

# PUSA V1.0: SURPASSING WAN-I2V WITH \$500 TRAINING COST BY USING VECTORIZED Timestep ADAPTATION

Yaofang Liu<sup>1,7†\*</sup> Yumeng Ren<sup>1,7</sup> Aitor Artola<sup>1,7</sup> Yuxuan Hu<sup>2,3</sup> Xiaodong Cun<sup>4</sup>  
 Xiaotong Zhao<sup>5</sup> Alan Zhao<sup>5</sup> Raymond H. Chan<sup>6,7</sup> Suiyun Zhang<sup>3</sup>  
 Rui Liu<sup>3†</sup> Dandan Tu<sup>3†</sup> Jean-Michel Morel<sup>1†</sup>

<sup>1</sup>City University of Hong Kong <sup>2</sup>The Chinese University of Hong Kong <sup>3</sup>Huawei Research

<sup>4</sup>Great Bay University <sup>5</sup>AI Technology Center, Tencent PCG <sup>6</sup>Lingnan University

<sup>7</sup>Hong Kong Centre for Cerebro-Cardiovascular Health Engineering

†Corresponding authors

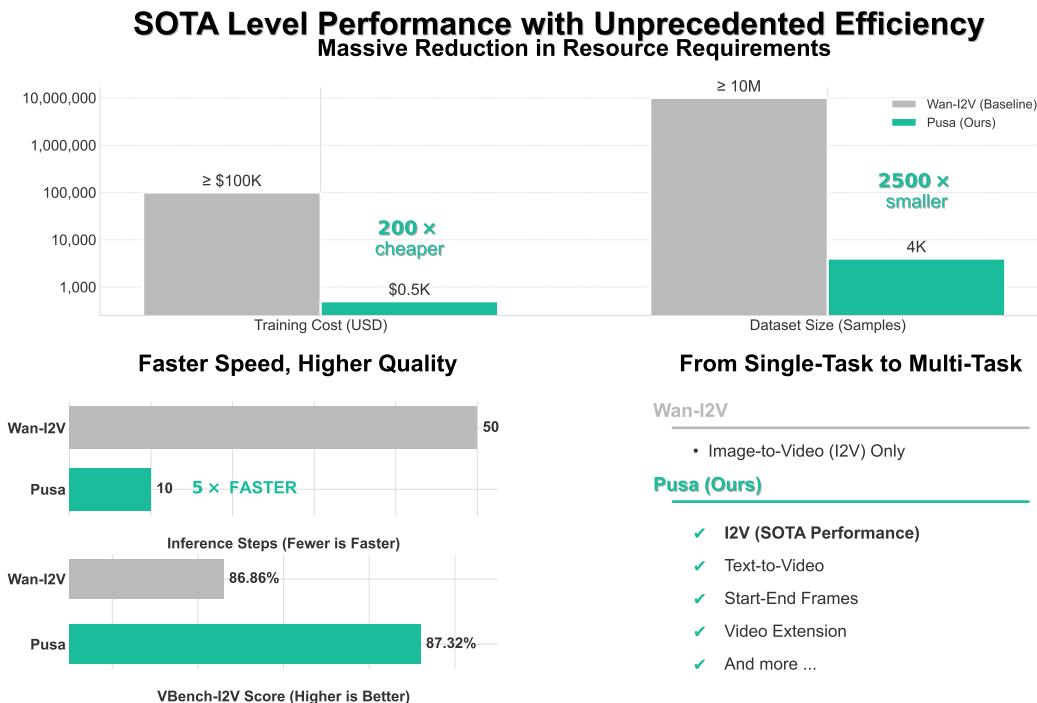


Figure 1: Overview of Pusa’s performance and efficiency. Specifically, Pusa outperforms Wan-I2V on Vbench-I2V with only  $\leq 1/2500$  dataset,  $\leq 1/200$  training budget, and  $1/5$  inference steps. Besides, Wan-I2V can only do image-to-video generation, while the same Pusa model has many other capabilities including: start-end frames, video extension, text-to-video, and so on.

## ABSTRACT

The rapid advancement of video diffusion models has been hindered by fundamental limitations in temporal modeling, particularly the rigid synchronization of frame evolution imposed by conventional scalar timestep variables. While task-specific adaptations and autoregressive models have sought to address these challenges, they remain constrained by computational inefficiency, catastrophic forgetting, or narrow applicability. In this work, we present **Pusa**<sup>1</sup>, a ground-

\*Work partially done during an internship at Huawei Research.

<sup>1</sup>Pusa (/pu: 'sa:/) normally refs to "Thousand-Hand Guanyin" in Chinese, reflecting the iconography of many hands to symbolize her boundless compassion and ability. We use this name to indicate that our model

breaking paradigm that leverages **vectorized timestep adaptation (VTA)** to enable fine-grained temporal control within a unified video diffusion framework. Besides, VTA is a non-destructive adaptation, which means it fully preserves the capabilities of the base model. By finetuning the SOTA Wan2.1-T2V-14B model with VTA, we achieve unprecedented efficiency—**surpassing the performance of Wan-I2V-14B with  $\leq 1/200$  of the training cost (\$500 vs.  $\geq \$100,000$ ) and  $\leq 1/2500$  of the dataset size (4K vs.  $\geq 10M$  samples)**. Pusa not only sets a new standard for image-to-video (I2V) generation, achieving a VBench-I2V total score of 87.32% (vs. 86.86% of Wan-I2V-14B), but also **unlocks many zero-shot multi-task capabilities such as start-end frames and video extension**—all without task-specific training. Meanwhile, **Pusa can still perform text-to-video generation**. Mechanistic analyses reveal that our approach preserves the foundation model’s generative priors while surgically injecting temporal dynamics, avoiding the combinatorial explosion inherent to vectorized timesteps. This work establishes a scalable, efficient, and versatile paradigm for next-generation video synthesis, democratizing high-fidelity video generation for research and industry alike. Code will be open-sourced at <https://github.com/Yaofang-Liu/Pusa-VidGen>

## 1 INTRODUCTION

The advent of diffusion models (Song et al., 2020; Ho et al., 2020) has heralded a paradigm shift in generative modeling, particularly in the domain of image synthesis. These models, which leverage an iterative noise reduction process, have demonstrated remarkable efficacy in producing high-fidelity samples. Naturally, extending this framework to video generation (Ho et al., 2022; He et al., 2022; Chen et al., 2023; Wang et al., 2023; Ma et al., 2024; OpenAI, 2024; Xing et al., 2023b; Liu et al., 2024a) has been a focal point, yet it exposes fundamental limitations in modeling complex temporal dynamics. Conventional video diffusion models (VDMs) typically employ scalar timestep variables, enforcing uniform temporal evolution across all frames. This approach, while effective for text-to-video (T2V) clips generation, struggles with nuanced temporal dependencies task like image-to-video (I2V) generation, as highlighted in the FVDM work (Liu et al., 2024b) which solved this problem by introducing a vectorized timestep approach to enable independent frame evolution.

Concurrently, methods like Diffusion Forcing (Chen et al., 2024) and AR-Diffusion (Sun et al., 2025) also explored autoregressive paradigms to avoid this rigid synchronization modeling form of conventional VDMs. Nonetheless, their applications w.r.t. video generation remained constrained by the autoregressive designs with token/frame-level noise. Large-scale models such as MAGI-1 (Teng et al., 2025) and SkyReels V2 Chen et al. (2025) advanced scalability but still faced challenges in balancing computational efficiency and multi-task capability.

