of generation quality and systematic reduction of physical errors.

3 Model Design and Architecture

Building on mainstream latent-diffusion and latent-flow transformer frameworks [28, 29, 30, 31, 32, 33], we adopt a two-stage generative pipeline. First, a variational auto-encoder (VAE) compresses

pixel-space video frames into a compact latent representation. Next, a 10-billion-parameter Diffusion Transformer (DiT) is trained to operate entirely in this latent domain, modeling spatiotemporal dynamics to synthesize videos. To unify text-to-video and image-to-video tasks within a single architecture, we devise an image-conditioning strategy that injects visual tokens from a reference image into the context stream of DiT, allowing controllable generation conditioned on either textual prompts or key frames.

3.1 Video Variational Autoencoder (Video VAE)

High-quality latent representations are pivotal for training video generation diffusion models. Our Video VAE balances three objectives: (i) maximal spatiotemporal compression, (ii) preservation of fine detail, and (iii) a lightweight encoding strategy that supports rapid iteration. The encoder downsamples each input clip by a factor of $8 \times 8 \times 8$ along the temporal, height, and width axes, achieving a 512× volumetric compression. Before entering the Diffusion Transformer (DiT), we apply a non-overlapping 2×2 spatial patchification that maps every four latents to a single token.

3.1.1 Video VAE Architecture

We initialise the Video VAE from a publicly available image VAE with strong reconstruction fidelity [34] and extend it to the video domain via hybrid convolutional stacks. Each down-sampling stage alternates a 2D spatial convolution, capturing intra-frame texture, with a 3D convolution that models inter-frame motion. This hybrid design retains the expressiveness of a fully 3D encoder while reducing FLOPs significantly relative to an all-3D counterpart.

Unlike prior work that separates ‘spatial’ and ‘temporal’ pathways [35], we adopt a unified architecture that jointly downsamples every dimension by eight. The resulting latent tensor $Z \in \mathbb{R}^{(T/8) \times (H/8) \times (W/8) \times C}$ encodes both appearance and motion cues in a compact form. Aggressive compression can harm fidelity, so we widen the bottleneck’s channel dimension to enhance latent capacity. Ablation studies show that increasing C markedly improves reconstruction until diminishing returns set in, we ultimately select $C = 24$ as the best trade-off between quality and storage budget. Similar observations are reported in [27].

3.1.2 Minimal Encoding Strategy

Temporal causal convolutions have become the de-facto choice in many existing Video VAE implementations because they (i) respect the arrow of time, (ii) allow a single model to encode variable-length clips, including degenerate cases such as still images or first-frame conditioning, and (iii) prevent information ‘leakage’ from future frames during video prediction. However, causal convolutions also introduce some drawbacks. When the distance from the current frame to the clip origin is smaller than the encoder’s temporal receptive field, earlier tokens aggregate less context than later ones, yielding an information imbalance across the latent sequence. Meanwhile, for clips whose length differs from the receptive field, the imbalance persists even after remedies such as fixed-length windows with overlapping weighted sums.

Minimal-Encoding Principle. To eliminate these issues, we proposed the *minimal encoding principle* for Video VAE. Specifically, we enforce that each latent token as an independent unit derived solely from its corresponding frame chunk (8 in our setting), thus no information is exchanged beyond this temporal window. We argue that the primary responsibility of Video VAE are compression and reconstruction, yet not generation. Thus, because the unit frame segment already contains the appearance and motion cues required to reconstruct itself, further context mixing is unnecessary and may even create shortcut learning. The minimal principle also yields a flexible latent interface: the same encoder can be used for arbitrary sequence lengths, for image-to-video or video continuation tasks, and for special cases such as first-, middle-, or last-frame conditioning.

Sharing Decoder Strategy. The decoder must reconstruct the complete clip from the latent sequence, it is not bound by the above ‘minimal principle’. Empirically, feeding an appropriate span of latents to the decoder in one shot leads to faster convergence than forcing unit-wise reconstruction. To balance throughput and memory, we train with single-latent encoding but vary the decoder’s input window across $\{1, 4, 8\}$ contiguous latents. At run time the encoder and decoder simply reshape their inputs to match the chosen window size (see Appendix A.1 Algorithm 1).

This minimal-encoding design removes the information-density imbalance of causal convolutions while retaining compatibility with downstream tasks and diverse clip lengths, contributing significantly to MUG-V 10B’s overall training efficiency.

3.2 MUG-V 10B Diffusion Transformer Model

The generative core of MUG-V is a 10-billion-parameter Diffusion Transformer. The model is trained jointly for text-to-video, image-to-video, and text-plus-image-to-video synthesis, thereby unifying the principal conditioning modalities required for modern video generation. Its backbone follows the DiT architecture [28], ensuring compatibility with state-of-the-art diffusion techniques. Our DiT backbone consists of four components: (i) input patchifying, (ii) text condition networks, (iii) stacked DiT blocks, and (iv) output unpatchifying. Its overall organisation follows some existing DiT models [28, 30, 35], so this report focuses on specific design choices rather than restating the entire architecture.

Transformer block. Instead of the MM-DiT block used in some image/video diffusion models [29, 36], we adopt a transformer block architecture closely aligned with that of autoregressive language models. A cross-attention module is inserted between the self-attention and feed-forward network (FFN) to enable direct interaction between textual embeddings and visual tokens.

Full attention v.s. spatio-temporal separated attention. Current DiT variants employ either full attention [33], where every token in the spatiotemporal sequence attends to every other, or spatio-temporal separated attention [37], which restricts attention to a local neighbourhood to reduce computation. Full attention provides stronger global coherence, for example, the same person or background appearing at the start and end of a clip can interact directly. Because our Video VAE and patchifying scheme yield a high compression ratio, full attention does not incur prohibitive cost, so we adopt it throughout.

3D RoPE encoding for visual tokens. To allow full attention to capture accurate positional cues, we apply three-dimensional Rotary Position Embedding (RoPE), which extends the original 1D formulation to jointly encode spatial and temporal coordinates [38].

Global signal embedding. Global signals such as diffusion timesteps and video frame-rate are embedded following [29]. A shared MLP maps each global scalar to the model dimension, and per-block learnable scale parameters modulate the resulting vector, balancing expressiveness with memory efficiency.

Normalisations. Consistent with prior large-scale models, normalisation improves training stability. Beyond the QK normalisation inside self-attention, we normalise input text features and the cross-attention module [39, 40]. Empirically, these layers markedly reduce parameter volatility and attenuate loss fluctuations, leading to fewer visual artefacts during the training procedure.

Image/frame conditioning. For image- or frame-conditioned video generation, we mask the video sequence rather than add conditional latents to the denoising latent. Conditioned regions receive the given image/frame latent and have their diffusion timestep set to zero (zero noise added), while the remaining tokens follow the standard noisy diffusion trajectory. During pre-training this strategy both clarifies the timestep signal and yields superior fidelity to the provided visual content at inference.

4 Model Training

4.1 Video VAE Training

The Video VAE is trained with the composite loss,

$$\mathcal{L}_{\text{VAE}} = \mathcal{L}_{\text{rec}} + \lambda \mathcal{L}_{\text{KL}} + \gamma \mathcal{L}_{\text{GAN}}, \quad (1)$$

where the three terms serve complementary purposes:

- **Reconstruction loss** $\mathcal{L}_{\text{rec}}$ is a weighted sum of $\mathcal{L}_{\text{MSE}}$, $\mathcal{L}_1$, and $\mathcal{L}_{\text{perc}}$, we encourage pixel-level accuracy (MSE, ℓ_1) and perceptual fidelity ($\mathcal{L}_{\text{perc}}$).
- **Kullback–Leibler divergence** $\mathcal{L}_{\text{KL}}$ regularises the latent distribution, suppressing outliers and promoting smooth interpolation.

