GREEDY DISTILL: EFFICIENT VIDEO GENERATIVE MODELING WITH LINEAR TIME COMPLEXITY

Anonymous authors
Paper under double-blind review

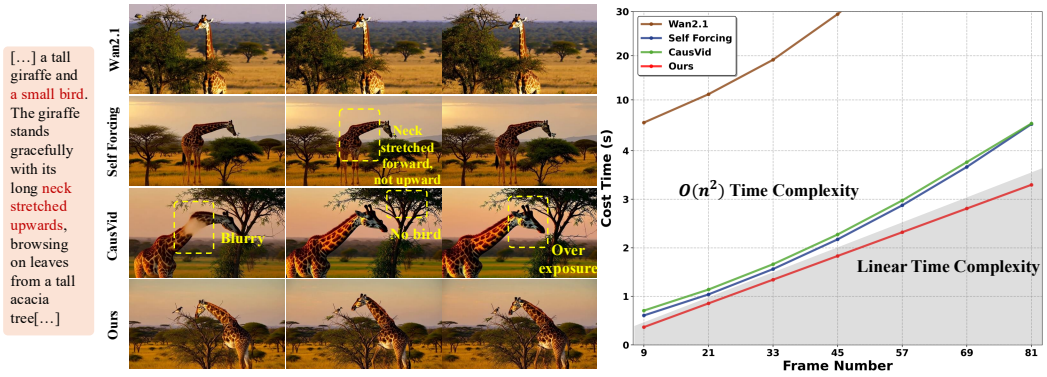


Figure 1: Comparison of results between our Greedy Distill (4 steps), the original Wan2.1 and other distill methods (left). Comparison of inference time across different methods (right) under the video synthesis configuration of 81 frames, which is measured on a single H100 GPU.

ABSTRACT

Due to bidirectional attention dependencies, video generation models generally suffer from $O(n^2)$ computational complexity. In this work, we find the "local inter-frame information redundancy" phenomenon which indicates strong local temporal dependencies in video generation, with global attention to distant frames contributing only marginally. Built upon this finding, we introduce a novel distillation training paradigm for video diffusion models, namely **GREEDY DISTILL**. Specifically, to generate the next frame using only the 0-th and the last frames, we propose the Streaming Diffusion Decoder (**SDD**) as the "Greedy Decoder" to avoid redundant computational costs from the other frames. Meanwhile, we introduce Efficient Temporal Module (**ETM**) to capture the global temporal information across frames. These two modules achieve the computational complexity reduction from $O(n^2)$ to linear. Moreover, to our knowledge, we make the **first attempt to apply RL fine-tuning** to address the error accumulation during streaming generation. Our method achieves an overall score of 84.60 on the VBench benchmark, surpassing previous state-of-the-art methods by large margins(**+4.18%**). Qualitative results also demonstrate superior performance. Leveraging its efficient model structure and KV cache, it is able to rapidly generate high-quality video streams at **24 FPS** (nearly 50% faster) on a single H100 GPU.

1 INTRODUCTION

Video generation models based on diffusion transformers (DiT) (Peebles & Xie, 2023) have achieved remarkable progress. DiT-based video generation models (Ho et al., 2022; Zhang et al.; Blattmann et al., 2023; Hu, 2024; Zheng et al., 2024b; Zhang et al., 2025; Wan et al., 2025) have made significant strides in recent advancements. However, as the demand for long-form video generation grows, the computational cost becomes a critical challenge. The slow iterative sampling process and the reliance on increasingly large denoising networks result in prohibitively high computational requirements, making practical deployment difficult. For DiT-based video generation (Hong et al., 2022; Kong et al., 2024; Yang et al., 2024; Peng et al., 2025), the computational cost is primarily

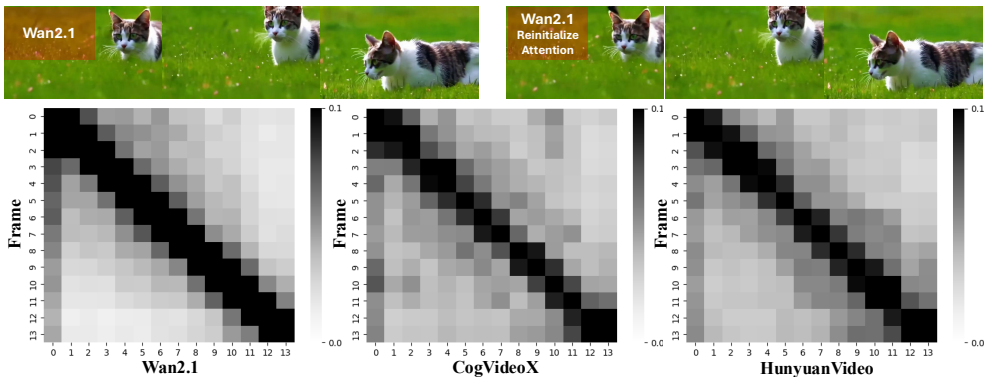


Figure 2: **Local Inter-frame Information Redundancy.**Video generation models assigns substantially higher attention to the first frame and its neighboring frames than to distant ones.

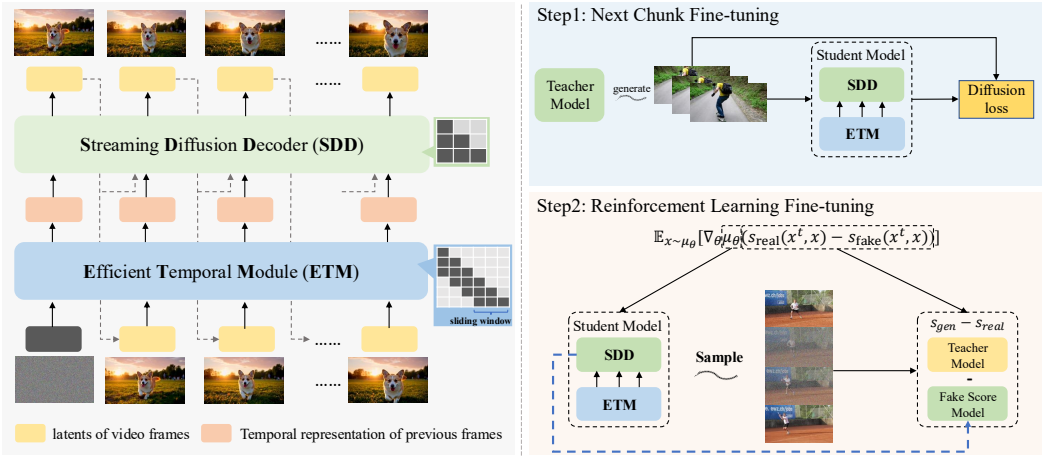
determined by the number of sampling steps (T), the number of frames (F), and the feature length in the latent space for one chunk of frames (L), leading to a total complexity of $T \times F^2 \times L^2$. As a result, generating high-quality videos necessitates substantial computational resources.

To reduce computational overhead, diffusion models for video increasingly adopt a paradigm, where a teacher model guides the training of a student (Luhman & Luhman, 2021; Liu et al., 2022; Salimans & Ho, 2022; Zheng et al., 2023; Meng et al., 2023; Liu et al., 2023). This approach is particularly effective in alleviating the slow, iterative sampling process of video diffusion models. However, existing distillation methods (Yin et al., 2024a;b;c) come with a common trade-off: they focus primarily on reducing computational costs by optimizing the sampling steps, while overlooking other computationally expensive factors. Notably, ***optimization along the Frames dimension, which contributes $O(f^2)$ complexity, remains underexplored***. Meanwhile, the recent CausVid (Yin et al., 2024c) employs an asymmetric structural distillation strategy that transfers knowledge from a bidirectional-attention teacher diffusion model to a causal student model. However, ***this approach still requires the teacher and student to be nearly isomorphic*** and mainly reduces cost by cutting sampling steps.

In our initial exploration, we observe that DiT assigns substantially higher attention to the first frame and its neighboring frames than to distant ones, as shown in Figure. 2; the same phenomenon is evident across multiple video-generation foundation models(*i.e.*Wan (Wan et al., 2025), Hunyuan Video (Kong et al., 2024), CogVideoX (Yang et al., 2024)). Consequently, in these foundation models, we reinitialize the attention matrices as constants whose attention standard variance exceeds 0.02 and observe that the generated videos are nearly indistinguishable from those of the original model. Other work (Meng et al., 2023) reports similar findings; we refer to this as "local inter-frame information redundancy" in video.