In this work, we bridge the gap between theoretical innovation and industrial deployment by extending the FVDM framework to industrial scale through fine-tuning on the SOTA open-source T2V model Wan2.1-T2V-14B (Wan-T2V) Wan et al. (2025) with vectorized timestep adaptation (VTA). Our key insight is to leverage the vectorized timestep variable (VTV) designs of FVDM within a robust large-model ecosystem, enabling efficient adaptation to diverse video generation tasks, especially I2V, while drastically reducing computational and data requirements.

As demonstrated in Table 1, our approach (Pusa) achieves a landmark improvement: with *thousands of times smaller datasets* (4K samples vs.  $\geq 10M$ ) and *hundreds of times less computation* (training cost reduced to  $0.5K$  vs.  $\geq 100K$ ), we surpass prior art Wan2.1-I2V-14B (Wan-I2V) Wan et al. (2025) on the VBench-I2V benchmark Huang et al. (2024). Notably, Pusa not only achieves a higher total score (87.32% vs. 86.86%) with superior I2V quality (94.84% vs. 92.90%), but also extends capabilities beyond T2V and I2V generation to support start-end frame generation, video extension, and other complex temporal tasks—all within a single unified model. This marks a critical departure from previous models like Wan-I2V, which are limited to I2V and require prohibitive resources.

---

uses many timestep variables to achieve numerous video generation capabilities, and we will fully open source it to let the community benefit from this tech.

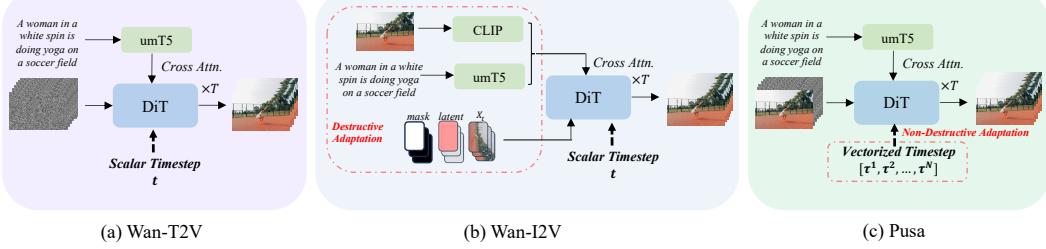


Figure 2: Paradigm comparison among (c) Pusa and (b) Wan-I2V, both support image-to-video (I2V) generation, and are finetuned from a text-to-video model (a) Wan-T2V. As illustrated by the figure (b), Wan-I2V modifies the model with an additional mask mechanism and adds a clip embedding of the condition image to enable I2V capability. However, this is a destructive adaptation of the original model that changes the model’s input and internal calculation process, which indicates it cannot fully utilize the pretrained priors of the base model. In contrast, our proposed model, Pusa, we inflate the model’s timestep variable from a scalar to a vector, which is a non-destructive adaptation. With this method, Pusa can fully utilize the pretrained priors and use much less resources to learn temporal dynamics. Regarding the I2V task, Pusa achieves unprecedented efficiency, surpassing Wan-I2V with  $\leq 1/2500$  training data, revolutionizing the video diffusion paradigm.

The core of our innovation lies in the synergistic combination of FVDM’s temporal modeling prowess with the SOTA generative capacity of Wan-T2V, optimized through a lightweight fine-tuning strategy. By preserving the vectorized timestep formulation, each frame evolves along its independent temporal trajectory during diffusion, enabling the model to capture intricate inter-frame dependencies without global synchronization. This architecture not only enhances temporal coherence but also enables zero-shot generalization to new tasks, as validated by our results in I2V generation without task-specific training.

Our contributions can be summarized as:

- Industrial-Scale Efficiency: We demonstrate the first large-model adaptation of FVDM, achieving unprecedented data efficiency ( $\leq 1/2500$  dataset size) and computational efficiency ( $\leq 1/200$  training cost) compared to Wan-I2V, revolutionizing the video diffusion paradigm.
- Multi-Task Generalization: our proposed model supports not only T2V and I2V, but also start-end frames, video extension, and more without additional training.
- Quality-Throughput Tradeoff: Despite significantly reduced resources, Pusa achieves a superior total score (87.32% vs. 86.86%), proving the FVDM paradigm works well for large foundational models and greatly exceeds previous methods.

This work represents a pivotal step toward democratizing advanced video generation: by unlocking the full potential of FVDM paradigm within a practical, scalable framework, we enable high-fidelity, multi-task video synthesis accessible to researchers and industries alike. Through rigorous benchmarking and novel fine-tuning strategies, we establish that temporal modeling innovation, when paired with strategic large-model adaptation, can overcome the long-standing tradeoff between performance, efficiency, and versatility in video diffusion.

## 2 METHODOLOGY

This section details the mathematical framework for our proposed Pusa model. We begin by reviewing the fundamentals of flow matching for generative modeling. Subsequently, we introduce the concept of a vectorized timestep variable from FVDM. Finally, we integrate these concepts to formulate the Pusa’s objective and describe the video generation process.

## 2.1 PRELIMINARIES: FLOW MATCHING FOR GENERATIVE MODELING

Generative modeling aims to learn a model capable of synthesizing samples from a target data distribution  $q_0(\mathbf{z})$  over  $\mathbb{R}^D$ . Continuous Normalizing Flows (CNFs) (Chen, 2018) achieve this by transforming samples  $\mathbf{z}_1$  from a simple base distribution  $q_1(\mathbf{z})$  (e.g., a standard Gaussian  $\mathcal{N}(\mathbf{0}, \mathbf{I})$ ) to samples  $\mathbf{z}_0$  that approximate the target distribution  $q_0(\mathbf{z})$ . This transformation is defined by an invertible mapping, often conceptualized as an ordinary differential equation (ODE) trajectory. Specifically, a probability path  $\{\mathbf{z}_t\}_{t \in [0,1]}$  is defined, connecting  $\mathbf{z}_0 \sim q_0$  at  $t = 0$  to  $\mathbf{z}_1 \sim q_1$  at  $t = 1$ . The dynamics along this path are described by an ODE:

$$\frac{d\mathbf{z}_t}{dt} = v_t(\mathbf{z}_t, t), \quad t \in [0, 1] \quad (1)$$

where  $v_t : \mathbb{R}^D \times [0, 1] \rightarrow \mathbb{R}^D$  is a time-dependent vector field.

Flow Matching (FM) (Lipman et al., 2022; Liu et al., 2022; Tong et al., 2023) is a simulation-free technique to directly learn this vector field  $v_t(\mathbf{z}_t, t)$  by training a neural network  $v_\theta(\mathbf{z}_t, t)$  to approximate it. This is achieved by regressing  $v_\theta(\mathbf{z}_t, t)$  against a target vector field  $u_t(\mathbf{z}_t | \mathbf{z}_0, \mathbf{z}_1)$ . This target field is defined along specified probability paths  $p_t(\mathbf{z}_t | \mathbf{z}_0, \mathbf{z}_1)$  that connect samples  $\mathbf{z}_0 \sim q_0$  to corresponding samples  $\mathbf{z}_1 \sim q_1$ .