- **Adversarial loss** $\mathcal{L}_{\text{GAN}}$ is applied only during the final fine-tuning stage to sharpen texture and colour. Because excessive adversarial weighting can introduce hue shifts or over-enhanced details, we keep γ small and monitor validation PSNR/SSIM.

Adaptive Reconstruction Weighting. After the core objective stabilises, we observe that the model readily reconstructs global structure but oscillates on highly dynamic, fine-detail regions. To focus learning on these harder cases, we introduce an *adaptive reconstruction loss*.

For each reconstructed frame x_t we compute a spatiotemporal saliency map

$$w_t = \left| \Delta_t(\nabla^2 x_t) \right|,$$

where ∇^2 is the Laplacian (extracting high-frequency spatial edges) and Δ_t is the temporal forward difference (highlighting fast motion). Then, we employ w_t to form the weighted loss term $\mathcal{L}_{\text{adaptive}}$ to replace the plain ℓ_1 component in $\mathcal{L}_{\text{rec}}$. Regions with rapid spatiotemporal change thus contribute a larger gradient signal, improving convergence without additional data passes.

4.2 MUG-V 10B Diffusion Transformer Training

To achieve high training efficiency and maintain convergence stability at this scale, besides the model architectural refinements, we incorporate three technical measures: (i) a parameter-expansion strategy accompanied with systematic hyper-parameter search; (ii) a multi-stage pre-training curriculum; and (iii) supervised and preference optimization based post-training. These designs enable stable and resource-efficient training of the 10B parameter DiT without compromising video generation quality.

4.2.1 Parameter Expansion

Considering that perform exhaustive scaling law studies and hyper-parameter sweeps would cost plenty of computing resources, we adopted a two-stage workflow: first train a compact model, then expand its parameters to the 10B scale for continued training.

Similar to zero-shot hyper-parameter transfer researches [41, 42], we fixed the target depth at 56 Transformer blocks and built a smaller DiT with hidden size 1728 (leading to a approximate 2B parameters model). Its low training cost and fast inference made it ideal for rapid experimentation and recipe validation. Once this 2B model achieved satisfactory video-generation quality, we enlarged it via a hidden-size equi-variant expansion.

Our strategy closely related to the *HyperCloning* expansion method [43], we both increase channel width while preserving the network’s functional behaviour. Consider a linear layer with weights $W \in \mathbb{R}^{d \times d}$ and bias $b \in \mathbb{R}^d$. Expanding the hidden dimension by a factor e produces $W' \in \mathbb{R}^{ed \times ed}$ and $b' \in \mathbb{R}^{ed}$ by tiling the original parameters and dividing by e to keep feature scaling unchanged. Meanwhile, random perturbations are added to avoid the gradient duplication problem. Thus,

$$W' = \frac{1}{e} \left(\begin{pmatrix} W & \cdots & W \\ \vdots & \ddots & \vdots \\ W & \cdots & W \end{pmatrix} - \begin{pmatrix} \epsilon_{11} & \cdots & \epsilon_{1n} \\ \vdots & \ddots & \vdots \\ \epsilon_{m1} & \cdots & \epsilon_{nm} \end{pmatrix} \right), \quad b' = \frac{1}{e} \begin{pmatrix} b \\ \vdots \\ b \end{pmatrix}. \quad (2)$$

Inputs are replicated, $x'_i = [x_i; \dots; x_i]$, and the outputs $x'_o \approx [x_o; \dots; x_o]$, each repeated e times. Setting $e = 2$ increased the total parameter count by roughly $4\times$. After initialising the 10B model with these expanded weights, we transferred the hyper-parameters tuned on the 2B model and resumed training [41, 44, 42]. This output-preserving expansion accelerated convergence, while the small-model stage substantially reduced overall experimentation cost.

4.2.2 Multi-stage Pre-training Curriculum

The heterogeneous nature of video data, where low-level texture and high-level semantics coexist, makes curriculum learning particularly effective for training video generation model. At low spatial resolution, semantic content dominates, as resolution increases, richer textures emerge. Moreover, a video can be viewed as a dynamic extension of static image, with motion learned on top of appearance. Leveraging these properties, we adopt a three-stage curriculum:

- Stage 1 mixes image data with low-resolution (360p) video clips. The image-to-video ratio is annealed during training until video dominates, at which point the model reliably produces plausible images and coarse video clips.
- Stage 2 retains the 360p resolution but increases clip length from 2s to 5s, and training continues until the validation loss plateaus.
- Stage 3 replaces the training set with 5s clips at 720p, curated from around 12M high-quality videos, constituting the final pre-training phase.

Note that (i) the relatively small model before parameter-expansion use only images and 360p videos; (ii) aforementioned masking strategy for image/frame conditioning is compatible with the text-to-video generation pretraining, and we introduce the first frame masking in both stage 2 and 3.

This curriculum not only guides the model to acquire video-generation skills progressively but also boosts training efficiency. In Stages 1 and 2, shorter sequences and higher throughput allow the model to see over ten times more samples than in Stage 3, fostering robust general abilities. Stage 3, although computationally costly, refines detail thanks to its rigorously filtered, high-resolution data.

4.2.3 Post-training and Alignment

After the multi-stage pre-training described above, the validation loss plateaued and began to oscillate, the model’s outputs exhibited two persistent failure modes: (i) fine-grained artefacts, especially in articulated regions such as human hands, (ii) violations of basic physical plausibility (*e.g.*, interpenetration and distortions). To further improve generative quality we adopted two post-training approaches: annealed supervised fine-tuning (SFT) with post-EMA, and preference-based optimization.

Annealed SFT with post-EMA. We first refined the training corpus, manually selecting around 0.3 M high-quality clips. Continuing training on this subset with a gradually decaying learning rate proved effective. We compared online exponential-moving-average (EMA) parameter smoothing with a post-hoc EMA [45] variant. The latter not only removed the need for expensive grid search over EMA hyper-parameters but also more likely to produce higher video quality. Instead of the post-hoc EMA proposed by Karras *et al.* [45], we approximate it by exponentially decayed model ensembling, which is conceptually similar to the *model merging* strategy in [46] and empirically outperforms standard online EMA in our setting.

Preference optimisation. Although preference-based reinforcement learning has achieved notable success in large language models, its application to video generation remains challenging due to (i) the limited capacity of current video evaluation (reward) models and (ii) the multiplicity of optimization axes, such as appearance, motion, temporal coherence, and so on. We therefore resorted to human-annotated preferences, focusing on two objectives:

- *Error-free generation.* For failures such as interpenetration, deformation, or other physical implausibilities we collected absolute positive/negative labels and optimised the model with the KTO algorithm [47, 48].
- *Motion quality.* To improve dynamic realism we obtained pairwise “better/worse” annotations and applied the DPO algorithm [49, 50].

Retaining the original supervised fine-tuning (SFT) objective as a regularizer during preference optimization mitigated the risk of the model adopting undesirable statistical biases (*e.g.*, exaggerated motion amplitude or recurring texture patterns). Conducting preference optimization in multiple stages and interleaving batches from different annotation sources allowed the model to sequentially expose distinct classes of errors, thereby achieving continuous quality improvements.