The "local inter-frame information redundancy" phenomenon indicates strong local temporal dependencies in video generation, with global attention to distant frames contributing only marginally. Building on this finding, we propose a new asymmetric structural distillation framework (**Greedy Distill**). Specifically, the student model combines an AR Transformer with a diffusion decoder. The AR Transformer adopts a chunk-wise sliding window attention (Berthelot et al., 2023) mechanism as the **E**fficient **T**emporal **M**odule (**ETM**), enabling it to capture strong local and weak global cues to form a temporal representation while avoiding the cost of full global attention. The **S**treaming **D**iffusion **D**ecoder(**SDD**) follows the diffusion paradigm to generate the next frame in a streaming manner, conditioned on the 0-th, the last frame and the temporal representation.

With causal attention and a sliding-window mechanism with window size w, the AR Transformer in ETM reduces the $(L^2) \times (F^2)$ term to $(L^2) \times w$, and unlocks streaming generation. The SDD uses the 0-th, the last frames and the temporal features of the ETM to generate the next frame, reducing the original diffusion complexity from $T \times (F^2) \times (L^2)$ to $T \times F \times (L^2)$. The total complexity becomes $(F \times w) \times (L^2) + T \times F \times (L^2) = (w + T) \times F \times (L^2)$, where L, S and w are constants, substantially lower than $T \times (F^2) \times (L^2)$. Our approach can also leverage step distillation to further reduce T, yielding additional savings. Meanwhile, to address the inevitable error accumulation, we make the first attempt to apply RL fine-tuning to address the exposure bias (Schmidt, 2019; Ning et al., 2023), where a model is trained exclusively on ground-truth context but must rely on its own



(a)Model Archhitecture (b)Training Pipeline

Figure 3: **Greedy Distill** comprises two main components: **E**fficient **T**emporal **M**odule (**ETM**) and **S**treaming **D**iffusion **D**ecoder (**SDD**) as shown (a).And training pipeline comprises two key stages: Next Block Fine-tuning and RL Fine-tuning.The score function s is defined in Sec.2.2.2.

imperfect predictions at inference time. The rollout paradigm in RL effectively tackles this issue, as policy gradients are directly applied to the model's inner predictions throughout the entire generation process, thus reducing the reliance on ground-truth context during inference.

We apply Greedy Distill to the Wan2.1 video diffusion model, reducing the latency time to 0.24 s and achieving **a speed-up of $\times$2**. On the Wan2.1 1.3B model, **inference speed reaches 24 FPS**, enabling real-time, high-fidelity video synthesis for interactive applications. Experiments show that Greedy Distill attains few-step quality comparable to the multi-step teacher while offering stronger interactivity and faster generation. To our knowledge, this is the first distillation paradigm that allows substantial architectural differences between teacher and student models.

We provide a detailed review and discussion of related work in Appendix B.1.

2 Methodology

Greedy Distill introduces a new asymmetric distillation framework 3, which distills a pretrained bidirectional video diffusion model as teacher model into an efficient student model comprising two main components: **E**fficient **T**emporal **M**odule (**ETM**) and **S**treaming **D**iffusion **D**ecoder (**SDD**). Specifically, ETM employs an autoregressive transformer with sliding window attention to capture local and global features and produces a temporal representation, while SDD follows the diffusion paradigm to generate the next frame in a streaming manner conditioned on the first frame, the last frame and the temporal representation.

The framework overview and the training pipeline is shown in Figure. 3, comprises two key stages: Next Block Fine-tuning (Sec.2.2.1) and Reinforcement Learning Fine-tuning (Sec.2.2.2). Notably, both ETM and SDD are trained via Low Rank Adaptation (LoRA) (Hu et al., 2022) layers and initialized from the teacher. We also enable efficient inference using the sliding window attention mechanism and KV caching.

2.1 Model Architecture

2.1.1 Efficient Temporal Module(ETM)

We begin by compressing the video into a latent space using a 3D VAE. The VAE encoder processes each chunk of video frames independently, compressing them into shorter chunks of latents. The decoder then reconstructs the original video frames from each latent. Our causal diffusion transformer operates in this latent space, generating latents sequentially.

Unlike common AR models (Radford et al., 2019; Brown et al., 2020), ETM employ a chunk-wise sliding window attention mechanism inspired by prior work that combines autoregressive models with diffusion (Zhen et al., 2025). Within each chunk, we apply bidirectional attention among latents. To capture both local and global dependencies while controlling computational cost, we employ sliding window attention across chunks of latents. Formally, the attention mask M of chunk-wise sliding window attention is typically defined as:

$$M_{i,j} = \begin{cases} 1, & \text{if } \left\lfloor \frac{i}{k} - w \right\rfloor \leq \left\lfloor \frac{j}{k} \right\rfloor \leq \left\lfloor \frac{i}{k} \right\rfloor \\ 0, & \text{otherwise.} \end{cases}, \tag{1}$$

where i and j index of latents of input frames, k is the chunk size, w is a fixed window size, and $\lfloor \cdot \rfloor$ denotes the floor function. ETM follows an autoregressive paradigm, compressing historical latents into a temporal representation:

$$c_f = \text{ETM}_\theta((x_0, x_1, ..., x_{f-1}), Mask = M) \tag{2}$$

where c_f denotes a temporal representation of the index of f of the latents of input frames, and x_i denotes the index of i of the latents. In this way, ETM can capture strong local and weak global cues to form a temporal representation while avoiding the cost of full global attention.

2.1.2 Streaming Diffusion Decoder(SDD)

We then apply SDD to generate the next frame in a streaming manner. Concretely, **SDD** employs DiT with Flow Matching (Lipman et al., 2022; Liu et al., 2022), which assumes that the trajectory connecting a data sample x and a noise sample ϵ in latent space follows a straight-line path:

$$x^t = (1 - t) \cdot x + t \cdot \epsilon \tag{3}$$

where $\epsilon \sim \mathcal{N}(0, I)$ and $t \in [0, 1]$. **SDD** learns a transformation from noise sample to samples drawn from the data distribution, formulated through an Ordinary Differential Equation (ODE):

$$\frac{dx^t}{dt} = \text{SDD}_\theta(x^t,\ t,\ c_f) \tag{4}$$

Here, SDD_θ represents a learnable velocity field parameterized by the model weights θ. $t \in [0, 1]$ denotes the continuous time variable and x^t refers to the data point at time t. c_f is the temporal representation generated by ETM. Putting Eq. 2 and Eq. 4 all together, the forward process is:

$$G_\theta \triangleq \hat{x}_{1:F} = \{\Psi_{T:1}(\text{SDD}_\theta,\ t,\ c_f) | f = 1, 2, ..., F\} \tag{5}$$

where G_θ denotes video generator which uses SDD_θ and c_f from ETM to autoregressively generate all latents of frames. $\hat{x}_f$ denotes the predicted latents at index f, Ψ denotes the integrator (*i.e.*UniPC Solver (Zhao et al., 2023)) , that simulates the forward diffusion process from T-th step to 0-th step to get x_0.

2.1.3 Inference Process

During inference, both SDD and ETM in our model architecture utilize the KV cache strategy to make the inference more efficient. Specifically: ① When using ETM for inference, we use the key and value of the previous w chunks to predict the next chunk. Therefore, the computational complexity of this part of our inference is $(L^2) \times w$ for one chunk, where L is the feature length in the latent space for one chunk. ② When using SDD for inference, we use the key and value of the previous chunk to predict the next chunk. Therefore, the computational complexity of this part of our inference is $(L^2) \times T$ for one chunk, where the number of sampling steps is T.

Therefore, the overall time with linear time complexity can be expressed as the sum of the two components is $(w + T) \times (L^2) \times F$, where F is the number of chunks of video, the feature length(L) is fixable for given teacher model. Finally, the inference cost of Greedy Distill is a linear time complexity that is only dependent on F. Complete description refers to the Algorithm 1.

Algorithm 1 Inference Process with KV Caching

Require: Denoising timesteps $\{t_0 = 0, t_1, \ldots, t_Q\}$, video length F, few-step autoregressive video generator G_θ, sliding window size w

Initialize KV cache $\text{C}_{\text{ETM}} \leftarrow \emptyset$, $\text{C}_{\text{SDD}} \leftarrow \emptyset$
Initialize 0-th frame: $x_0 \sim \mathcal{N}(0, I)$
for $f = 1$ to F **do**
  Generate temporal representation: $c_f = ETM_\theta(x_{f-w}, ..., x_{f-1})$ using cache C_{ETM}
  Append new KV pairs to cache C_{ETM}
  if $f > w$ **then**
    Remove oldest KV pairs
  end if
  Initialize current frame: $x_f^T \sim \mathcal{N}(0, I)$
  for $t = T$ to 1 **do**
    Generate current frame: $x_f^t = \Psi_{T:t}(\text{SDD}_\theta,\ t,\ c_f)$ using cache C_{SDD}
    if $t = T$ **then**
      Append new KV pairs to cache C_{SDD}
    end if
    if $t = 1$ **then**
      Clear KV cache C_{SDD}
    end if
  end for
  $x_f = x_f^0$
end for
Return $\{x_f\}_{f=1}^F$

2.2 TRAINING PIPELINE

2.2.1 NEXT CHUNK FINE-TUNING

We follow the common *next-token fine-tuning* paradigm in autoregression, but extend it here to *next-chunk fine-tuning*. SDD generates the next frame in a streaming manner, conditioned on the first frame, the last frame and the temporal representation from ETM. The loss function is defined as:

$$\mathcal{L}_{\text{nc}} = \mathbb{E}_{t\sim\text{Uniform}([0,1]),x\sim P}\frac{1}{F}\sum_{f=1}^{F}\left\|(\epsilon - x_f) - \text{SDD}_\theta(x_f^t,\ t,\ \text{ETM}_\theta(x_0, x_1, \ldots, x_{f-1}))\right\|^2 \tag{6}$$

2.2.2 ADDRESSING ERROR ACCUMULATION WITH REINFORCEMENT LEARNING

In experiments (Tab. 3), we observe that after *next-chunk fine-tuning*, the model already demonstrates certain generative capabilities, but it still suffers from **error accumulation**. In addition, qualitative results reveal that some frames remain insufficiently sharp after next-chunk fine-tuning.