A common choice for these paths is a linear interpolation between a data sample  $\mathbf{z}_0$  and a prior sample  $\mathbf{z}_1$ :

$$\mathbf{z}_t = (1 - t)\mathbf{z}_0 + t\mathbf{z}_1, \quad t \in [0, 1] \quad (2)$$

For such paths, the conditional target vector field is the time derivative of  $\mathbf{z}_t$ :

$$u_t(\mathbf{z}_0, \mathbf{z}_1) = \frac{d\mathbf{z}_t}{dt} = \mathbf{z}_1 - \mathbf{z}_0 \quad (3)$$

Note that for linear interpolation paths,  $u_t$  is independent of  $t$  and  $\mathbf{z}_t$ , depending only on the endpoints  $\mathbf{z}_0$  and  $\mathbf{z}_1$ . The (conditional) flow matching objective function to train the neural network  $v_\theta$  is then:

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{t \sim \mathcal{U}[0,1], \mathbf{z}_0 \sim q_0, \mathbf{z}_1 \sim q_1} \left[ \|v_\theta((1-t)\mathbf{z}_0 + t\mathbf{z}_1, t) - (\mathbf{z}_1 - \mathbf{z}_0)\|_2^2 \right] \quad (4)$$

where  $\mathcal{U}[0, 1]$  is the uniform distribution over  $[0, 1]$  and  $\|\cdot\|_2^2$  denotes the squared Euclidean norm. Once  $v_\theta$  is trained, new samples approximating  $q_0$  can be generated by first sampling  $\mathbf{z}_1 \sim q_1$  and then solving the ODE  $\frac{d\mathbf{z}_t}{dt} = v_\theta(\mathbf{z}_t, t)$  from  $t = 1$  down to  $t = 0$ . The resulting  $\mathbf{z}_0$  is a generated sample.

## 2.2 FRAME-AWARE FLOW MATCHING

We now extend the flow matching framework to video generation with the vectorized timestep proposed by FVDM Liu et al. (2024b), allowing for nuanced temporal modeling in video generation, which differs from conventional video diffusion or flow matching models.

A video clip  $\mathbf{X}$  is represented as a sequence of  $N$  frames. Each frame  $\mathbf{x}^i \in \mathbb{R}^d$  is a  $d$ -dimensional column vector. The entire video clip can be represented as an  $N \times d$  matrix  $\mathbf{X}$ , where the  $i$ -th row is  $\mathbf{x}^{i\top}$ . This can be written as  $\mathbf{X} = [\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^N]^\top$ , thus  $\mathbf{X} \in \mathbb{R}^{N \times d}$ .

In contrast to the single scalar time variable  $t$  used in standard flow matching (Eq. 4), we introduce a **vectorized timestep variable**  $\tau \in [0, 1]^N$ , defined as:

$$\tau = [\tau^1, \tau^2, \dots, \tau^N]^\top \quad (5)$$

Here, each component  $\tau^i \in [0, 1]$  represents the individual progression parameter of the  $i$ -th frame along its respective probability path from the data distribution to a prior distribution. This vectorization allows each frame to evolve at a potentially different rate or stage within the generative process.

Let  $\mathbf{X}_0 = [\mathbf{x}_0^1, \dots, \mathbf{x}_0^N]^\top$  be a video sampled from the true data distribution  $q_{\text{data}}(\mathbf{X})$ , where  $\mathbf{x}_0^i \in \mathbb{R}^d$ . Similarly, let  $\mathbf{X}_1 = [\mathbf{x}_1^1, \dots, \mathbf{x}_1^N]^\top$  be a video sampled from a simple prior distribution  $q_{\text{prior}}(\mathbf{X})$  (e.g., each frame  $\mathbf{x}_1^i$  is independently drawn from  $\mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_d)$ ).

For each frame  $i$ , we define a conditional probability path  $p(\mathbf{x}_{\tau^i}^i | \mathbf{x}_0^i, \mathbf{x}_1^i)$  indexed by its individual timestep  $\tau^i$ . Adopting the linear interpolation strategy from Eq. 2 for each frame (which are  $d$ -dimensional vectors):

$$\mathbf{x}_{\tau^i}^i = (1 - \tau^i)\mathbf{x}_0^i + \tau^i\mathbf{x}_1^i \quad (6)$$

The state of the entire video, corresponding to a specific vectorized timestep  $\boldsymbol{\tau}$ , is then given by the  $N \times d$  matrix  $\mathbf{X}_{\boldsymbol{\tau}}$ , whose  $i$ -th row is  $\mathbf{x}_{\tau^i}^{i\top}$ :

$$\mathbf{X}_{\boldsymbol{\tau}} = [\mathbf{x}_{\tau^1}^1, \mathbf{x}_{\tau^2}^2, \dots, \mathbf{x}_{\tau^N}^N]^\top \quad (7)$$

We aim to learn a single neural network  $v_{\theta}(\mathbf{X}, \boldsymbol{\tau})$  that models the joint dynamics of all frames conditioned on their respective timesteps. This network takes the current video state  $\mathbf{X} \in \mathbb{R}^{N \times d}$  (which is  $\mathbf{X}_{\boldsymbol{\tau}}$  during training) and the vectorized timestep  $\boldsymbol{\tau} \in [0, 1]^N$  as input. It outputs a velocity field for the entire video, an  $N \times d$  matrix denoted as  $v_{\theta}(\mathbf{X}, \boldsymbol{\tau}) = [\mathbf{v}^1, \dots, \mathbf{v}^N]^\top$ , where each  $\mathbf{v}^i \in \mathbb{R}^d$  is the velocity vector for the  $i$ -th frame. Thus,  $v_{\theta} : \mathbb{R}^{N \times d} \times [0, 1]^N \rightarrow \mathbb{R}^{N \times d}$ .

The target vector field for the entire video  $\mathbf{X}_{\boldsymbol{\tau}}$ , conditioned on the initial video  $\mathbf{X}_0$  and target prior  $\mathbf{X}_1$ , is an  $N \times d$  matrix  $\mathbf{U}(\mathbf{X}_0, \mathbf{X}_1)$ . Its  $i$ -th row is the transpose of the derivative of the  $i$ -th frame's path (Eq. 6) with respect to its individual timestep  $\tau^i$ . Using the derivative from Eq. 3 for each frame:

$$\frac{d\mathbf{x}_{\tau^i}^i}{d\tau^i} = \mathbf{x}_1^i - \mathbf{x}_0^i \quad (8)$$

Thus, the target video-level vector field is:

$$\mathbf{U}(\mathbf{X}_0, \mathbf{X}_1) = [(\mathbf{x}_1^1 - \mathbf{x}_0^1), \dots, (\mathbf{x}_1^N - \mathbf{x}_0^N)]^\top = \mathbf{X}_1 - \mathbf{X}_0 \quad (9)$$

Notably, for the linear interpolation path, this target vector field  $\mathbf{X}_1 - \mathbf{X}_0$  is independent of both the current video state  $\mathbf{X}_{\boldsymbol{\tau}}$  and the vectorized timestep  $\boldsymbol{\tau}$  itself, simplifying the regression target. The video state  $\mathbf{X}_{\boldsymbol{\tau}}$  at timestep  $\boldsymbol{\tau}$  is constructed via frame-wise linear interpolation:

$$\mathbf{X}_{\boldsymbol{\tau}} = (1 - \boldsymbol{\tau}) \odot \mathbf{X}_0 + \boldsymbol{\tau} \odot \mathbf{X}_1 \quad (10)$$

where  $\odot$  denotes element-wise multiplication between the timestep vector  $\boldsymbol{\tau} = [\tau^1, \tau^2, \dots, \tau^N]^\top$  and each frame.