5 Infrastructure

Beyond algorithmic design, infrastructure is pivotal to achieving efficient and stable training for large-scale video generation. Our video generation DiT model faces *processing long sequences with full attention*, *scaling to billions of parameters*, and *preserving numerical precision during training* three core challenges. We therefore build a Megatron-Core [13] based training framework

for MUG-V 10B, concentrating on three optimizations: (i) model-parallel strategy, (ii) balanced data-loading/training pipelines, and (iii) fused kernel, to overcome these obstacles.

5.1 Model Parallel Strategy

Given the long-sequence nature of video data, which incurs higher dynamic memory consumption than language models’ pretraining, we systematically explored parallelization techniques to maximize throughput. Our hybrid scheme combines data parallelism (DP), tensor parallelism (TP), pipeline parallelism (PP), and sequence parallelism (SP).

To train our 10B DiT model, we first enable TP within a single node. To alleviate the memory burden of long sequences, we shard activations across the TP group via SP. Next, we apply PP, vertically partitioning layers and leveraging point-to-point communication to exploit inter-node bandwidth while disabling activation recomputation. Finally, we introduce DP to enlarge the effective batch size and improve training stability. Extensive benchmarking identifies an optimal 10B-scale configuration that delivers near-linear efficiency scaling, thereby maximizing hardware utilization.

5.2 Data Loading and Computation Balance

Beyond optimizing parameter updates, efficient data ingestion is crucial to overall training throughput. We build an asynchronous I/O pipeline with aggressive pre-fetching and caching, overlapping data preprocessing and transfer with computation to hide latency. To minimize pipeline stalls arising from variable video sequence lengths, we also introduce dynamic balanced sampling across all ranks. This scheme ensures that each GPU receives batches of comparable computational cost, reducing idle cycles and further improving hardware utilization.

5.3 Kernel Fusion

To reduce DiT’s memory overhead from pixel-wise modulation and residual paths, we design a two-tier fusion of low-level kernels and block refactoring.

We merge three tightly coupled operations, (i) linear-layer bias addition, (ii) per-pixel scale-and-shift modulation, and (iii) residual accumulation into a single GPU kernel. Collapsing the read–compute–write sequence into one pass cuts global-memory transactions from N down to one. The fused kernel is handwritten in Triton, leveraging warp-level shuffles to broadcast bias and modulation vectors without shared-memory spills. A persistent-threads scheduling pattern keeps intermediate data resident in registers across the three fused stages, pushing bandwidth utilisation toward hardware limits and further trimming memory traffic.

At a higher level, we reshape the DiT block to expose additional fusion opportunities:

- **LayerNorm + QKV Projection.** Layer normalization is executed in tandem with the query–key–value (QKV) projection, eliminating an extra memory round-trip.
- **Masked Softmax Fusion.** Attention-score masking is folded directly into a FlashAttention-2 soft-max kernel, avoiding redundant reads of the score matrix.
- **Zero-Padding Removal.** Static shape inference removes unnecessary padding, ensuring fully coalesced accesses.

Together, these optimizations lower memory traffic, increase arithmetic intensity, and deliver an end-to-end speed-up.

6 Applications & Model Performance

Video-generation technology is now routinely applied in film, gaming, advertising, and e-commerce, where it offers substantial gains in creativity and cost efficiency. As an e-commerce company, we focus on retail-specific situations: generating dynamic product videos such as try-on showcases, still-life displays, functional demonstrations, and advertising assets. To be viable in this setting, generated videos must exhibit (i) generative correctness (semantically accurate content and physically plausible motion), (ii) content consistency, and (iii) visual appeal. These requirements largely align with established evaluation protocols, so we first benchmark our models with standard automatic

metrics. However, we find that existing metrics often overlook fine-grained defects, *e.g.*, altered fabric textures or incorrect hand poses, that are critical for product fidelity. We therefore supplement automatic scores with human evaluations to judge overall usability and quality.

Table 1: Quantitative comparisons of video generation¹.

Model	Model Size	VTM	VISC	VIBC	SC	BC	MS	DD	AQ	IQ	I2V Score	Quality Score	Total Score
CogVideoX [51]	5b	<u>67.68</u>	97.19	96.74	94.34	96.42	98.40	33.17	61.87	70.01	94.79	78.61	86.70
STIV [52]	8.7b	11.17	98.96	97.35	98.40	<u>98.39</u>	99.61	15.28	66.00	70.81	93.48	79.98	86.73
Step-Video [53]	30b	49.23	97.86	98.63	<u>96.02</u>	<u>97.06</u>	<u>99.24</u>	48.78	62.29	<u>70.44</u>	<u>95.50</u>	<u>81.22</u>	88.36
Dynamic-I2V [54]	5b	88.10	<u>98.83</u>	98.97	<u>96.21</u>	<u>98.39</u>	<u>98.88</u>	27.15	60.10	<u>69.23</u>	98.12	78.78	88.45
HunyuanVideo [36]	13b	49.91	98.53	97.37	95.26	96.70	<u>99.23</u>	22.20	62.55	<u>70.14</u>	95.10	78.54	86.82
Wan2.1 [33]	14b	34.76	96.95	96.44	94.86	<u>97.07</u>	97.90	<u>51.38</u>	<u>64.75</u>	<u>70.44</u>	92.90	80.82	86.86
MAGI-1 [55]	24b	<u>50.85</u>	98.39	<u>99.00</u>	93.96	96.74	98.68	68.21	<u>64.74</u>	69.71	<u>96.12</u>	82.44	89.28
MUG-V(Ours)	10b	23.17	<u>98.82</u>	99.51	95.73	98.52	98.90	<u>57.24</u>	61.37	68.48	95.37	<u>81.55</u>	<u>88.46</u>

6.1 Quantitative Evaluation of Video Generation

To evaluate the quality of videos generated by MUG-V 10B, especially the text-image to video(TI2V) setting emphasized in e-commerce, we adopt the VBench protocol and related metrics. We assess overall quality along three dimensions, *i.e.*, temporal consistency, motion dynamics, and perceptual aesthetics/distortion, using six metrics: Subject Consistency (SC), Background Consistency (BC), Motion Smoothness (MS), Dynamic Degree (DD), Aesthetic Quality (AQ), and Imaging Quality (IQ). Additionally, three I2V-specific metrics are included: Video-Text Camera Motion (VTM), Video-Image Subject Consistency (VISC), and Video-Image Background Consistency (VIBC). The final VBench score is computed as a weighted sum of these components [56, 57]. In our experiments, we strictly follow the VBench-I2V evaluation and submit results to the public leaderboard. As shown in Table 1, our model performs strongly across almost all metrics. At submission time, MUG-V 10B ranks third on the VBench I2V leaderboard, behind Magi-1 and the commercial system PI.

6.2 Human Evaluation on E-commerce Video Generation Tasks

To more directly compare against leading open-source model, HunyuanVideo and Wan 2.1, we conducted a human evaluation tailored to e-commerce video generation. Test inputs were randomly sampled from publicly available model showroom images. For each method, we used its default prompt generator to create video prompts and produced 5 seconds clips. All clips were pooled and randomly ordered, then evaluated in parallel by three independent annotators, final labels were determined by consensus (*i.e.*, ≥ 2 of 3).

The annotation proceeded in three stages. First, annotators judged whether a clip was discernibly AI-generated, considering both the presence of errors (from physical implausibilities to minor artifacts) and overall visual realism. Second, for clips deemed sufficiently realistic, annotators assessed product consistency relative to the input image, requiring that color, material, texture, and other attributes remain unchanged. We consider a clip deployable in e-commerce only if it satisfies these two criteria. Third, for deployable clips, annotators judged whether the video is “high quality,” defined by the hallmarks of professional cinematography and model performance. Finally, our model achieves strong results on both the pass rate and the high-quality rate. Since the space limitation, the detail evaluation results are reported in Appendix B.2. Nevertheless, we observe that residual minor artifacts and geometric distortions still limit overall quality, indicating substantial headroom for improvement in e-commerce applications.