Generally, error accumulation problem is more broadly known as exposure bias, where a model is trained exclusively on ground-truth context but must rely on its own imperfect predictions at inference time, resulting in a distributional mismatch that compounds errors as generation progresses. To mitigate this issue, we make **the first attempt to incorporate RL fine-tuning** into the distillation process. The key advantage of RL lies in its actor–critic architecture and rollout paradigm, which directly optimizes the model's own predictions across the entire generation process. By doing so, RL effectively alleviates error accumulation by reducing the model's reliance on ground-truth context at inference time, thereby addressing the exposure bias problem.

Here, we use **deterministic policy gradient** (Lillicrap et al., 2015; Fujimoto et al., 2018), where the policy gradient is defined as:

$$\nabla_\theta \mathcal{J} = -\mathbb{E}_{s\sim\rho^\mu}\left[\nabla_\theta \mu_\theta(s)\ \nabla_a Q^\mu(s,a)|_{a=\mu_\theta(s)}\right] \tag{7}$$

where μ_θ is a parameterized actor function which specifies the current policy by deterministically mapping states to a specific action. The critic $Q(s, a)$ is learned using the Bellman equation (Bellman, 1954) as in Q-learning (Watkins & Dayan, 1992).

Specifically, in our approach, we define the expected reward as the KL divergence $KL(P_{gen}(x)|P_{real}(x))$. It can be proven that (refer to the Appendix C):

$$\nabla_\theta \mathcal{J} \propto -\mathbb{E}_{x_t\sim\rho^\mu}\left[\nabla_\theta \mu_\theta(x_t)\cdot \mathbb{E}_{p_{fake}}\nabla_{x_{t-1}}\log\frac{p_{\text{fake}}(x_{t-1}|x_t)}{p_{\text{real}}(x_{t-1}|x_t)}\right] \tag{8}$$

Following the derivations in DMD (Yin et al., 2024b), we rewrite $\nabla_{x_{t-1}}\log\frac{p_{\text{fake}}(x_{t-1}|x_t)}{p_{\text{real}}(x_{t-1}|x_t)}$ as $s_{gen}(x_{t-1}, t-1) - s_{real}(x_{t-1}, t-1))$, where s denotes the score function. In practice, we approximate the score function by the velocity field ψ_θ as DMD. Combining Equations 8 and 5, the overall loss of reinforcement learning fine-tuning is:

$$\nabla_\theta \mathcal{J} \propto -\mathbb{E}_{x\sim\mu_\theta}\left[\nabla_\theta \mu_\theta(x_t^f)\,(s_{gen}(x_{t-1}, t-1) - s_{real}(x_{t-1}, t-1))\right] \tag{9}$$

where $x_{t-1} = \mu_\theta(x_t^f) = \Psi_{T:t}(\text{SDD}_\theta(x_f^t,\ t,\ \text{ETM}_\theta(x_0, x_1, \ldots, x_{f-1})))$, s_{gen}, s_{real} are score function, while s_{gen} is provided by the SDD_θ and s_{real} corresponds to the teacher model. The RL fine-tuning loss $\mathcal{L}$ is formulated as:

$$\mathcal{L}_{RL} = \mathbb{E}_{x\sim\mu_\theta, f\in[1,2,\ldots F], t\in\mathcal{T}}\text{MSE}\big[\mu_\theta(x_t^f) - \text{ sg}[\mu_\theta(x_t^f) - (s_{real}(x_{t-1}^f, t-1) - s_{gen}(x_{t-1}^f, t-1))]\big] \tag{10}$$

where $\mathcal{T}$ denotes the set of steps in our distillation objective which is set as $[1000, 750, 500, 250]$, MSE denotes Mean Squared Error, and sg denotes the stop-gradient operation.

The key difference between our method and DMD is that DMD's score function relies on training another network, while ours is derived from SDD_θ. Additionally, DMD's critic model update incurs high computational costs, whereas our method updates both the policy and critic models simultaneously without additional training, significantly reducing computational costs. The overall training of our framework is done in two steps, the model is optimized with $\mathcal{L}_{\text{nc}}$ at the first stage while the $\mathcal{L}_{RL}$ is used at the second stage for further fine-tuning.

3 EXPERIMENT

3.1 SETUP

Implementation. Our teacher model is based on the bidirectional DiT architecture of Wan2.1, which processes video data in the latent space. A 3D VAE is used to compress video frames into latents. The student model adopts our Greedy Distill architecture, consisting of ETM and SDD. Both components are initialized from the Wan2.1 and fine-tuned via Low Rank Adaptation (LoRA) (Hu et al., 2022). The key distinction is that ETM is a causal attention mechanism combined with a sliding-window mechanism to capture the global temporal information across frames, which substantially reduces computational cost. The temporal representation produced by ETM is able to attend not only to the current window but also to information from preceding windows, thereby balancing efficiency with temporal context modeling.

Training and Inference. Our training process consists of two stages: Next Block Fine-tuning (Sec.2.2.1) and Reinforcement Learning Fine-tuning (Sec.2.2.2). The prompts for training are taken from OpenVidHD (Nan et al., 2024), while the video data are sampled from the teacher model, i.e., WAN 2.1 (Wan et al., 2025).

The model is trained on 64 NVIDIA H100 GPUs using the AdamW optimizer. The input video resolution is set to 832×480, and each video contains between 81 and 301 frames. In the Next Block Fine-tuning stage, the student model is trained for 10 epochs with a batch size of 1024 and a learning rate of $1 \times e^{-5}$.In the Reinforcement Learning Fine-tuning stege, the student model is trained for 3 epochs with a batch size of 256 and a learning rate of $2 \times e^{-6}$.

Evaluation metrics. We conduct a comprehensive evaluation of our method using VBench (Huang et al., 2024) as the primary metric. We rewrite the test prompts using Qwen/Qwen2.5-7B-Instruct (Team, 2024), a practice that has already been widely adopted in prior work(*i.e.*, Self Forcing (Huang et al., 2025) and OpenVid (Nan et al., 2024)). In addition, we perform human evaluation to assess the perceptual quality of the generated videos. We compare our approach against state-of-the-art distillation methods from multiple perspectives, and the aggregated results demonstrate that our method consistently outperforms competing approaches.

Table 1: We compare Greedy Distill with representative open-source video generation models of similar parameter sizes and resolutions.

Model	#Params	Resolution	Throughput (FPS) ↑	Latency (s) ↓	Evaluation scores ↑		
					Total Score	Quality Score	Semantic Score
Diffusion models							
LTX-Video (HaCohen et al., 2024)	1.9B	768×512	8.98	13.5	80.00	82.30	70.79
Wan2.1 (Wan et al., 2025)	1.3B	832×480	0.78	103	84.26	85.30	80.09
Chunk-wise autoregressive models							
SkyReels-V2 (Chen et al., 2025b)	1.3B	960×540	0.49	112	82.67	84.70	74.53
CausVid (Yin et al., 2025)*	1.3B	832×480	17.0	0.69	81.20	84.05	69.80
Self Forcing(chunk-wise) (Huang et al., 2025)	1.3B	832×480	17.0	0.69	84.31	85.07	81.28
Frame-wise Autoregressive models							
NOVA (Deng et al., 2024)	0.6B	768×480	0.88	4.1	80.12	80.39	79.05
Pyramid Flow (Jin et al., 2024)	2B	640×384	6.7	2.5	81.72	84.74	69.62
Self Forcing(frame-wise) (Huang et al., 2025)	1.3B	832×480	8.9	0.45	84.26	85.25	80.30
Greedy Distill (Ours)	1.3B	832×480	**24.0**	**0.24**	**84.60**	**85.37**	**81.52**

3.2 Main Results

Quantitative Comparison. To ensure fairness, all experimental results are obtained from the same model scale, *i.e.*, 1.3B-2.0B. Table 1 presents a comprehensive comparison between Greedy Distill and existing state-of-the-art methods. Obviously, our method achieves the **highest VBench score**, while meets the **real-time** requirements, *i.e.*, 24 FPS in Throughput and a Latency Time of only 0.24 seconds. Moreover, Greedy Distill maintains generation quality comparable to that of the teacher model (**85.60** vs. 85.26 of Wan2.1-1.3B).