**Key Properties:** 1. *Path Consistency*: Each frame evolves linearly:  $\mathbf{x}_{\tau^i}^i = (1 - \tau^i)\mathbf{x}_0^i + \tau^i\mathbf{x}_1^i$ . 2. *Vector Field Simplicity*:  $\frac{d\mathbf{X}_{\boldsymbol{\tau}}}{d\boldsymbol{\tau}} = \mathbf{X}_1 - \mathbf{X}_0$  (constant for all  $\boldsymbol{\tau}$ ). 3. *Decoupling*: Frame dynamics depend only on their own  $\tau^i$ , enabling asynchronous evolution.

The parameters  $\theta$  of the neural network  $v_{\theta}$  are optimized by minimizing the Frame-Aware Flow Matching (FAFM) objective function:

$$\mathcal{L}_{\text{FAFM}}(\theta) = \mathbb{E}_{\mathbf{X}_0 \sim q_{\text{data}}, \mathbf{X}_1 \sim q_{\text{prior}}, \boldsymbol{\tau} \sim p_{\text{PTSS}}(\boldsymbol{\tau})} \left[ \|v_{\theta}(\mathbf{X}_{\boldsymbol{\tau}}, \boldsymbol{\tau}) - (\mathbf{X}_1 - \mathbf{X}_0)\|_F^2 \right] \quad (11)$$

where:

- $\mathbf{X}_{\boldsymbol{\tau}}$  is the video state constructed according to Eq. 7 using the sampled  $\mathbf{X}_0, \mathbf{X}_1$ , and  $\boldsymbol{\tau}$ .
- $\|\cdot\|_F^2$  denotes the squared Frobenius norm. If  $v_{\theta}(\mathbf{X}_{\boldsymbol{\tau}}, \boldsymbol{\tau}) = [\mathbf{v}^1, \dots, \mathbf{v}^N]^\top$ , then the squared Frobenius norm is equivalent to  $\sum_{i=1}^N \|\mathbf{v}^i - (\mathbf{x}_1^i - \mathbf{x}_0^i)\|_2^2$ .
- $\boldsymbol{\tau} \sim p_{\text{PTSS}}(\boldsymbol{\tau})$  indicates that the vectorized timestep  $\boldsymbol{\tau}$  is sampled according to a **Probabilistic Timestep Sampling Strategy (PTSS)**. This strategy is designed to expose the model to both synchronized and desynchronized frame evolutions during training. The PTSS is defined as follows: With a probability  $p_{\text{async}} \in [0, 1]$ :

- Each component  $\tau^i$  of  $\boldsymbol{\tau}$  is sampled independently from  $\mathcal{U}[0, 1]$ :  $\tau^i \stackrel{\text{i.i.d.}}{\sim} \mathcal{U}[0, 1]$  for  $i = 1, \dots, N$ . This represents asynchronous evolution.

With probability  $1 - p_{\text{async}}$ :

- A single timestep  $\tau_{\text{sync}}$  is sampled from  $\mathcal{U}[0, 1]$ .
- All components of  $\boldsymbol{\tau}$  are set to this value:  $\tau^i = \tau_{\text{sync}}$  for  $i = 1, \dots, N$ . This represents synchronous evolution, where all frames share the same progression parameter.

### 2.3 PUSA VIA VECTORIZED Timestep Adaptation

Our model, named **Pusa**, adapts a large-scale, pre-trained T2V diffusion transformer to the vectorized timestep introduced by FVDM Liu et al. (2024b). The adaptation, achieved via a lightweight fine-tuning process we term Vectorized Timestep Adaptation (VTA), imbues the model with fine-grained temporal control, enabling advanced capabilities such as zero-shot Image-to-Video (I2V) generation.

#### 2.3.1 IMPLEMENTATION OF VECTORIZED Timestep ADAPTATION

The foundational principle of our implementation is to re-engineer the core architecture to process a vectorized timestep  $\tau$  instead of a scalar timestep  $t$ .

The architectural adaptation is primarily focused on the model’s temporal conditioning mechanism. We introduce two key modifications:

**Vectorized Timestep Embedding:** The timestep embedding module is modified to process the input vector  $\tau$ , generating a sequence of frame-specific embeddings  $\mathbf{E}_\tau \in \mathbb{R}^{N_1 \times D}$ , where  $N_1$  is the number of frames in the video latent sequence, each vector in the sequence corresponds to a latent frame’s individual.

**Per-Frame Modulation:** These frame-specific embeddings are subsequently projected to produce per-frame modulation parameters (i.e., scale, shift, and gate) within each block of the DiT architecture. The operation of a DiT Peebles & Xie (2023) block on the latent representation  $\mathbf{z}^i$  of the  $i$ -th frame is thus conditioned on its individual timestep  $\tau^i$ , which can be conceptually expressed as:

$$\mathbf{z}_{\text{out}}^i = \text{DiTBlock}(\mathbf{z}_{\text{in}}^i, \text{context}, \text{modulate}(\tau^i))$$

This design allows the model to process a batch of frames that exist at different points along their respective generative paths simultaneously, forming the basis for fine-grained temporal control. Most importantly, this modification is non-destructive, which means it fully preserves the T2V capability of the base model. The adapted model generates same samples by setting all frame timesteps to same values as the base model.

#### 2.3.2 TRAINING PROCEDURE

The optimization follows the FAFM objective defined in Eq. 11. A key advantage of our approach is its simplicity: by leveraging the robust generative prior of the base model, we circumvent the need for sampling synchronous timesteps. Instead, we train the model directly with a fully randomized vectorized timestep ( $p_{\text{async}} = 1$ ), where each component  $\tau^i$  is sampled independently from  $U[0, 1]$ . This stochastic training regimen compels the model to learn fine-grained temporal control from a maximally diverse distribution of temporal states.

#### 2.3.3 INFERENCE FOR IMAGE-TO-VIDEO GENERATION

Pusa performs zero-shot I2V generation by strategically manipulating the vectorized timestep  $\tau$  during sampling. To condition generation on a starting image  $\mathbf{I}_0$ , we clamp its timestep component to zero throughout inference (i.e.,  $\tau_s^0 = 0$  for all steps  $s$ ). During sampling, which follows the Euler method for ODE integration, this ensures the change in the first frame’s latent is always zero, effectively fixing it as a clean condition. This flexible control scheme naturally extends to other complex temporal tasks, such as start-end frames and video extension. The detailed I2V sampling procedure is outlined in Algorithm 1.

## 3 EXPERIMENTS

Our experiments are designed to rigorously validate the three core contributions of this work: (1) the unprecedented efficiency of adapting large foundation models via VTA; (2) the superior performance of our model, Pusa, on the primary task of I2V generation; and (3) the emergent zero-shot multi-task capabilities of Pusa.