7 Related Works

7.1 Diffusion Models

Diffusion-based generative modeling originates from score matching and denoising autoencoders, culminating in denoising diffusion probabilistic models (DDPM) and the continuous-time score-

¹Given that VBench is a widely used benchmark for video-generation evaluation, we submitted our results to enable direct comparison with prior methods. We present a subset of the VBench-I2V leaderboard, restricted to recent methods accompanied by a technical report. The complete leaderboard is available at this link.

based SDE/ODE formulations [58, 59, 60]. These methods learn a reverse-time denoising process to transform Gaussian noise into data, and support conditioning through classifier/classifier-free guidance as well as latent-space diffusion with learned encoders for efficiency [61, 62, 63].

A subsequent line of work replaces stochastic reverse diffusion with deterministic transport, framing generation as learning a velocity field that pushes a simple prior to the data distribution. Rectified flow and flow matching objectives directly supervise this transport via continuity equations or optimal-transport-inspired training, often yielding faster sampling and simpler training dynamics [11, 12]. Complementary advances distillation to few/one-step samplers, consistency models, improved solvers, and DiT backbones—further reduce inference cost while preserving fidelity [64, 65, 28].

Diffusion and flow matching have been successfully applied across modalities. In images, latent diffusion enabled text-conditional, high-resolution synthesis at scale [34, 66, 29, 67], DiT backbones improved scaling and training stability [28]. In 3D, score-distillation and related techniques optimize neural or explicit 3D representations from text or image supervision [68, 69]. Audio and music generation commonly operate in spectrogram space with text or melody conditioning [70, 71]. Motion, trajectories, and robotics have leveraged diffusion priors for controllable dynamics [72]. Extensions to discrete domains (code or text) use relaxed tokenizations or hybrid AR–diffusion designs [73, 74, 75]. These developments establish diffusion/flow matching as flexible, scalable foundations for high-dimensional generative tasks, including video.

7.2 Video Generation Models

Early text-to-video systems extended image diffusion with temporal priors or cascaded frame synthesis and super-resolution, but were limited in duration, resolution, and temporal coherence [76, 77, 78]. Latent video diffusion improved efficiency by compressing videos with video VAEs before applying spatiotemporal denoising [79, 80], enabling longer clips and higher fidelity [81].

A dominant family today uses DiT-based generators operating in a latent video space: a video VAE provides compact spatiotemporal latents, while a DiT (with factorized or windowed spatiotemporal attention) performs conditional generation under text, image, or control signals [82, 83]. Advances include stronger conditioning (pose, depth, camera paths, audio), longer context handling (memory-efficient attention, sliding windows), and faster sampling via consistency or flow matching objectives. Large proprietary systems, *e.g.*, Sora demonstrate long-duration, high-resolution generation with improved physical plausibility through large-scale training, aggressive latent compression, and optimized inference [84, 85, 33, 36, 37, 86, 87, 88].

In parallel, autoregressive (AR) based approaches tokenize videos with vector-quantized encoders and model generation as next-token prediction across spatiotemporal tokens [89, 90]. These models integrate naturally with multimodal LLMs and show strengths in long-horizon structure and discrete controllability, but often trade off visual fidelity and suffer from compression artifacts. Hybrid systems combine AR planning (for structure and semantics) with diffusion/flow decoders (for photorealism), narrowing this gap [91, 92].

Positioned within this landscape, MUG-V 10B follows the DiT-in-latent-video paradigm with an emphasis on efficient training (Video VAE compression, kernel/system optimizations) and modern training curriculum, while targeting practical conditioning modes (text-to-video and image-to-video) and alignment for e-commerce content.

8 Conclusion

In this report, we presented the training framework of MUG-V 10B diffusion transformer (DiT) model for video generation. Under constrained compute, we pursued an end-to-end design that integrates scalable data processing, a high-compression Video VAE, a DiT-based generator, multi-stage pre-training and post-training, and systems-level optimizations for efficient training and evaluation. Our study not only validates several recent advances for large-scale DiT model training, but also introduces practical strategies that stabilize optimization and improve generated sample quality. Across qualitative and quantitative evaluations, MUG-V 10B delivers competitive or superior performance, particularly in e-commerce scenarios.

A Additional Technical Details

In this section, we provide additional technical details that were omitted from the main text due to space constraints.

A.1 More Details of Video VAE

Algorithm 1 presents the pseudocode of the Video VAE Minimal Encoding Strategy. As shown, the proposed strategy can be implemented with a simple tensor reshape operation, introducing no additional computational overhead to the overall process.

Algorithm 1 Video VAE Minimal Encoding Strategy

Input: $\mathbf{V} \in \mathbb{R}^{1 \times T \times C \times H \times W}$, $R \in \{1, 4, 8\}$ $\triangleright$ an input video $\mathbf{V}$, and the decoder window R
1: $\mathbf{V}_{\text{in}} \leftarrow \text{reshape}(\mathbf{V}, [T/8, 8])$ $\triangleright (T/8) \times 8 \times C \times H \times W$
2: $\mathbf{E} \leftarrow \text{VaeEncoder}(\mathbf{V}_{\text{in}})$
3: $(\boldsymbol{\mu}, \log \boldsymbol{\sigma}^2) \leftarrow \text{Linear}(\mathbf{E})$
4: $\mathbf{z} \leftarrow \text{reparameterize}(\boldsymbol{\mu}, \log \boldsymbol{\sigma}^2)$ $\triangleright (T/8) \times 1 \times C_z$
5: $\mathbf{D} \leftarrow \text{reshape}(\mathbf{z}, [(T/8)/R, R])$ $\triangleright (T/8)/R \times R \times C_z$
6: $\hat{\mathbf{V}} \leftarrow \text{VaeDecoder}(\mathbf{D})$ $\triangleright (T/8)/R \times (8 \cdot R) \times C \times H \times W$
Output: $\text{reshape}(\hat{\mathbf{V}})$ $\triangleright$ reconstructed video, $1 \times T \times C \times H \times W$

A.2 More Details of Preference Optimisation

In addition to direct preference optimization (DPO) and KTO with human-labeled data, we introduce Real Data Preference Optimization (RDPO) [26]. By applying reverse sampling on real video data, we observe that the flow sampling paths derived from these reverse samples are statistically superior to those obtained from randomly initialized noise and its associated flow trajectories. This property allows RDPO to automatically construct preference pairs without the need for manual annotation. Furthermore, a multi-stage iterative training schedule is employed to progressively improve the generator’s performance.

B Additional Experiments

B.1 Video VAE Reconstruction

Within our video generation pipeline, the Video VAE is dedicated to compressing the video signal and reconstructing it. We therefore curated a set of real-world clips for validation and evaluated reconstruction fidelity with standard metrics, PSNR, SSIM, LPIPS, and FloLPIPS, against several baseline VAE models. As summarized in Table 2, our Video VAE surpasses most comparators on these metrics. Although its score on SSIM (720p setting) is slightly lower than that of CogVideoX

Table 2: Quantitative comparisons of video reconstruction.