Qualitative Comparison. We compare the videos generated by our method with those from CausVid (Yin et al., 2025) and Self Forcing (Huang et al., 2025), as in Figure. 10. The results indicate that CausVid and Self Forcing suffers from inconsistencies with physical dynamics and CausVid in particular is prone to error accumulation, whereas Greedy Distill maintains high-quality video generation while ensuring fast inference speed, effectively avoiding the error accumulation problem and adhering more closely to physical principles.

Specifically, as shown in Figure. 10 (a) and Figure. 10 (b), CausVid and Self Forcing produce object details that deviate from their natural properties (e.g., unrealistic cycling postures and incorrect numbers of cat tails). In Figure. 10 (c) andFigure. 10 (d), the generated videos from CausVid and Self Forcing violate physical dynamics (e.g., a ball remaining static in midair or an airplane following an implausible curved trajectory during landing). In contrast, our method does not suffer from these issues, producing results that are both physically consistent and visually coherent.

User Study. To further evaluate the perceptual quality of videos generated by our method, we conducted a human assessment study. For each model, we selected 40 video samples of varying durations to reflect challenges across different video lengths: 15 short clips (0–5 seconds), 15 medium-length clips (5–10 seconds), and 10 long clips (10–18 seconds). The evaluation involved 60 participants (50% aged 18–30, 30% aged 30–40, and 20% aged 40–50; 41.7% male, 58.3% female). Each participant was presented with a text prompt alongside videos generated by different models, with all videos displayed in random order to minimize ordering bias. Following prior work (Kong et al., 2024), participants were asked to select the video they perceived as better in terms of text alignment, motion quality, and visual quality. As shown in Fig. 5, Our videos are preferred over others, especially in text alignment and motion quality.

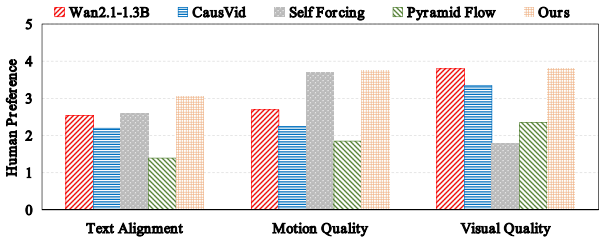


Figure 5: **User preference study.** Greedy Distill outperforms all baselines in human preference.

3.3 Ablation Study

Long Video Generation. To demonstrate that our method's ability in mitigating the error accumulation problem, we conduct long-video generation experiments (*i.e.*, 18s). From the qualitative results(*i.e.*, Figure. 7) on long video generation, we also observe that both CausVid and Self-Forcing

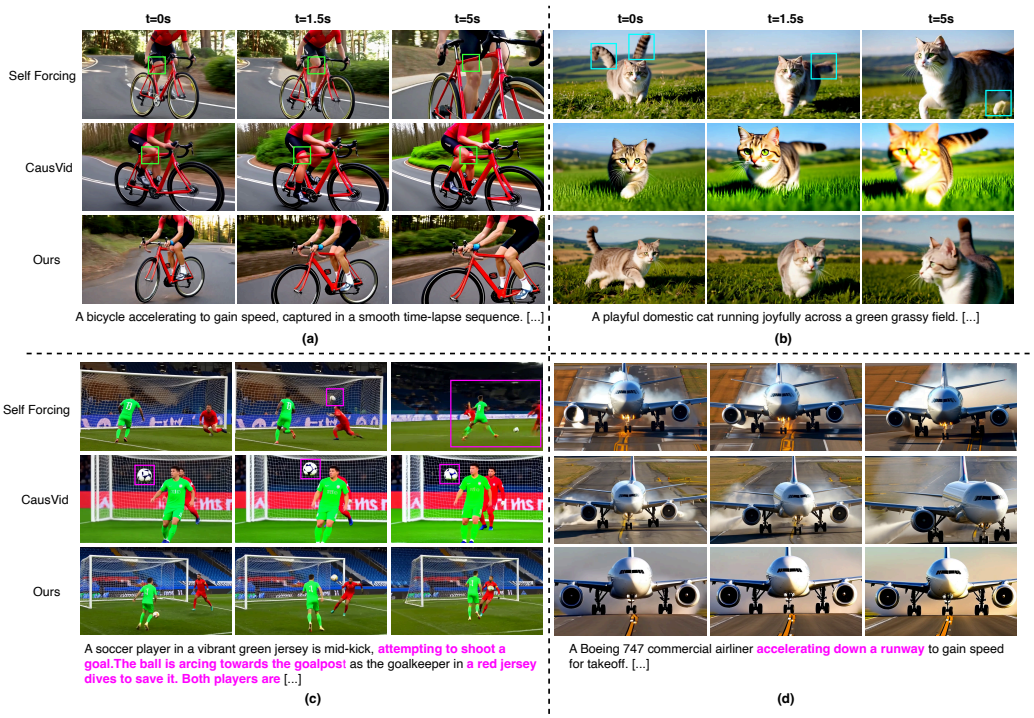


Figure 4: **Qualitative comparisons** between Greedy Distill and Self Forcing(chunk-wise), CausVid. All models are distilled from the same teacher model of Wan2.1.***Videos are available in the supplementary materials.***

Table 2: Evaluation of long video generation.

Method	Throughput (FPS)	Temporal Coherence	Frame Quality	Semantic Alignment
Streaming T2V (Henschel et al., 2025)	0.5	88.9	45.3	27.1
CausVid (Yin et al., 2025)	17.0	88.5	60.1	25.8
Self Forcing (Huang et al., 2025)	17.0	90.3	61.3	26.9
Greedy Distill (Ours)	24.0	**94.2**	**61.7**	**27.7**

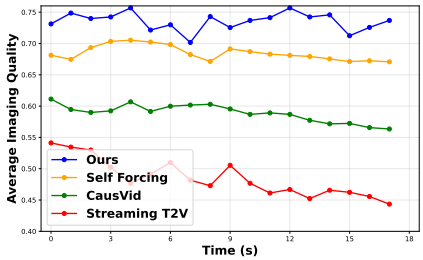


Figure 6: Imaging Quality scores show that **Greedy Distill** maintains stable and superior image quality in long video.

suffer from error accumulation and produce videos with low dynamics or static content in the later segments. For example, after the 9-second mark, the main objects in videos generated by CausVid and Self-Forcing either remain motionless or exhibit only minor movements. In addition, these methods suffer from over-exposure artifacts caused by accumulated errors. Our method shows a clear advantage(*i.e.*Table. 2) on long-video evaluation with VBench, particularly in the **Temporal Coherence** metric, where it achieves an improvement of **4.3%**. This demonstrates that our approach not only maintains frame-level fidelity but also better preserves consistency across extended time horizons, effectively addressing the error accumulation and stagnation issues observed in prior methods, as shown in Fig. 6,

Differences between Greedy Distill and previous approaches. We compare Greedy Distill with prior approaches in two aspects. ① **Teacher-Student Model Architecture:** As in Figure 8, previous methods use nearly identical teacher and student models, with only a causal attention mechanism, incurring significant computational overhead of $T \times w \times F \times L^2$, where L is the sequence length. In contrast, Greedy Distill introduces a novel student architecture and reduce complexity to $(T + w) \times F \times L^2$. The ETM effectively captures global features for temporal representations while avoiding the high cost of global attention. ② Previous methods generally adopt a **chunk-wise inference** strategy, outputting results every 3 chunks and resetting the context every 7 chunks. This design, however, leads to severe error accumulation and temporal artifacts. While some approaches attempt to mitigate these issues through diffusion forcing (*i.e.*CausVid, SkyReels-V2) or self forcing (*i.e.*Self Forcing), the problems remain significant (shown in Figure. 7). In contrast, our method leverages a