**Algorithm 1** Pusa: Sampling for Image-to-Video Generation

---

**Require:** Trained model  $v_\theta$ , VAE Encoder  $E$  and Decoder  $D$ , Scheduler.

**Require:** Initial image  $\mathbf{I}_0$ , text prompt  $c$ , number of frames  $N_1$ , inference steps  $S$ .

- 1: Encode prompt:  $\mathbf{c}_{emb} \leftarrow \text{EncodePrompt}(c)$ .
- 2: Encode image to initial latent:  $\mathbf{z}_0^0 \leftarrow E(\mathbf{I}_0)$ .
- 3: Sample noise for remaining frames:  $[\mathbf{z}_1^1, \dots, \mathbf{z}_1^{N_1-1}] \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ .
- 4:  $\triangleright$  Initialize video with clean first frame and noisy subsequent frames.
- 5: Construct initial latent video:  $\mathbf{Z}_1 \leftarrow [\mathbf{z}_0^0, \mathbf{z}_1^1, \dots, \mathbf{z}_1^{N_1-1}]$ .
- 6: Retrieve scheduler noise levels  $\{\sigma_s\}_{s=1}^S$ , where  $\sigma_1 > \dots > \sigma_S \approx 0$ .
- 7: **for**  $s \leftarrow 1, \dots, S-1$  **do**
- 8:   Let  $\sigma_{current} \leftarrow \sigma_s$  and  $\sigma_{next} \leftarrow \sigma_{s+1}$ .  
 $\triangleright$  Construct vectorized timestep to freeze the first frame.
- 9:   Set current path parameter  $\tau_s \leftarrow \sigma^{-1}([0, \sigma_{current}, \dots, \sigma_{current}]^\top)$ .
- 10:   Set current noise levels  $\boldsymbol{\sigma}_s \leftarrow [0, \sigma_{current}, \dots, \sigma_{current}]^\top$ .
- 11:   Set next noise levels  $\boldsymbol{\sigma}_{s+1} \leftarrow [0, \sigma_{next}, \dots, \sigma_{next}]^\top$ .
- 12:   Predict vector field:  $\hat{\mathbf{U}}_s \leftarrow v_\theta(\mathbf{Z}_s, \tau_s, \mathbf{c}_{emb})$ .
- 13:    $\triangleright$  Update latents; first frame remains unchanged.
- 14:    $\mathbf{Z}_{s+1} \leftarrow \mathbf{Z}_s + \hat{\mathbf{U}}_s \odot (\boldsymbol{\sigma}_{s+1} - \boldsymbol{\sigma}_s)$ .
- 15: **end for**
- 16: Decode final latent video:  $\mathbf{X}_{out} \leftarrow D(\mathbf{Z}_S)$ .
- 18: **return** Output video  $\mathbf{X}_{out}$ .

---

**Table 1: Comprehensive Vbench-I2V Model Performance Leaderboard.** This table presents a comparison of leading Image-to-Video (I2V) models, evaluated on a suite of automated metrics. Models are ranked by the overall Total Score. Our model demonstrates state-of-the-art performance, achieving a top-tier rank among open-source models and notably surpassing its architectural baseline, Wan-I2V, especially in total score and dynamic motion generation. All scores are reported in percentages (%). Higher is better for all metrics. Best score in each column is in **bold**. **Abbreviations:** **SC**: Subject Consistency, **BC**: Background Consistency, **MS**: Motion Smoothness, **DD**: Dynamic Degree, **AQ**: Aesthetic Quality, **IQ**: Imaging Quality, **I2V-S**: I2V Subject Consistency, **I2V-B**: I2V Background Consistency, **CM**: Camera Motion.

Model	Overall Scores $\uparrow$			Quality Metrics $\uparrow$						I2V Metrics $\uparrow$		
	Total	I2V	Quality	SC	BC	MS	DD	AQ	IQ	I2V-S	I2V-B	CM
<b>— Closed / Proprietary Models —</b>												
Gen-4-I2V (API)	88.27	95.65	80.89	93.23	96.79	98.99	55.20	61.77	70.41	97.84	97.46	<b>68.26</b>
STIV (Apple)	86.73	93.48	79.98	98.40	98.39	<b>99.61</b>	15.28	66.00	70.81	<b>98.96</b>	97.35	11.17
<b>— Open Source Models —</b>												
Magi-1	<b>89.28</b>	<b>96.12</b>	<b>82.44</b>	93.96	96.74	98.68	68.21	64.74	69.71	98.39	<b>99.00</b>	50.85
Step-Video-TI2V	88.36	95.50	81.22	96.02	97.06	99.24	48.78	62.29	70.44	97.86	98.63	49.23
DynamiCrafter-512	86.99	93.53	80.46	93.81	96.64	96.84	<b>69.67</b>	60.88	68.60	97.21	97.40	31.98
CogVideoX-5b-I2V	86.70	94.79	78.61	94.34	96.42	98.40	33.17	61.87	70.01	97.19	96.74	67.68
Animate-Anything	86.48	94.25	78.71	<b>98.90</b>	98.19	98.61	2.68	<b>67.12</b>	<b>72.09</b>	98.76	98.58	13.08
SEINE-512x512	85.52	92.67	78.37	95.28	97.12	97.12	27.07	64.55	71.39	97.15	96.94	20.97
I2VGen-XL	85.28	92.11	78.44	94.18	97.09	98.34	26.10	64.82	69.14	96.48	96.83	18.48
ConsistI2V	84.07	91.91	76.22	95.27	98.28	97.38	18.62	59.00	66.92	95.82	95.95	33.92
VideoCrafter	82.57	86.31	78.84	97.86	<b>98.79</b>	98.00	22.60	60.78	71.68	91.17	91.31	33.60
CogVideoX1.5-5B	71.58	92.25	50.90	91.80	94.66	40.98	62.29	70.21	97.07	96.46	95.50	39.71
SVD-XT-1.1	–	–	79.40	95.42	96.77	98.12	43.17	60.23	70.23	97.51	97.62	–
SVD-XT-1.0	–	–	80.11	95.52	96.61	98.09	52.36	60.15	69.80	97.52	97.63	–
<b>Wan-I2V</b>	86.86	92.90	<b>80.82</b>	<b>94.86</b>	<b>97.07</b>	97.90	51.38	<b>64.75</b>	<b>70.44</b>	96.95	96.44	<b>34.76</b>
<b>Ours</b>	<b>87.32</b>	<b>94.84</b>	79.80	92.27	96.02	<b>98.49</b>	<b>52.60</b>	63.15	68.27	<b>97.64</b>	<b>99.24</b>	29.46

## 3.1 SETUP

Towards efficient adaptation with fewer resources, we perform lightweight fine-tuning on the SOTA open-source Wan-T2V model using the LoRA (Low-Rank Adaptation) technique Hu et al. (2022), which enables parameter-efficient training. The training infrastructure consists of 8 GPUs, each has

80GB memory and high memory bandwidth, with DeepSpeed Zero2 Rajbhandari et al. (2020) for memory optimization, achieving a total batch size of 8. This is much more efficient compared to full fine-tuning, which requires at least  $4 \times 8$  GPUs with DeepSpeed Zero3 and is prohibitively expensive for most researchers. Note that our method should also work for full finetuning as already implemented in Pusa V0.5 Liu & Liu (2025). Overall, this configuration reduces the training cost of our final model to  $0.5K$ , which is at least  $200\times$  more efficient than the Wan-I2V baseline ( $\geq 100K$ ). The LoRA implementation is based on DiffSynth-Studio<sup>2</sup>, leveraging its optimized diffusion model training pipeline.