Model	Downsample Factor	Res.	Evaluation Metrics			
			PSNR($\uparrow$)	SSIM($\uparrow$)	LPIPS($\downarrow$)	FloLPIPS($\downarrow$)
Opensora VAE	$4 \times 8 \times 8$	256p	28.2	0.821	0.114	0.108
CogVideoX VAE	$4 \times 8 \times 8$	256p	30.3	0.902	0.055	0.053
MUG-V VAE	$8 \times 8 \times 8$	256p	32.2	0.912	0.053	0.048
Opensora VAE	$4 \times 8 \times 8$	480p	30.0	0.857	0.107	0.101
CogVideoX VAE	$4 \times 8 \times 8$	480p	30.5	0.918	0.044	0.045
MUG-V VAE	$8 \times 8 \times 8$	480p	31.2	0.911	0.043	0.041
Opensora VAE	$4 \times 8 \times 8$	720p	30.6	0.866	0.109	0.105
CogVideoX VAE	$4 \times 8 \times 8$	720p	31.8	0.912	0.058	0.058
MUG-V VAE	$8 \times 8 \times 8$	720p	32.9	0.911	0.056	0.056

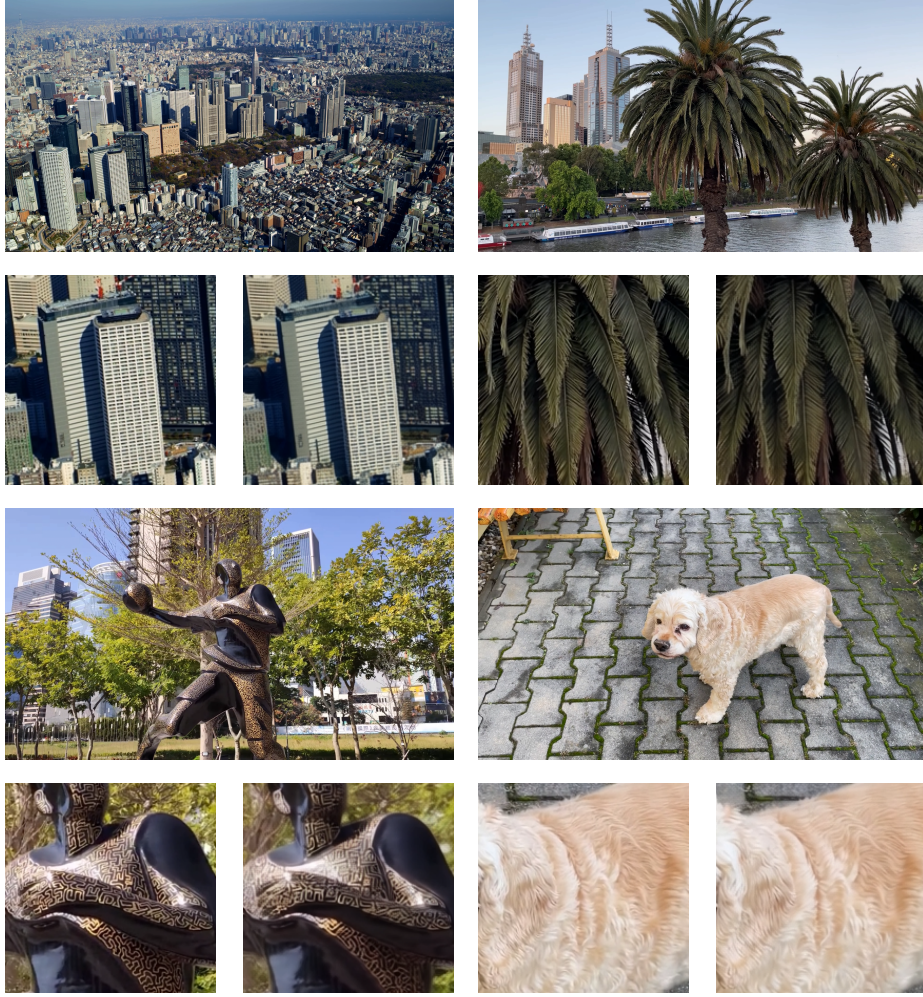


Figure 1: The visualization of Video VAE reconstruction examples. For each example, we provide the input video frame (the whole frame and local details) and the local patch extracted from the reconstructed video clip (the right enlarge part).

Human Evaluation on E-commerce Video Generation Tasks

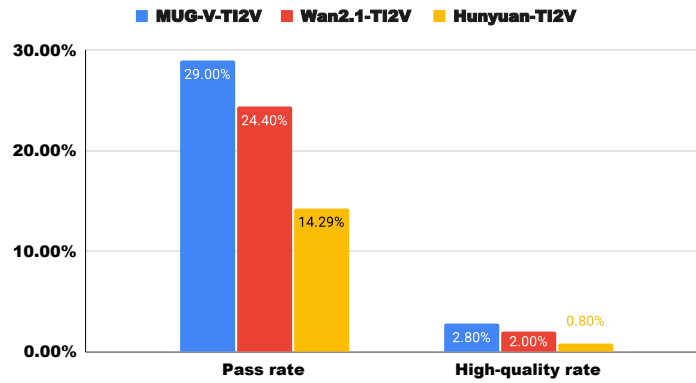


Figure 2: The human evaluation comparisons on generated e-commerce video of their quality.

VAE, it delivers an $8 \times 8 \times 8$ compression ratio, achieving a favorable efficiency–quality balance. Qualitative examples in Fig. 1 show that fine details such as drifting smoke and rapidly changing textures are faithfully reproduced.

B.2 Evaluation Results of E-commerce Video Generation

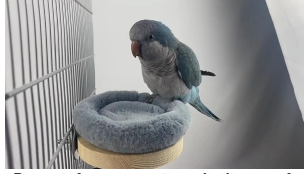
In Fig. 2, we report the evaluation results of different models on e-commerce video generation, measured by pass rate and high-quality rate. In mixed blind evaluations, our MUG-V 10B achieves superior scores. Specifically, the higher pass rate indicates that our model generates a larger proportion of e-commerce videos without noticeable artifacts or errors, making them indistinguishable from real footage. Meanwhile, the improved high-quality rate reflects better performance in terms of motion coherence, visual fidelity, and aesthetics. Nonetheless, the relatively low absolute values of both metrics highlight that substantial room for improvement remains, underscoring the need for further advancement in video generation models, including ours.

B.3 Visualizing Generated Videos

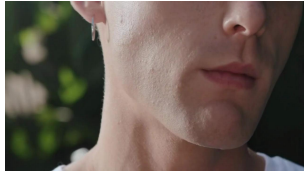
We present representative qualitative results in Fig. 3 and Fig. 4. Fig. 3 shows text-to-video (T2V) samples, while Fig. 4 displays image-to-video (I2V) results. Moreover, the generated samples of e-commerce video generation evaluation tasks are presented in Figs. 5, 6, 7. More video demonstrations are available on our project website.

C Challenges and Future Work

Despite these advances, our experiments highlight several open challenges. First, strengthening the faithfulness and controllability of the mapping from conditioning signals (text, image, or mixed inputs) to generated videos remains a prerequisite for reliable real-world deployment. Second, fine-grained appearance fidelity, such as material and texture preservation, still lags, with sensitivity to VAE compression and DiT noise initialization leading to subtle but consequential degradations. Third, scaling to longer durations and higher resolutions demands algorithms and systems that cope with long-sequence training, inference efficiency, and long-range temporal consistency. In light of these challenges, we remain committed to advancing the capabilities of video generation models and look forward to continued progress from the broader research community.



Prompt: A gray parrot perched on a soft, plush cushion inside a cage. The cage itself is made of metal bars. The cushion it is sitting on appears to be comfortable and well-suited for the bird's needs. The background is plain wall.