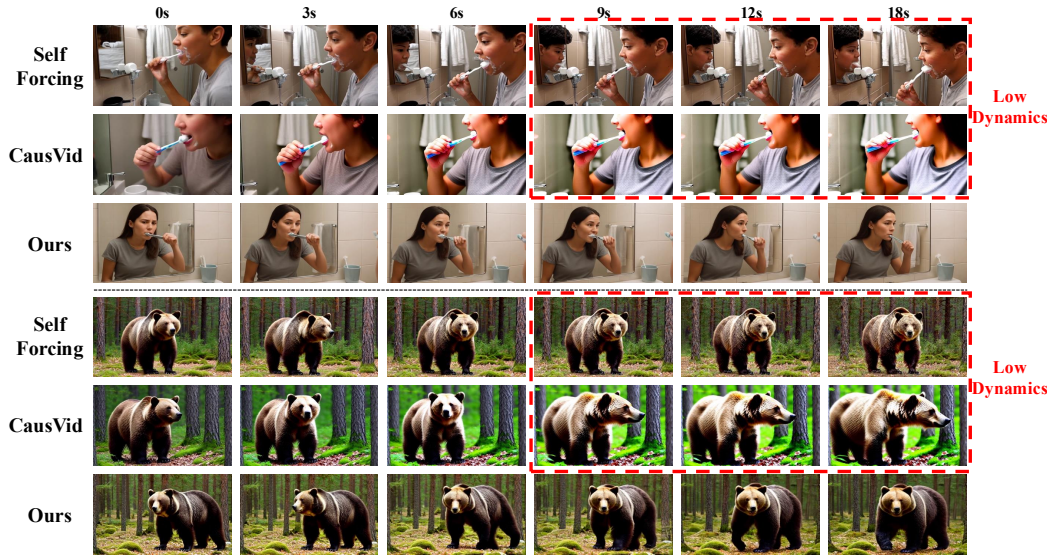


Figure 7: Qualitative results on long video generation show that our method avoids error accumulation and low-dynamics issues.***Videos are available in the supplementary materials.***

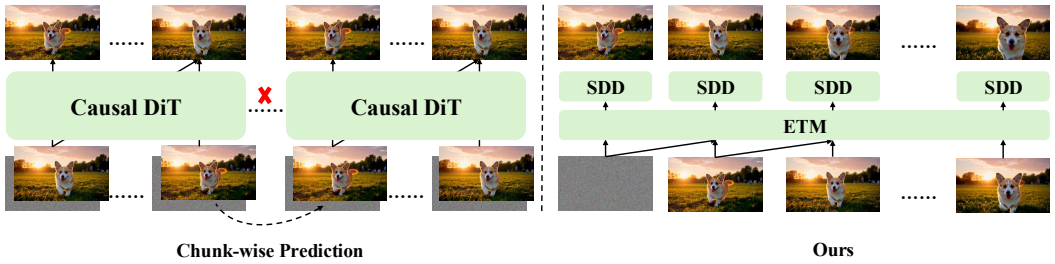


Figure 8: Differences between Greedy Distill and prior approaches.

ETM module, which allows the decoder to access global information, effectively avoiding the error accumulation problem and adhering more closely to physical principles.

Necessity of ETM and RL Fine-tuning. We investigate the need for ETM in the student model of Greedy Distill and Reinforcement Learning Fine-tuning, as shown in Table 3. Removing ETM and performing direct distillation (similar to CausVid and Self-Forcing with a chunk size of 1) reveals the limitations of using only a diffusion decoder in the student model. Introducing ETM with sliding-window attention improves Quality Score (+3.3%). Finally, RL Fine-tuning yields the best results, improving Quality Score by 5.73% and mitigating error accumulation. Meanwhile, the training loss curve of RL Fine-tuning, which corresponds to the action-value, is illustrated in Fig. 9.

Table 3: **Ablation studies.** We compares different components in our distillation framework..The last row is our final configuration

RL	ETM	Total Score	Quality Score	Semantic Score
✗	✗	81.01	80.93	81.32
✗	✓	83.2	83.63	81.49
✓	✓	**84.60**	**85.37**	**81.52**

4 CONCLUSION

In this paper, we present GREEDY DISTILL, a novel distillation training paradigm for autoregressive video diffusion models. It employs the Streaming Diffusion Decoder (SDD) to generate intermediate frames using only the 0-th and the last frames, avoiding redundant computation, and the Efficient Temporal Module (ETM) to capture global temporal dependencies. GREEDY DISTILL reduces the computational complexity from $O(n^2)$ to linear. Moreover, for the first time, we applies reinforcement learning fine-tuning to mitigate error accumulation in streaming generation. Our approach achieves strong improvements in both real-time and long-duration video generation.

REPRODUCIBILITY STATEMENT

This statement presents a comprehensive report detailing the reproduction process for our Greedy Distill, a distillation training paradigm for autoregressive video diffusion models. The implementation builds upon Diffusers's code base and integrates components from additional open-source libraries, to which we extend our gratitude.

KEY IMPLEMENTATION DETAILS

- **Code Base**: In terms of implementation, we build upon the **Diffusers** project as our code base, using /src/diffusers/models/transformers/transformer_wan.py as the foundation for the DiT architecture. For Low-Rank Adaptation (LoRA), we leverage the open-source **PEFT** library as an additional component.
- **SDD Implementation**: The SDD is initialized from Wan2.1 and fine-tuned using Low-Rank Adaptation (LoRA). Furthermore, the teacher model's global attention is replaced with **causal attention**, enabling streaming inference and leveraging KV cache to improve inference efficiency.
- **ETM Implementation**: The ETM is initialized from Wan2.1 and fine-tuned using Low-Rank Adaptation (LoRA). In this process, the teacher model's global attention is replaced with **causal sliding-window attention** (window size set to 3), enabling streaming inference and leveraging KV cache to further improve efficiency.
- **RL Fine-tuning Implementation**: In the RL Fine-tuning stage, we randomly sample $f \in [1, 2, ...F], t \in \mathcal{T}$, compute the loss according to Eq. 10, and adopt a smaller learning rate. Notably, in distributed training, to improve GPU utilization, we enforce the same f and t within each minibatch. The student model is trained for 3 epochs with a batch size of 256 and a learning rate of $2 \times e^{-6}$. The training loss curve is illustrated in Fig. 9.

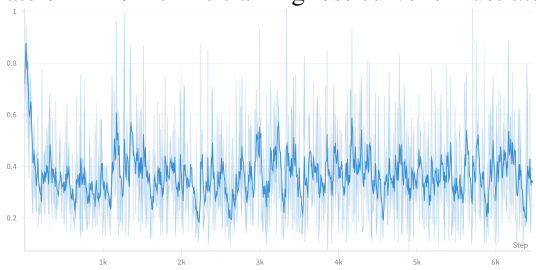


Figure 9: The training loss curve in RL Fine-tuning.

RESULTS

Using the aforementioned process, we successfully distill the Wan2.1 model into a student model that meets real-time requirements, achieving 24 FPS throughput and a VBench score of 84.60.

CONCLUSION

This reproduction report documents the end-to-end procedure for replicating Greedy Distill on Wan2.1, covering codebase choices (Diffusers/PEFT), module initialization (SDD/ETM via LoRA), and training protocols (Next-Block and RL fine-tuning). We provide exact hyperparameters, data sources, and inference settings—including causal sliding-window attention with KV cache—to enable faithful re-creation. Our reimplementation attains 24 FPS throughput and a VBench score of 84.60, validating the method's reproducibility. We release scripts and configuration files to streamline replication and adaptation to other backbones or hardware budgets. These artifacts offer a robust foundation for advancing efficient, real-time, long-duration video generation research.

REFERENCES

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos J Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in atari. *Advances in Neural Information Processing Systems*, 37:58757–58791, 2024.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

Richard Bellman. The theory of dynamic programming. *Bulletin of the American Mathematical Society*, 60(6):503–515, 1954.

David Berthelot, Arnaud Autef, Jierui Lin, Dian Ang Yap, Shuangfei Zhai, Siyuan Hu, Daniel Zheng, Walter Talbott, and Eric Gu. Tract: Denoising diffusion models with transitive closure time-distillation. *arXiv preprint arXiv:2303.04248*, 2023.

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

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024.

Dar-Yen Chen, Hmrishav Bandyopadhyay, Kai Zou, and Yi-Zhe Song. Nitrofusion: High-fidelity single-step diffusion through dynamic adversarial training. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 7654–7663, 2025a.

Guibin Chen, Dixuan Lin, Jiangping Yang, Chunze Lin, Junchen Zhu, Mingyuan Fan, Hao Zhang, Sheng Chen, Zheng Chen, Chengcheng Ma, et al. Skyreels-v2: Infinite-length film generative model. *arXiv preprint arXiv:2504.13074*, 2025b.

Haoge Deng, Ting Pan, Haiwen Diao, Zhengxiong Luo, Yufeng Cui, Huchuan Lu, Shiguang Shan, Yonggang Qi, and Xinlong Wang. Autoregressive video generation without vector quantization. *arXiv preprint arXiv:2412.14169*, 2024.

Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pp. 1587–1596. PMLR, 2018.