Regarding the finetuning dataset, we utilize the samples from Vbench2.0 Zheng et al. (2025), which contains 3,860 high-quality caption-video pairs generated by Wan-T2V. This dataset spans diverse visual domains and temporal structures (e.g., natural scenes, human activities, camera motion), ensuring robust model generalization and aligned with Wan’s original distribution. Notably, the dataset size is less than 4K samples, representing a  $\geq 2,500$  reduction compared to the  $\geq 10M$  samples required by Wan-I2V, demonstrating our data efficiency.

**Table 2: Comprehensive studies on key hyperparameters.** This table presents a detailed analysis of our model’s performance by varying training iterations, LoRA configurations, and the number of inference steps. All scores are reported in percentages (%), with higher values indicating better performance. For each ablation group, the best score per metric is highlighted in **bold**.

Group	Setting	Overall			Quality Metrics					I2V Metrics			
		Total	Quality	I2V	SC	BC	MS	DD	AQ	IQ	I2V-S	I2V-B	CM
<b>(a) LoRA Configurations</b>													
256	$\alpha = 1.0$	79.75	73.25	86.25	84.35	88.20	98.91	28.92	60.90	64.73	91.69	91.18	<b>28.00</b>
	$\alpha = 1.4$	83.12	76.07	90.17	85.65	96.67	98.99	<b>30.06</b>	60.13	67.15	90.33	<b>98.78</b>	23.48
	$\alpha = 1.7$	<b>85.86</b>	<b>77.96</b>	<b>93.76</b>	<b>97.18</b>	<b>96.68</b>	<b>99.06</b>	10.40	<b>63.04</b>	<b>70.70</b>	<b>98.60</b>	98.30	8.40
	$\alpha = 2.0$	84.22	76.30	92.14	96.71	94.63	99.05	6.40	60.34	69.63	97.96	95.77	16.00
512	$\alpha = 1.0$	80.17	74.35	85.99	83.16	86.99	<b>98.05</b>	62.40	58.44	62.49	91.23	90.70	34.40
	$\alpha = 1.4$	86.46	79.79	93.13	88.64	93.54	97.68	<b>78.80</b>	61.12	67.47	95.72	97.60	<b>38.40</b>
	$\alpha = 1.7$	<b>87.11</b>	<b>80.42</b>	93.80	92.93	<b>95.58</b>	97.72	62.80	<b>61.78</b>	70.34	<b>97.30</b>	<b>97.93</b>	29.60
	$\alpha = 2.0$	85.98	79.10	<b>93.87</b>	<b>93.88</b>	93.93	97.96	51.20	60.57	<b>70.38</b>	96.92	97.69	17.60
<b>(b) Inference Steps</b>													
—	2 steps	79.92	67.97	91.87	77.18	92.90	98.18	16.80	51.69	55.44	94.20	97.55	30.40
	5 steps	86.03	77.59	94.48	88.53	95.65	<b>98.25</b>	50.00	60.21	65.96	97.03	99.25	29.20
	10 steps	87.69	80.55	<b>94.83</b>	91.39	<b>96.02</b>	98.10	66.40	<b>62.42</b>	68.57	<b>97.78</b>	<b>99.33</b>	26.40
	20 steps	<b>87.84</b>	<b>81.17</b>	94.51	<b>89.54</b>	95.89	98.10	<b>79.88</b>	61.02	<b>68.99</b>	97.07	99.10	<b>31.10</b>
<b>(c) Training Iterations</b>													
150	$\alpha = 1.0$	78.23	73.36	83.11	80.29	86.09	98.63	50.80	<b>60.19</b>	<b>63.62</b>	88.67	88.47	<b>34.00</b>
	$\alpha = 1.4$	79.33	73.01	85.65	81.61	86.20	98.74	48.00	59.32	61.92	90.79	91.11	26.91
	$\alpha = 1.7$	82.79	74.99	90.60	84.37	89.92	98.84	48.40	60.05	63.25	93.62	95.99	31.60
	$\alpha = 2.0$	<b>84.31</b>	<b>76.25</b>	<b>92.37</b>	<b>87.23</b>	<b>91.86</b>	<b>98.86</b>	<b>53.97</b>	59.03	62.36	<b>95.71</b>	<b>97.04</b>	30.20
450	$\alpha = 1.0$	79.27	74.20	84.35	82.20	86.81	<b>98.32</b>	59.20	59.48	62.62	90.32	89.00	<b>33.60</b>
	$\alpha = 1.4$	85.72	79.41	92.04	88.07	93.34	98.12	<b>73.54</b>	61.88	66.69	94.94	96.98	32.94
	$\alpha = 1.7$	<b>87.37</b>	<b>80.95</b>	<b>93.80</b>	<b>91.67</b>	<b>95.51</b>	97.61	72.54	<b>62.25</b>	<b>69.91</b>	<b>96.87</b>	<b>98.23</b>	30.34
	$\alpha = 2.0$	85.96	79.54	92.39	92.88	94.12	97.46	60.69	61.04	70.22	96.51	97.52	14.51
750	$\alpha = 1.0$	80.17	74.35	85.99	83.16	86.99	<b>98.05</b>	62.40	58.44	62.49	91.23	90.70	34.40
	$\alpha = 1.4$	86.46	79.79	93.13	88.64	93.54	97.68	<b>78.80</b>	61.12	67.47	95.72	97.60	<b>38.40</b>
	$\alpha = 1.7$	<b>87.11</b>	<b>80.42</b>	93.80	92.93	<b>95.58</b>	97.72	62.80	<b>61.78</b>	70.34	<b>97.30</b>	<b>97.93</b>	29.60
	$\alpha = 2.0$	85.98	79.10	<b>93.87</b>	<b>93.88</b>	93.93	97.96	51.20	60.57	<b>70.38</b>	96.92	97.69	17.60
900	$\alpha = 1.0$	81.78	74.63	88.93	84.23	88.48	98.44	60.73	58.56	60.10	93.14	93.73	<b>32.80</b>
	$\alpha = 1.4$	<b>87.69</b>	<b>80.55</b>	<b>94.83</b>	91.39	<b>96.02</b>	98.10	<b>66.40</b>	<b>62.42</b>	68.57	<b>97.78</b>	<b>99.33</b>	26.40
	$\alpha = 1.7$	86.93	79.19	94.67	94.78	95.80	98.22	41.20	61.84	<b>70.14</b>	98.31	99.26	18.00
	$\alpha = 2.0$	85.91	77.96	93.86	<b>95.15</b>	94.26	<b>98.38</b>	32.40	60.57	70.17	98.22	98.95	6.15
1200	$\alpha = 1.0$	82.08	75.01	89.14	84.81	88.30	98.15	63.20	58.17	61.90	93.02	94.04	<b>34.40</b>
	$\alpha = 1.4$	<b>87.32</b>	<b>80.30</b>	<b>94.34</b>	90.72	<b>95.44</b>	97.58	<b>73.20</b>	<b>61.51</b>	68.01	<b>97.19</b>	<b>98.83</b>	29.89
	$\alpha = 1.7$	86.86	79.70	94.02	<b>93.66</b>	95.46	98.02	54.80	60.77	<b>69.70</b>	97.65	98.77	18.80
	$\alpha = 2.0$	86.01	78.32	93.69	94.21	94.16	<b>98.40</b>	41.20	59.51	69.96	97.90	98.74	9.16

<sup>2</sup><https://github.com/modelscope/DiffSynth-Studio>

### 3.2 BASELINE COMPARISON

As shown in Table 1, Pusa, with only 10 inference steps, achieves SOTA performance among open-source models and surpasses its direct baseline, Wan-I2V, which was trained with vastly greater resources. Our model obtains a total score of 87.32, outperforming Wan-I2V’s 86.86. This is achieved despite using less than 1/2500th of the training data ( $4K$  vs.  $\geq 10M$  samples) and 1/200th of the computational budget.