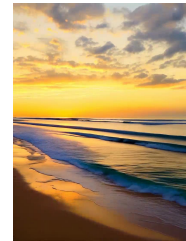
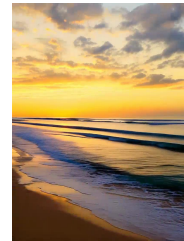
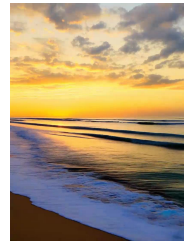
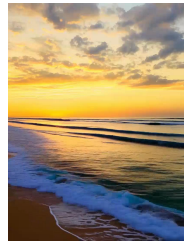
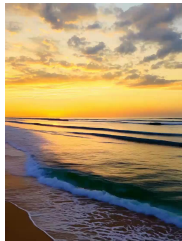
Prompt: A close-up shot of a man's face, focusing on his neck and lower face. The background is blurred, with hints of greenery plants. The lighting is soft and natural, casting gentle shadows on the person's skin. The camera slowly moves to reveal his eyes.



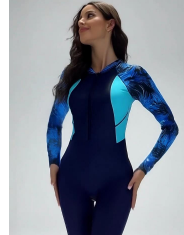
Prompt: A wooden table with several glass containers. In the foreground, there is a clear glass pitcher filled with a tea and slices of lemon. Next to the pitcher, there are empty glass cups. A person is seen pouring a drink into a glass. The action is smooth and graceful.



Prompt: Two Christmas elf dolls sitting on a green wreath. The wreath is adorned with red berries and small white lights, creating a warm and inviting atmosphere. The background of the scene includes a wooden deer head and several wooden trees, adding to the holiday theme.



Prompt: A serene and breathtaking beach scene at sunset. The waves create delicate white foam as they break on the shore, adding a sense of tranquility to the scene.



Prompt: A woman wearing a blue and black tight suit with long sleeves. The tight has a zipper in the front and is designed with a combination of solid colors and patterns. The woman is standing against a plain, light-colored background. She is posing for the camera, showcasing the fit and design of the tight.

Figure 3: Visualization of text-to-video generation results produced by the MUG-V 10B model. (enlarge for more details.)



Figure 4: Visualization of image-to-video generation results produced by the MUG-V 10B model. In each example, the first frame corresponds to the conditioning image. (enlarge for more details.)

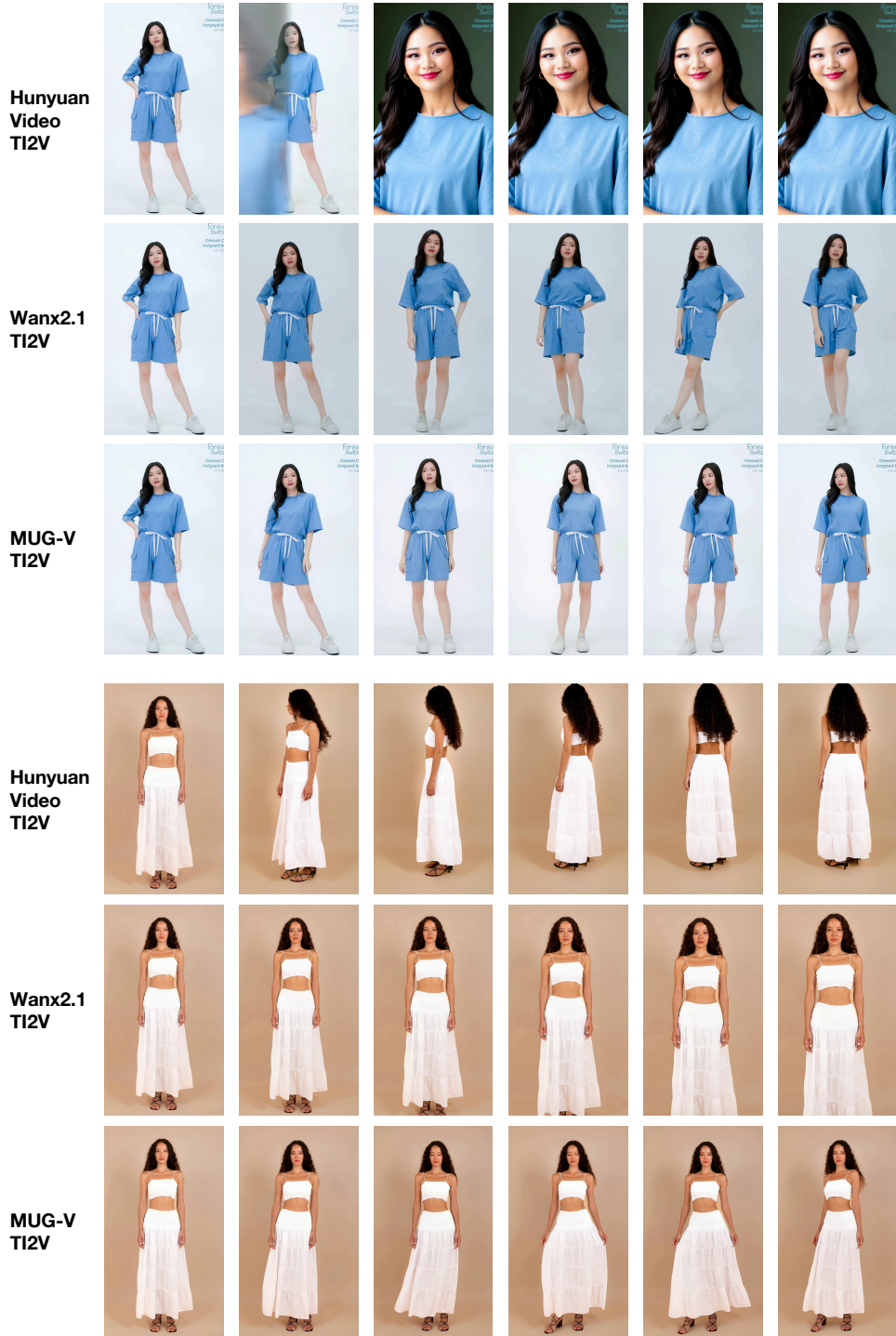


Figure 5: Visualization of e-commerce video generation results across different models. Since each model is optimized for distinct prompt styles, we employed their respective default prompts or prompt-rewriting tools for fair comparison. (enlarge for more details.)



Figure 6: Visualization of e-commerce video generation results across different models. Since each model is optimized for distinct prompt styles, we employed their respective default prompts or prompt-rewriting tools for fair comparison. (enlarge for more details.)



Figure 7: Visualization of e-commerce video generation results across different models. Since each model is optimized for distinct prompt styles, we employed their respective default prompts or prompt-rewriting tools for fair comparison. (enlarge for more details.)