Kaifeng Gao, Jiaxin Shi, Hanwang Zhang, Chunping Wang, and Jun Xiao. Vid-gpt: Introducing gpt-style autoregressive generation in video diffusion models. *arXiv preprint arXiv:2406.10981*, 2024.

Songwei Ge, Thomas Hayes, Harry Yang, Xi Yin, Guan Pang, David Jacobs, Jia-Bin Huang, and Devi Parikh. Long video generation with time-agnostic vqgan and time-sensitive transformer. In *European Conference on Computer Vision*, pp. 102–118. Springer, 2022.

Jiatao Gu, Shuangfei Zhai, Yizhe Zhang, Lingjie Liu, and Joshua M Susskind. Boot: Data-free distillation of denoising diffusion models with bootstrapping. In *ICML 2023 Workshop on Structured Probabilistic Inference $\{\backslash\&\}$ Generative Modeling*, volume 3, 2023.

Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, et al. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024.

Roberto Henschel, Levon Khachatryan, Hayk Poghosyan, Daniil Hayrapetyan, Vahram Tadevosyan, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Streamingt2v: Consistent, dynamic, and extendable long video generation from text. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 2568–2577, 2025.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

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.

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

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Li Hu. Animate anyone: Consistent and controllable image-to-video synthesis for character animation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8153–8163, 2024.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *arXiv preprint arXiv:2506.08009*, 2025.

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*, pp. 21807–21818, 2024.

Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. *arXiv preprint arXiv:2410.05954*, 2024.

Minguk Kang, Richard Zhang, Connelly Barnes, Sylvain Paris, Suha Kwak, Jaesik Park, Eli Shechtman, Jun-Yan Zhu, and Taesung Park. Distilling diffusion models into conditional gans. In *European Conference on Computer Vision*, pp. 428–447. Springer, 2024.

Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ode trajectory of diffusion. *arXiv preprint arXiv:2310.02279*, 2023.

Jihwan Kim, Junoh Kang, Jinyoung Choi, and Bohyung Han. Fifo-diffusion: Generating infinite videos from text without training. *Advances in Neural Information Processing Systems*, 37: 89834–89868, 2024.

Akio Kodaira, Chenfeng Xu, Toshiki Hazama, Takanori Yoshimoto, Kohei Ohno, Shogo Mitsuhori, Soichi Sugano, Hanying Cho, Zhijian Liu, and Kurt Keutzer. Streamdiffusion: A pipeline-level solution for real-time interactive generation. *arXiv preprint arXiv:2312.12491*, 2023.

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.

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.

Feng Liang, Akio Kodaira, Chenfeng Xu, Masayoshi Tomizuka, Kurt Keutzer, and Diana Marculescu. Looking backward: Streaming video-to-video translation with feature banks. *arXiv preprint arXiv:2405.15757*, 2024.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Shanchuan Lin, Xin Xia, Yuxi Ren, Ceyuan Yang, Xuefeng Xiao, and Lu Jiang. Diffusion adversarial post-training for one-step video generation. *arXiv preprint arXiv:2501.08316*, 2025.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Hongjian Liu, Qingsong Xie, Tianxiang Ye, Zhijie Deng, Chen Chen, Shixiang Tang, Xueyang Fu, Haonan Lu, and Zheng-Jun Zha. Scott: Accelerating diffusion models with stochastic consistency distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 5451–5459, 2025.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, et al. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In *The Twelfth International Conference on Learning Representations*, 2023.

Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models. *arXiv preprint arXiv:2410.11081*, 2024.

Eric Luhman and Troy Luhman. Knowledge distillation in iterative generative models for improved sampling speed. *arXiv preprint arXiv:2101.02388*, 2021.

Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023a.

Simian Luo, Yiqin Tan, Suraj Patil, Daniel Gu, Patrick von Platen, Apolinário Passos, Longbo Huang, Jian Li, and Hang Zhao. Lcm-lora: A universal stable-diffusion acceleration module. *arXiv preprint arXiv:2311.05556*, 2023b.

Weijian Luo, Tianyang Hu, Shifeng Zhang, Jiacheng Sun, Zhenguo Li, and Zhihua Zhang. Diff-instruct: A universal approach for transferring knowledge from pre-trained diffusion models. *Advances in Neural Information Processing Systems*, 36:76525–76546, 2023c.

Weijian Luo, Zemin Huang, Zhengyang Geng, J Zico Kolter, and Guo-jun Qi. One-step diffusion distillation through score implicit matching. *Advances in Neural Information Processing Systems*, 37:115377–115408, 2024a.

Yihong Luo, Xiaolong Chen, Xinghua Qu, Tianyang Hu, and Jing Tang. You only sample once: Taming one-step text-to-image synthesis by self-cooperative diffusion gans. *arXiv preprint arXiv:2403.12931*, 2024b.

Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14297–14306, 2023.

Kepan Nan, Rui Xie, Penghao Zhou, Tiehan Fan, Zhenheng Yang, Zhijie Chen, Xiang Li, Jian Yang, and Ying Tai. Openvid-1m: A large-scale high-quality dataset for text-to-video generation. *arXiv preprint arXiv:2407.02371*, 2024.

Mang Ning, Mingxiao Li, Jianlin Su, Albert Ali Salah, and Itir Onal Ertugrul. Elucidating the exposure bias in diffusion models. *arXiv preprint arXiv:2308.15321*, 2023.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

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 $200k$. *arXiv preprint arXiv:2503.09642*, 2025.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

David Ruhe, Jonathan Heek, Tim Salimans, and Emiel Hoogeboom. Rolling diffusion models, 2024. URL `https://arxiv.org/abs/2402.09470`.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

Axel Sauer, Frederic Boesel, Tim Dockhorn, Andreas Blattmann, Patrick Esser, and Robin Rombach. Fast high-resolution image synthesis with latent adversarial diffusion distillation. In *SIGGRAPH Asia 2024 Conference Papers*, pp. 1–11, 2024a.

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *European Conference on Computer Vision*, pp. 87–103. Springer, 2024b.

Florian Schmidt. Generalization in generation: A closer look at exposure bias. *arXiv preprint arXiv:1910.00292*, 2019.

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

Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. *arXiv preprint arXiv:2310.14189*, 2023.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. 2023.

Qwen Team. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2, 2024.

Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. *arXiv preprint arXiv:2408.14837*, 2024.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Yuqing Wang, Tianwei Xiong, Daquan Zhou, Zhijie Lin, Yang Zhao, Bingyi Kang, Jiashi Feng, and Xihui Liu. Loong: Generating minute-level long videos with autoregressive language models. *arXiv preprint arXiv:2410.02757*, 2024.

Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3):279–292, 1992.

Wenming Weng, Ruoyu Feng, Yanhui Wang, Qi Dai, Chunyu Wang, Dacheng Yin, Zhiyuan Zhao, Kai Qiu, Jianmin Bao, Yuhui Yuan, et al. Art-v: Auto-regressive text-to-video generation with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7395–7405, 2024.

Chenfei Wu, Jian Liang, Xiaowei Hu, Zhe Gan, Jianfeng Wang, Lijuan Wang, Zicheng Liu, Yuejian Fang, and Nan Duan. Nuwa-infinity: Autoregressive over autoregressive generation for infinite visual synthesis. *arXiv preprint arXiv:2207.09814*, 2022.

Yecheng Wu, Zhuoyang Zhang, Junyu Chen, Haotian Tang, Dacheng Li, Yunhao Fang, Ligeng Zhu, Enze Xie, Hongxu Yin, Li Yi, et al. Vila-u: a unified foundation model integrating visual understanding and generation. *arXiv preprint arXiv:2409.04429*, 2024.

Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion gans. *arXiv preprint arXiv:2112.07804*, 2021.

Desai Xie, Zhan Xu, Yicong Hong, Hao Tan, Difan Liu, Feng Liu, Arie Kaufman, and Yang Zhou. Progressive autoregressive video diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 6322–6332, 2025.

Zhening Xing, Gereon Fox, Yanhong Zeng, Xingang Pan, Mohamed Elgharib, Christian Theobalt, and Kai Chen. Live2diff: Live stream translation via uni-directional attention in video diffusion models. *arXiv preprint arXiv:2407.08701*, 2024.

Yanwu Xu, Yang Zhao, Zhisheng Xiao, and Tingbo Hou. Ufogen: You forward once large scale text-to-image generation via diffusion gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8196–8206, 2024.

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

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.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and Bill Freeman. Improved distribution matching distillation for fast image synthesis. *Advances in neural information processing systems*, 37:47455–47487, 2024a.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6613–6623, 2024b.

Tianwei Yin, Qiang Zhang, Richard Zhang, William T Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast causal video generators. *arXiv e-prints*, pp. arXiv–2412, 2024c.

Tianwei Yin, Qiang Zhang, Richard Zhang, William T Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 22963–22974, 2025.