Notably, Pusa demonstrates superior performance in key I2V metrics, such as I2V Background Consistency (99.24 vs. 96.44) and I2V Subject Consistency (97.64 vs. 96.95), indicating a more faithful adherence to the input image condition. Furthermore, our model exhibits a higher Dynamic Degree (52.60 vs. 51.38), producing more motion-rich videos while maintaining high Motion Smoothness (98.49 vs. 97.90).

### 3.3 HYPERPARAMETER STUDY

To validate our hyperparameter choices and understand their impact on performance, we conducted a series of studies, summarized in Table 2.

**Lora Configurations** Lora rank is a critical ingredient that influences the finetuning performance. As we know, Lora learns much less with small ranks compared to full finetuning Biderman et al. (2024), thus, Lora rank should be large enough to have the capacity to learn the new capabilities since tasks like I2V is a very general. We investigated the influence of LoRA rank, a proxy for the adaptation’s capacity. As shown in Table 2(a), a higher rank of 512 consistently outperforms a rank of 256 across most metrics, particularly in overall quality. This suggests that a larger adaptation capacity is beneficial for capturing the nuances of temporal dynamics required for I2V tasks. We also find that the LoRA alpha scaling at inference time is critical; an alpha of 1.7 yields the best results for the 750-iteration checkpoint, balancing the influence of the LoRA weights against the pretrained model.

**Inference Steps.** As detailed in Table 2(b), we analyzed the trade-off between computational cost at inference and generation quality using our best checkpoint with rank 512 and alpha 1.4 with 900 iterations. Performance scales predictably with the number of steps, with significant gains observed up to 10 steps. While 20 steps provide a marginal improvement, the results at 10 steps are nearly identical (87.69 vs. 87.84). Consequently, we adopt 10 inference steps as our default to ensure an optimal balance between quality and generation speed.

**Training Progression.** We evaluated checkpoints of Lora rank 512 at various stages of training, from 150 to 1200 iterations, using 10 inference steps. Table 2(c) shows a clear trend of improving performance up to 900 iterations, where we achieve our peak score of 87.69 with an alpha of 1.4. Beyond this point, performance begins to plateau or slightly degrade, indicating that the model has converged. This rapid convergence underscores the data efficiency of our approach. Our final model for comparison in Table 1 uses the 900-iteration checkpoint of rank 512.

### 3.4 ANALYSIS OF THE ADAPTATION MECHANISM

We now delve into the underlying mechanisms that enable Pusa’s remarkable efficiency and performance. Our analysis reveals that the Pusa framework facilitates a highly targeted adaptation that leverages, rather than overwrites, the pretrained knowledge of the foundation model.

**I2V Qualitative** In Fig. 3, Wan-T2V with VTA uses the same configurations as our method to do I2V generation, which is by directly setting the first frame/input image noise free. Since the model is purely for the T2V task, the generated frames are just aligned with the text prompt but have no relation to the first frame. Meanwhile, after finetuning of Wan-T2V model, our Pusa model can generate content aligned with both text and the input image seamlessly, and the result is better than Wan-I2V, which has visible distortions and does not preserve the character well from the condition image.

**Attention Mechanism.** A visualization of the self-attention maps using queries and keys within the final Transformer (block 39) block across different inference steps (10 steps in total, we visu-

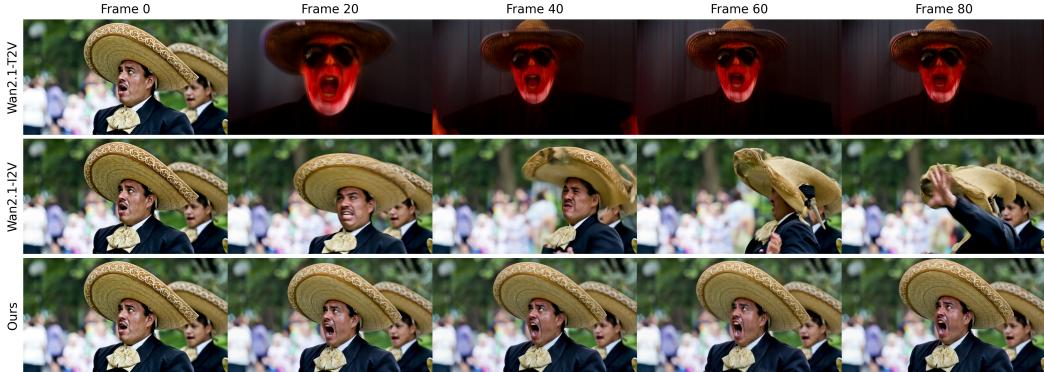


Figure 3: Image-to-video results, where our method generates a smooth and realistic animation of the first frame while Wan2.1-T2V generates completely different frames from the first frame, though aligned with text prompt, and Wan2.1-I2V generates frames with noticeable distortions. All generated with the same input image (Frame 0) and text prompt: "A man in a black suit and a sombrero, shouting loudly".

alize steps 0, 3, 6, and 9) (Fig. 4) provides critical insights. The base Wan-T2V model exhibits a diagonal attention pattern, indicating that each frame primarily attends to itself, with little temporal cross-attention. In contrast, both Wan-I2V and Pusa show strong attention from all frames to the first frame (the first column of the map), which is essential for maintaining consistency with the input image. However, a key difference emerges: the attention patterns in Wan-I2V are globally altered compared to the base model, and the attention to the first frame only strengthened in the early inference steps (e.g., step 0), which indicates lots of ineffective learning redundancy. In Pusa, the attention to the first frame is significantly modified across all steps. The rest of the attention map resembles that of the original Wan-T2V more closely than Wan-I2V. This demonstrates that our method successfully decouples the learning of temporal dynamics and basic generation priors, surgically injects the conditioning mechanism while preserving the base model’s powerful, pretrained capabilities. Wan-I2V, on the other hand, appears to have learned a fundamentally new attention distribution, revealing the disadvantages of extensive retraining with its I2V method.

**Parameter Divergence.** This observation is further substantiated by an analysis of parameter changes (Fig. 5). The parameter drift in Wan-I2V is substantial and concentrated in modules critical for content generation, such as the text encoder and cross-attention blocks. This implies a significant alteration of the model’s core generative priors. Pusa, conversely, exhibits minimal parameter changes, with modifications almost exclusively in the self-attention blocks responsible for temporal dynamics. The magnitude of parameter change in Wan-I2V is over an order of magnitude larger than in Pusa. This confirms that our approach constitutes a minimal, targeted adaptation, preserving the integrity of the foundation model and explaining its efficiency.

**Why Vectorized Timestep Adaptation Succeeds.** A fundamental challenge of VTV, as first identified in the FVDM paper Liu et al. (2024b), is the combinatorial explosion of the temporal composition space. With each frame possessing an independent timestep, the number of possible configurations grows exponentially (e.g., to  $10^{48}$  for 16 frames), making convergence from scratch exceedingly difficult. The FVDM paper introduced the PTSS to solve this, decoupling the learning of temporal dynamics from foundational generation capability by alternating between synchronized and asynchronous timesteps during training.