References

- [1] Sophia Maria. Compass: Large multilingual language model for south-east asia, 2024.
- [2] Sophia Maria. Compass-v2 technical report, 2025.
- [3] Sophia Maria. Compass-v3: Scaling domain-specific llms for multilingual e-commerce in southeast asia, 2025.
- [4] Yimu Wang, Xuye Liu, Wei Pang, Li Ma, Shuai Yuan, Paul Debevec, and Ning Yu. Survey of video diffusion models: Foundations, implementations, and applications. *arXiv preprint arXiv:2504.16081*, 2025.
- [5] Wenquan Lu, Yufei Xu, Jing Zhang, Chaoyue Wang, and Dacheng Tao. Handrefiner: Refining malformed hands in generated images by diffusion-based conditional inpainting. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 7085–7093, 2024.
- [6] Yuxin Wen, Jim Wu, Ajay Jain, Tom Goldstein, and Ashwinee Panda. Analysis of attention in video diffusion transformers. *arXiv preprint arXiv:2504.10317*, 2025.
- [7] Ailing Zeng, Yuhang Yang, Weidong Chen, and Wei Liu. The dawn of video generation: Preliminary explorations with sora-like models. *arXiv preprint arXiv:2410.05227*, 2024.
- [8] Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. Challenges and applications of large language models. *arXiv preprint arXiv:2307.10169*, 2023.
- [9] Chi Zhang, Yuanzhi Liang, Xi Qiu, Fangqiu Yi, and Xuelong Li. Vast 1.0: A unified framework for controllable and consistent video generation. *arXiv preprint arXiv:2412.16677*, 2024.
- [10] Harold Haodong Chen, Haojian Huang, Xianfeng Wu, Yexin Liu, Yajing Bai, Wen-Jie Shu, Harry Yang, and Ser-Nam Lim. Temporal regularization makes your video generator stronger. *arXiv preprint arXiv:2503.15417*, 2025.
- [11] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.
- [12] Michael S Albergo and Eric Vanden-Eijnden. Buildings normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*, 2022.
- [13] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.
- [14] Tsai-Shien Chen, Aliaksandr Siarohin, Willi Menapace, Ekaterina Deyneka, Hsiang-wei Chao, Byung Eun Jeon, Yuwei Fang, Hsin-Ying Lee, Jian Ren, Ming-Hsuan Yang, et al. Panda-70m: Captioning 70m videos with multiple cross-modality teachers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13320–13331, 2024.
- [15] Xuan Ju, Yiming Gao, Zhaoyang Zhang, Ziyang Yuan, Xintao Wang, Ailing Zeng, Yu Xiong, Qiang Xu, and Ying Shan. Miradata: A large-scale video dataset with long durations and structured captions. *Advances in Neural Information Processing Systems*, 37:48955–48970, 2024.
- [16] Zhiyu Tan, Xiaomeng Yang, Luo Zheng Qin, and Hao Li. Vidgen-1m: A large-scale dataset for text-to-video generation. *arXiv preprint arXiv:2408.02629*, 2024.
- [17] Qiuheng Wang, Yukai Shi, Jiarong Ou, Rui Chen, Ke Lin, Jiahao Wang, Boyuan Jiang, Haotian Yang, Mingwu Zheng, Xin Tao, et al. Koala-36m: A large-scale video dataset improving consistency between fine-grained conditions and video content. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 8428–8437, 2025.
- [18] Brandon Castellano. Pyscenedetect. *Last accessed*, 2020.
- [19] Irfan Maliki. Open cv. 2020.

- [20] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35:25278–25294, 2022.
- [21] 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.
- [22] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pages 402–419. Springer, 2020.
- [23] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.
- [24] 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.
- [25] Minghui Lin, Xiang Wang, Yishan Wang, Shu Wang, Fengqi Dai, Pengxiang Ding, Cunxiang Wang, Zhengrong Zuo, Nong Sang, Siteng Huang, et al. Exploring the evolution of physics cognition in video generation: A survey. *arXiv preprint arXiv:2503.21765*, 2025.
- [26] Wenxu Qian, Chaoyue Wang, Hou Peng, Zhiyu Tan, Hao Li, and Anxiang Zeng. Rdp: Real data preference optimization for physics consistency video generation. *arXiv preprint arXiv:2506.18655*, 2025.
- [27] Team Seaweed, Ceyuan Yang, Zhijie Lin, Yang Zhao, Shanchuan Lin, Zhibei Ma, Haoyuan Guo, Hao Chen, Lu Qi, Sen Wang, et al. Seaweed-7b: Cost-effective training of video generation foundation model. *arXiv preprint arXiv:2504.08685*, 2025.
- [28] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.
- [29] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.
- [30] Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024.
- [31] Xiangyu Peng, Zangwei Zheng, Chenhui Shen, Tom Young, Xinying Guo, Binluo Wang, Hang Xu, Hongxin Liu, Mingyan Jiang, Wenjun Li, et al. Open-sora 2.0: Training a commercial-level video generation model in \$200 k. *arXiv preprint arXiv:2503.09642*, 2025.
- [32] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in neural information processing systems*, 35:8633–8646, 2022.
- [33] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Fei Wu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.
- [34] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

- [35] 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, 2024. URL <https://github.com/hpcaitech/Open-Sora>, 2024.
- [36] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.
- [37] Yu Gao, Haoyuan Guo, Tuyen Hoang, Weilin Huang, Lu Jiang, Fangyuan Kong, Huixia Li, Jiashi Li, Liang Li, Xiaojie Li, et al. Seedance 1.0: Exploring the boundaries of video generation models. *arXiv preprint arXiv:2506.09113*, 2025.
- [38] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.
- [39] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.
- [40] Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in neural information processing systems*, 32, 2019.
- [41] Ge Yang, Edward Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tuning large neural networks via zero-shot hyperparameter transfer. *Advances in Neural Information Processing Systems*, 34:17084–17097, 2021.
- [42] Chenyu Zheng, Xinyu Zhang, Rongzhen Wang, Wei Huang, Zhi Tian, Weilin Huang, Jun Zhu, and Chongxuan Li. Scaling diffusion transformers efficiently via μ p. *arXiv preprint arXiv:2505.15270*, 2025.
- [43] Mohammad Samragh, Iman Mirzadeh, Keivan Alizadeh Vahid, Fartash Faghri, Minsik Cho, Moin Nabi, Devang Naik, and Mehrdad Farajtabar. Scaling smart: Accelerating large language model pre-training with small model initialization. *arXiv preprint arXiv:2409.12903*, 2024.
- [44] Sam McCandlish, Jared Kaplan, Dario Amodei, and OpenAI Dota Team. An empirical model of large-batch training. *arXiv preprint arXiv:1812.06162*, 2018.
- [45] Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24174–24184, 2024.
- [46] Yunshui Li, Yiyuan Ma, Shen Yan, Chaoyi Zhang, Jing Liu, Jianqiao Lu, Ziwen Xu, Mengzhao Chen, Minrui Wang, Shiyi Zhan, et al. Model merging in pre-training of large language models. *arXiv preprint arXiv:2505.12082*, 2025.
- [47] Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2024.
- [48] Shufan Li, Konstantinos Kallidromitis, Akash Gokul, Yusuke Kato, and Kazuki Kozuka. Aligning diffusion models by optimizing human utility. *Advances in Neural Information Processing Systems*, 37:24897–24925, 2024.
- [49] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.
- [50] Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8228–8238, 2024.
- [51] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024.