Y Zhang, Y Wei, D Jiang, X Zhang, W Zuo, and Q Tian. Controlvideo: Training-free controllable text-to-video generation. arxiv 2023. *arXiv preprint arXiv:2305.13077*.

Yabo Zhang, Xinpeng Zhou, Yihan Zeng, Hang Xu, Hui Li, and Wangmeng Zuo. Framepainter: Endowing interactive image editing with video diffusion priors. *arXiv preprint arXiv:2501.08225*, 2025.

Zhicheng Zhang, Junyao Hu, Wentao Cheng, Danda Paudel, and Jufeng Yang. Extdm: Distribution extrapolation diffusion model for video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19310–19320, 2024.

Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. *Advances in Neural Information Processing Systems*, 36:49842–49869, 2023.

Dingcheng Zhen, Qian Qiao, Tan Yu, Kangxi Wu, Ziwei Zhang, Siyuan Liu, Shunshun Yin, and Ming Tao. Marrying autoregressive transformer and diffusion with multi-reference autoregression. *arXiv preprint arXiv:2506.09482*, 2025.

Hongkai Zheng, Weili Nie, Arash Vahdat, Kamyar Azizzadenesheli, and Anima Anandkumar. Fast sampling of diffusion models via operator learning. In *International conference on machine learning*, pp. 42390–42402. PMLR, 2023.

Jianbin Zheng, Minghui Hu, Zhongyi Fan, Chaoyue Wang, Changxing Ding, Dacheng Tao, and Tat-Jen Cham. Trajectory consistency distillation. *CoRR*, 2024a.

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. *arXiv preprint arXiv:2412.20404*, 2024b.

Mingyuan Zhou, Huangjie Zheng, Zhendong Wang, Mingzhang Yin, and Hai Huang. Score identity distillation: Exponentially fast distillation of pretrained diffusion models for one-step generation. In *Forty-first International Conference on Machine Learning*, 2024.

A THE USE OF LARGE LANGUAGE MODELS(LLMS)

In preparing this paper, large language models (LLMs) were used solely for language refinement, such as improving grammar, clarity, and fluency. All research questions, conceptual and theoretical frameworks, methodology, data analysis, and conclusions were developed and carried out independently by the author. The LLMs did not generate or influence any core ideas, interpretations, or findings. Their role was limited to enhancing readability while preserving the originality and integrity of the scholarly work.

B RELATED WORK

B.1 AUTOREGRESSIVE VIDEO GENERATION

Autoregressive video generation aims to synthesize videos frame by frame along the temporal dimension, thereby achieving lower latency and improved temporal coherence. Inspired by the remarkable success of large language models (LLMs) (Bai et al., 2023) in natural language processing, early studies (Yan et al., 2021; Ge et al., 2022; Wu et al., 2022; Kondratyuk et al., 2023; Wang et al., 2024; Wu et al., 2024) encoded videos into discrete tokens and employed autoregressive Transformers to generate video tokens sequentially. More recently, diffusion models (Ho et al., 2020; Song et al., 2020; Lipman et al., 2022) have achieved significant advances in video generation and have been widely adopted in this domain. Some works (Alonso et al., 2024; Jin et al., 2024; Valevski et al., 2024; Zhang et al., 2024; Chen et al., 2024; Kim et al., 2024; Ruhe et al., 2024) train diffusion models to denoise new frames conditioned on the given context frames, thereby enabling autoregressive generation. Even more recently, a line of research has explored leveraging pre-trained text-to-image (Kodaira et al., 2023; Liang et al., 2024; Valevski et al., 2024; Weng et al., 2024) or text-to-video (Gao et al., 2024; Kim et al., 2024; Xing et al., 2024; Xie et al., 2025; Henschel et al., 2025) models and adapting them to perform autoregressive next-frame generation conditioned on context frames. Our approach is closely related to this research direction. The key difference is that we propose an innovative adaptation method via diffusion distillation. This method not only significantly improves efficiency but also makes autoregressive methods competitive with bidirectional diffusion in video generation.

B.2 DIFFUSION MODEL DISTILLATION

Diffusion distillation aims to distill knowledge from a pre-trained teacher diffusion model to a student diffusion model, enabling the student to generate samples in fewer steps and thereby reducing inference costs. According to the distillation mechanism, previous studies can be roughly divided into two categories: trajectory-preserving distillation and Distribution-matching distillation. Trajectory-preserving distillation aims to predict the ordinary differential equation (ODE) trajectory of the teacher model with fewer steps. Luhman & Luhman (2021); Zheng et al. (2023) trained the student model on noise–image pairs precomputed by the teacher model using an ODE solver. Progressive distillation (Meng et al., 2023; Salimans & Ho, 2022) trains a series of student models, iteratively halving the number of sampling steps at each stage to reduce the total number of steps required. instaflow (Liu et al., 2022; 2023) uses a reflow-based distillation approach to align noise and image mappings, enabling accurate one-step generation. Consistency Distillation (Luo et al., 2023a;b; Song & Dhariwal, 2023; Song et al., 2023; Gu et al., 2023; Berthelot et al., 2023; Kim et al., 2023; Zheng et al., 2024a; Lu & Song, 2024; Liu et al., 2025) trains student models to produce outputs aligned with the teacher at all timesteps along the ODE trajectory, thereby achieving self-consistency. Unlike trajectory-preserving distillation, which imposes constraints via ODE paths, distribution-matching distillation supervises at the distributional level, aligning the output distributions of the student and teacher models. Some approaches (Xiao et al., 2021; Kang et al., 2024; Luo et al., 2024b; Sauer et al., 2024a;b; Xu et al., 2024; Chen et al., 2025a; Lin et al., 2025) reduce the distribution discrepancy through adversarial training, while others (Luo et al., 2023c; 2024a; Yin et al., 2024a;b;c; Zhou et al., 2024) achieve this via score-based distillation.

In contrast to aforementioned approaches that primarily focus on distillation through reducing the number of generation steps, we propose a novel student distillation architecture. Our approach not only reduces the number of generation steps but also decreases the computational complexity along

the frame dimension from quadratic to linear. In particular, compared with the asymmetric strategy adopted by CausVid (Yin et al., 2025), which distills a bidirectional teacher into a unidirectional student, our method introduces a more substantial structural innovation by integrating autoregressive (AR) and diffusion paradigms. This fusion design achieves higher inference efficiency while preserving generation quality.

C Derivation for Reinforcement Learning Fine-tuning

We frame the training objective within a reinforcement learning paradigm. We consider the denoising process as a Markov Decision Process (MDP). We define $\bar{t}$ to be the timestep of MDP, and the relation between MDP timestep $\bar{t}$ and denoising timestep t is $\bar{t} = T - t$. Let $s_{\bar{t}} \in \mathcal{S}$ to be the state at MDP timestep $\bar{t}$, where $\mathcal{S}$ is the state space. We define $s_{\bar{t}} = x_{T-\bar{t}} = x_t$, which is the noisy image x_t at timestep t. So when $\bar{t} = 0$, $t = T$, and we have $s_{\bar{t}=0} = x_T$, which is the initial noise. When $\bar{t} = T$, $t = 0$, and we have $s_{\bar{t}=T} = x_0$, which is the fully denoised image. $a_{\bar{t}} \in \mathcal{A}$ is the action at timestep $\bar{t}$, where $\mathcal{A}$ is the action space. We define $a_{\bar{t}}$ to be the noise ϵ_t added to noise image x_t. Thus we have $s_{\bar{t}+1} = x_{t-1}$, because of the claim $\bar{t} = T - t$.

The expected value of cumulative return along the entire trajectory $\tau = (x_T, x_{T-1}, \ldots, x_0)$ can be defined as:

$$\begin{aligned} E_{\tau\sim\rho}(R(\tau)) &= D_{KL}\left(p_{\text{fake}}(\tau) \parallel p_{\text{real}}(\tau)\right) \\ &= E_{x\sim p_{\text{fake}}} \log \frac{p_{\text{fake}}(x_{t=0})}{p_{\text{real}}(x_{t=0})} \\ &= E_{x\sim p_{\text{fake}}} \sum_{t=1}^{T} \log \frac{p_{\text{fake}}(x_{t-1}|x_t)}{p_{\text{real}}(x_{t-1}|x_t)} \\ &= \sum_{t=1}^{T} E_{x\sim p_{\text{fake}}} R(x_{t-1}, x_t) \end{aligned} \tag{11}$$

Thus, we define the $r(s_{\bar{t}}, a_{\bar{t}})$ as the log-ratio of transition probabilities under the fake and real distributions:

$$r(s_{\bar{t}}, a_{\bar{t}}) = R(x_{t-1}, x_t) = \log \frac{p_{\text{fake}}(x_{t-1}|x_t)}{p_{\text{real}}(x_{t-1}|x_t)} \tag{12}$$

This represents the log-ratio of the probabilities of the entire trajectory under the fake and real distributions.