- [52] Zongyu Lin, Wei Liu, Chen Chen, Jiasen Lu, Wenze Hu, Tsu-Jui Fu, Jesse Allardice, Zhengfeng Lai, Liangchen Song, Bowen Zhang, et al. Stiv: Scalable text and image conditioned video generation. *arXiv preprint arXiv:2412.07730*, 2024.
- [53] Nan Duan Xing Chen Changyi Wan Ranchen Ming Tianyu Wang Bo Wang Zhiying Lu Aojie Li Xianfang Zeng Xinhao Zhang Gang Yu Yuhe Yin Qiling Wu Wen Sun Kang An Xin Han Deshan Sun Wei Ji Bizhu Huang Brian Li Chenfei Wu Guanzhe Huang Huixin Xiong Jiaxin He Jianchang Wu Jianlong Yuan Jie Wu Jiashuai Liu Junjing Guo Kaijun Tan Liangyu Chen Qiaohui Chen Ran Sun Shanshan Yuan Shengming Yin Sitong Liu Wei Chen Yaqi Dai Yuchu Luo Zheng Ge Zhisheng Guan Xiaoni Song Yu Zhou Binxing Jiao Jiansheng Chen Jing Li Shuchang Zhou Xiangyu Zhang Yi Xiu Yibo Zhu Heung-Yeung Shum Daxin Jiang Haoyang Huang, Guoqing Ma. Step-video-ti2v technical report: A state-of-the-art text-driven image-to-video generation model, 2025.
- [54] Peng Liu, Xiaoming Ren, Fengkai Liu, Qingsong Xie, Quanlong Zheng, Yanhao Zhang, Haonan Lu, and Yujiu Yang. Dynamic-i2v: Exploring image-to-video generaion models via multimodal llm. *arXiv preprint arXiv:2505.19901*, 2025.
- [55] Hansi Teng, Hongyu Jia, Lei Sun, Lingzhi Li, Maolin Li, Mingqiu Tang, Shuai Han, Tianning Zhang, WQ Zhang, Weifeng Luo, et al. Magi-1: Autoregressive video generation at scale. *arXiv preprint arXiv:2505.13211*, 2025.
- [56] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21807–21818, 2024.
- [57] Ziqi Huang, Fan Zhang, Xiaojie Xu, Yinan He, Jiashuo Yu, Ziyue Dong, Qianli Ma, Nattapol Chanpaisit, Chenyang Si, Yuming Jiang, et al. Vbench++: Comprehensive and versatile benchmark suite for video generative models. *arXiv preprint arXiv:2411.13503*, 2024.
- [58] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- [59] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.
- [60] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.
- [61] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.
- [62] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.
- [63] Jing Zhao, Heliang Zheng, Chaoyue Wang, Long Lan, Wanrong Huang, and Wenjing Yang. Null-text guidance in diffusion models is secretly a cartoon-style creator. In *Proceedings of the 31st ACM international conference on multimedia*, pages 5143–5152, 2023.
- [64] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.
- [65] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. 2023.
- [66] Minghui Hu, Chuanxia Zheng, Heliang Zheng, Tat-Jen Cham, Chaoyue Wang, Zuopeng Yang, Dacheng Tao, and Ponnuthurai N Suganthan. Unified discrete diffusion for simultaneous vision-language generation. *arXiv preprint arXiv:2211.14842*, 2022.
- [67] Haibin He, Xinyuan Chen, Chaoyue Wang, Juhua Liu, Bo Du, Dacheng Tao, and Qiao Yu. Diff-font: Diffusion model for robust one-shot font generation. *International Journal of Computer Vision*, 132(11):5372–5386, 2024.

- [68] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.
- [69] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *Advances in neural information processing systems*, 36:8406–8441, 2023.
- [70] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020.
- [71] Andrea Agostinelli, Timo I Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, et al. Musiclm: Generating music from text. *arXiv preprint arXiv:2301.11325*, 2023.
- [72] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023.
- [73] Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993, 2021.
- [74] Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *Advances in neural information processing systems*, 35:4328–4343, 2022.
- [75] Yuxuan Song, Zheng Zhang, Cheng Luo, Pengyang Gao, Fan Xia, Hao Luo, Zheng Li, Yuehang Yang, Hongli Yu, Xingwei Qu, et al. Seed diffusion: A large-scale diffusion language model with high-speed inference. *arXiv preprint arXiv:2508.02193*, 2025.
- [76] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022.
- [77] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.
- [78] Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. Phenaki: Variable length video generation from open domain textual description. *arXiv preprint arXiv:2210.02399*, 2022.
- [79] Yazhou Xing, Yang Fei, Yingqing He, Jingye Chen, Jiabin Xie, Xiaowei Chi, and Qifeng Chen. Large motion video autoencoding with cross-modal video vae. *arXiv preprint arXiv:2412.17805*, 2024.
- [80] Junyu Chen, Han Cai, Junsong Chen, Enze Xie, Shang Yang, Haotian Tang, Muyang Li, Yao Lu, and Song Han. Deep compression autoencoder for efficient high-resolution diffusion models. *arXiv preprint arXiv:2410.10733*, 2024.
- [81] 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.
- [82] 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.
- [83] Haoxin Chen, Menghan Xia, Yingqing He, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Jinbo Xing, Yaofang Liu, Qifeng Chen, Xintao Wang, et al. Videocrafter1: Open diffusion models for high-quality video generation. *arXiv preprint arXiv:2310.19512*, 2023.
- [84] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. *OpenAI Blog*, 1(8):1, 2024.

- [85] Google DeepMind. Veo 3. <https://deepmind.google/models/veo/>, 2025. [Online; accessed 2025-08-07].
- [86] Guoqing Ma, Haoyang Huang, Kun Yan, Liangyu Chen, Nan Duan, Shengming Yin, Changyi Wan, Ranchen Ming, Xiaoni 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.
- [87] Shoufa Chen, Chongjian Ge, Yuqi Zhang, Yida Zhang, Fengda Zhu, Hao Yang, Hongxiang Hao, Hui Wu, Zhichao Lai, Yifei Hu, et al. Goku: Flow based video generative foundation models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 23516–23527, 2025.
- [88] Yan Team. Yan: Foundational interactive video generation. *arXiv preprint arXiv:2508.08601*, 2025.
- [89] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157*, 2021.
- [90] Lijun Yu, Yong Cheng, Kihyuk Sohn, José Lezama, Han Zhang, Huiwen Chang, Alexander G Hauptmann, Ming-Hsuan Yang, Yuan Hao, Irfan Essa, et al. Magvit: Masked generative video transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10459–10469, 2023.
- [91] Dan Kondratyuk, Lijun Yu, Xiuye Gu, José Lezama, Jonathan Huang, Grant Schindler, Rachel Hornung, Vighnesh Birodkar, Jimmy Yan, Ming-Chang Chiu, et al. Videopoet: A large language model for zero-shot video generation. *arXiv preprint arXiv:2312.14125*, 2023.
- [92] Haoxin Chen, Yong Zhang, Xiaodong Cun, Menghan Xia, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter2: Overcoming data limitations for high-quality video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7310–7320, 2024.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: Yes. Our main contributions are detailed in Sec. 1.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: Yes, please see Appendix C for limitations.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[NA\]](#)

Justification: Our work primarily focuses on developing an efficient training framework for large-scale video generation models and training such models with billions of parameters. The paper does not aim to provide formal theoretical results or rigorous proofs.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [NA]

Justification: Part of our experiments are based on open-source models and benchmarks, which are straightforward to reproduce; however, some experiments involve internal data and evaluations that are less readily replicable.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: Due to corporate operations and intellectual property considerations, releasing the model and data requires sufficient authorization, which we have not yet obtained. Nevertheless, this paper provides extensive technical details and analyses to ensure transparency and facilitate understanding.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Please see Sec. 6

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: All experiments in this work are conducted through multiple rounds of generation, with results repeatedly annotated and compared. However, since it is difficult to obtain full statistical metrics for other methods, we report only average scores rather than comprehensive statistics—an approach that is also common in prior work within this field.

Guidelines:

- The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We report the training details.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: We followed the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Yes, please see Conclusion and Appendix.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [No]

Justification: We do not foresee any high risk for misuse of this work.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Yes, we credited them in appropriate ways.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: The paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [Yes]

Justification: Yes, evaluation uses human preference data annotated internally by the research team.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [No]

Justification: Human preference data for evaluation was annotated internally by the research team (no external participants; minimal risk task). Based on institutional guidelines, formal IRB approval was not required for this type of internal data collection.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [No]

Justification: LLMs were used solely for writing and editing assistance in preparing this manuscript. This usage does not constitute an important, original, or non-standard component of the core research methods, and thus declaration according to the policy is not required.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.