The action-value function (Q-function) for taking action x_{t-1} in state x_t and following the policy thereafter is defined as:

$$\begin{aligned} Q(s_{\bar{t}}, a_{\bar{t}}) &= \mathbb{E}_{p_{\text{fake}}} \sum_{k=\bar{t}}^{T} r(s_{\bar{t}}, a_{\bar{t}}) \\ &= \mathbb{E}_{p_{\text{fake}}} \sum_{k=1}^{t} \log \frac{p_{\text{fake}}(x_{k-1}|x_k)}{p_{\text{real}}(x_{k-1}|x_k)} \end{aligned} \tag{13}$$

The defined Q-function satisfies the Bellman equation for the expected return:

$$\begin{aligned} Q(s_{\bar{t}}, a_{\bar{t}}) &= \mathbb{E}_{p_{\text{fake}}} \sum_{k=1}^{t} \log \frac{p_{\text{fake}}(x_{k-1}|x_k)}{p_{\text{real}}(x_{k-1}|x_k)} \\ &= \mathbb{E}_{p_{\text{fake}}} \left[\log \frac{p_{\text{fake}}(x_{t-1}|x_t)}{p_{\text{real}}(x_{t-1}|x_t)} + \mathbb{E}_{p_{\text{fake}}} \sum_{k=1}^{t-1} \log \frac{p_{\text{fake}}(x_{k-1}|x_k)}{p_{\text{real}}(x_{k-1}|x_k)} \right] \\ &= \mathbb{E}_{p_{\text{fake}}} \left[r(s_{\bar{t}}, a_{\bar{t}}) + Q(s_{\bar{t}+1}, a_{\bar{t}+1}) \right] \end{aligned} \tag{14}$$

Instead of parameterizing policy directly as $\pi(a_{\bar{t}}|s_{\bar{t}})$, we parameterize the policy with distillation model $x = G_\theta(\epsilon), \epsilon \sim \mathcal{N}(0; \mathbf{I})$.The training objective can now be expressed as:

$$\theta = \arg\max_{\theta} \mathcal{J}(\theta) \tag{15}$$

where

$$\mathcal{J}(\theta) = \mathbb{E}_{\tau\sim p_{\text{fake}}}[R(\tau)] = D_{KL}\left(p_{\text{fake}}(\tau) \parallel p_{\text{real}}(\tau)\right) \tag{16}$$

Follow DDPG Lillicrap et al. (2015), the gradient of this objective with respect to the generator parameters θ is:

$$\begin{aligned}\nabla_\theta \mathcal{J} &= -\mathbb{E}_{s\sim\rho^\mu}\left[\nabla_\theta \mu_\theta(s)\, \nabla_a Q^\mu(s,a)|_{a=\mu_\theta(s)}\right] \\ &= -\mathbb{E}_{x_t\sim\rho^\mu}\left[\nabla_\theta \mu_\theta(x_t) \cdot \nabla_{a_{\bar{t}}} x_{t-1} \cdot \nabla_{x_{t-1}} Q(x_t, a_{\bar{t}})\right]\end{aligned} \tag{17}$$

Since $a_{\bar{t}}$ is the noise and x_{t-1} is the next denoised image, so $\nabla_{a_{\bar{t}}} x_{t-1}$ is a constant, so we have:

$$\begin{aligned}\nabla_\theta \mathcal{J} &= C \cdot -\mathbb{E}_{x_t\sim\rho^\mu}\left[\nabla_\theta \mu_\theta(x_t) \cdot \nabla_{x_{t-1}} Q(x_t, a_{\bar{t}})\right] \\ &= C \cdot -\mathbb{E}_{x_t\sim\rho^\mu}\left[\nabla_\theta \mu_\theta(x_t) \cdot \nabla_{x_{t-1}} \mathbb{E}_{p_{fake}} \sum_{k=1}^{t} \log \frac{p_{\text{fake}}(x_{k-1}|x_k)}{p_{\text{real}}(x_{k-1}|x_k)}\right] \\ &= C \cdot -\mathbb{E}_{x_t\sim\rho^\mu}\left[\nabla_\theta \mu_\theta(x_t) \cdot \mathbb{E}_{p_{fake}} \nabla_{x_{t-1}} \log \frac{p_{\text{fake}}(x_{t-1}|x_t)}{p_{\text{real}}(x_{t-1}|x_t)}\right]\end{aligned} \tag{18}$$

D More Details of the User Study

To comprehensively evaluate the quality of generated videos, we conduct a user study based on a 5-point Likert scale (1 = Strongly Disagree, 2 = Disagree, 3 = Neutral, 4 = Agree, 5 = Strongly Agree). Participants were presented with paired videos generated by different models and were asked to score each video independently across three major dimensions: **Text Alignment, Motion Quality, and Visual Quality**. Each dimension contains multiple carefully designed questions targeting distinct aspects of video generation. The full questionnaire is provided below.

① Text Alignment

This dimension assesses how well the generated video matches the semantic intent of the input text prompt. Participants evaluated the following aspects:

Object Class Accuracy

The objects in the video correctly match the categories described in the prompt.

Multiple Objects Handling

The video correctly represents and maintains multiple objects without confusion or merging.

Color Accuracy

The colors in the video are accurate, stable, and consistent with the expected appearance.

Spatial Relationship

The spatial relationships between objects (e.g., relative positions, sizes) are logical and consistent.

Scene Coherence

The video presents a coherent and believable scene that aligns with the prompt.

Appearance Style Consistency

The appearance style (e.g., realistic, cartoon, cinematic) matches the intended style and remains stable.

② Motion Quality

This dimension focuses on the realism, stability, and smoothness of motion across the video sequence:

Temporal Flickering

The video shows minimal flickering or frame-to-frame jitter.

Motion Smoothness

Movements in the video appear smooth and natural without abrupt jumps.

Dynamic Degree

The video presents an appropriate level of dynamics that matches the scene and prompt.

Temporal Style Consistency

The stylistic elements of the video remain consistent across time.

Overall Consistency

The video maintains overall coherence and consistency across all frames.

③ Visual Quality

This dimension evaluates perceptual clarity, aesthetics, and frame-level consistency:

Subject Consistency

The main subject remains visually consistent throughout the video.

Background Consistency

The background stays stable and does not exhibit unexpected changes across frames.

Aesthetic Quality

The overall artistic and aesthetic quality of the video is appealing.

Imaging Quality

The video appears clear, sharp, and free from noticeable visual artifacts.

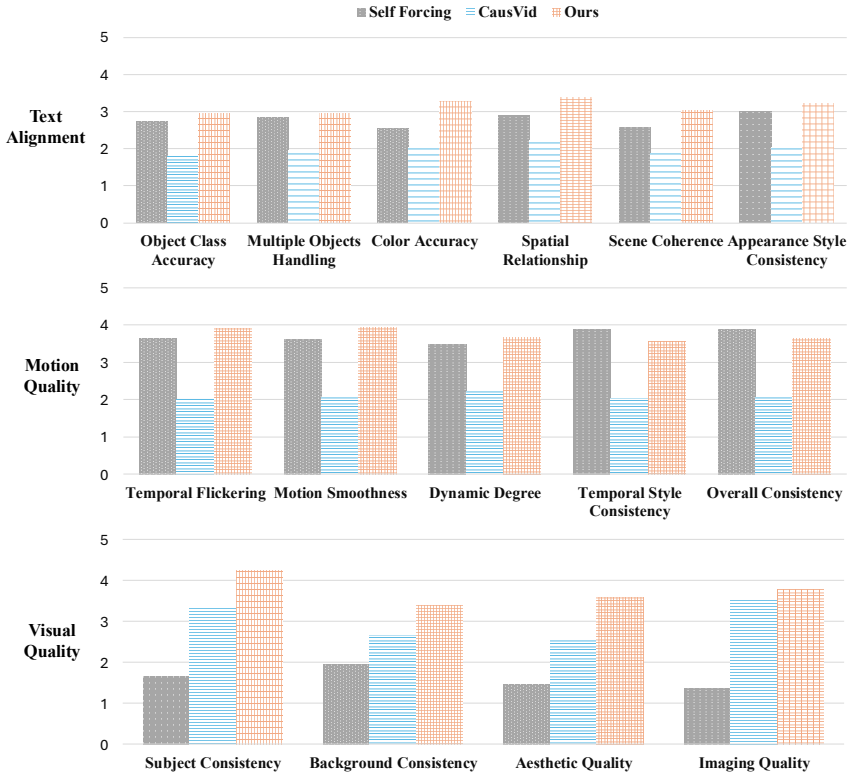


Figure 10: Detail scores of each dimension in the user study.