In our work, this challenge is elegantly circumvented by leveraging a powerful, pretrained foundation model. Since the Wan-T2V model has already mastered video content generation, our adaptation does not need to learn this capability from scratch. Instead, our finetuning method can focus exclusively on mastering the temporal control offered by the VTV architecture with totally random timesteps for all frames. The base model’s robust generative prior means only a brief period of finetuning with independent timesteps is required to instill this new, fine-grained control.

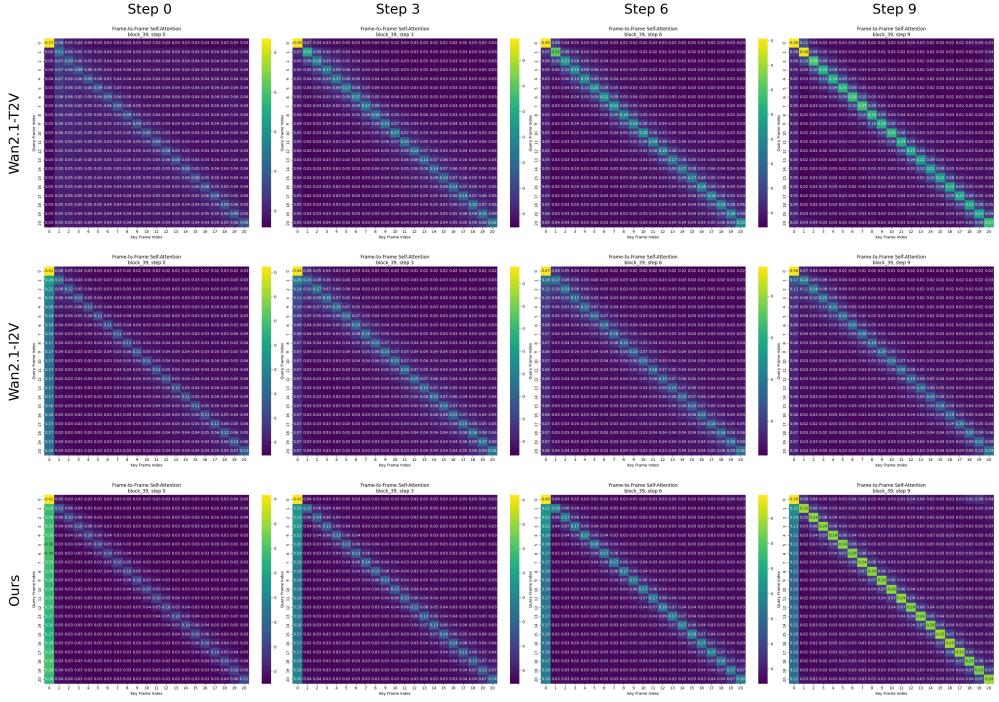


Figure 4: Visualization of attention maps corresponding to the generation of videos in Fig. 3. Each value in the attention map represents frame-to-frame correlation/attention, a larger value means higher correlation. Zoom in for better view.

Evidence of the base model’s inherent robustness to timestep desynchronization is presented in Fig. 3. When we attempt zero-shot I2V generation by setting the first frame’s timestep to zero, the base Wan-T2V model still produces a coherent video that aligns with the text prompt, even though it fails to adhere to the image condition. This demonstrates that the generative processes for frames are originally independent, not significantly influenced by other frames. The mechanistic explanation lies in the model’s attention structure (Fig. 4): the strongly diagonal self-attention patterns in Wan-T2V reveal that each frame’s synthesis relies primarily on its own latent features. Our fine-tuning surgically modifies this behavior, introducing targeted attention to conditioning frames while preserving the model’s stable generative core. This non-destructive adaptation is the key to sidestepping the VTV compositionality problem and achieving state-of-the-art results with unprecedented efficiency.

### 3.5 MULTI-TASK CAPABILITIES

A key advantage of our approach is its ability to generalize to a variety of video generation tasks, including and beyond T2V, without any task-specific training. This is a direct result of the flexible vectorized timestep settings, which allow for arbitrary conditioning on any subset of frames.

**Text-to-Video Generation.** Unlike specialized I2V models, Pusa still retains the text-to-video capabilities of its foundation model. As shown in Fig. 13, the qualitative output remains high. This demonstrates that our fine-tuning process does not cause catastrophic forgetting of the primary T2V task, making Pusa a truly unified video generation model.

**Complex Temporal Tasks.** The true power of the FVDM framework is revealed in its zero-shot performance on complex temporal synthesis tasks. More results of seamless I2V generation is given in Fig. 3. Apart from that, Pusa can perform start-end frames by conditioning on the first frame and the last frame (encoded to a single latent frame like the first frame) in Fig. 7, or conditioning on the first frame and the last 4 frames (encoded to a single latent frame as default) in Fig. 9. Since

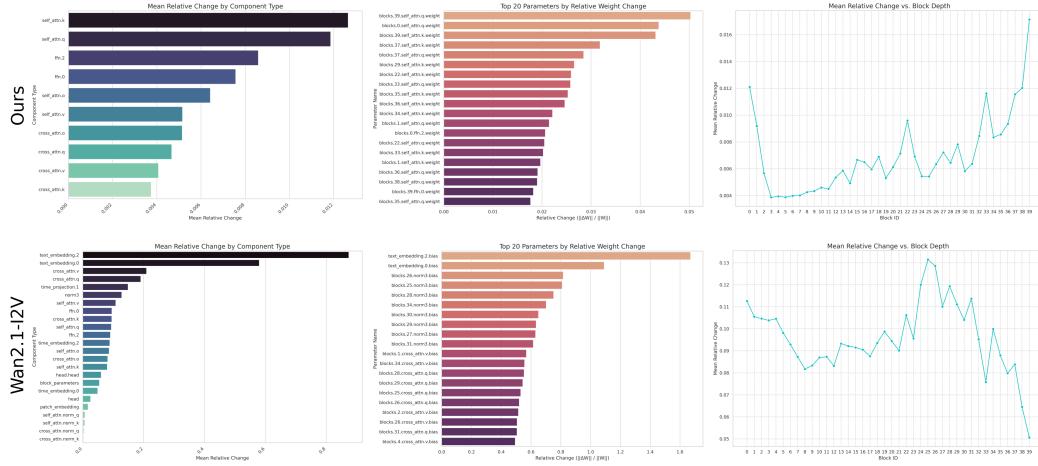


Figure 5: Analysis on finetuned model’s parameter shifts. The columns, from left to right, represent average relative change of parameters by model component/modules, top 20 parameters with largest relative change, and average change by transformer blocks (in total 40 blocks from 0 to 39). Overall, our model has less variations to Wan2.1-T2V compared to Wan2.1-I2V, which indicates our model preserves the original distribution better. Note that for our model, we use our final configurations in Table 1, i.e., Lora rank 512 with lora 1.4 at 900 steps. Zoom in for a better view.

there is a  $4\times$  compression rate for the last frames brought by VAE, only conditioning on the last frame yields worse results, as it will be viewed as 4 still frames. Thanks to the unique properties of our framework, we can add some noise (e.g., set  $\tau^1 = 0.3 * t$  and  $\tau^N = 0.7 * t$ ) to the encoded latents to generate motion and content for the condition frames and thus synthesize more coherent videos as shown in Fig. 8. Furthermore, Figures 10 11 and 12 showcase its capability for video completion/transition, video extension, seamlessly completing or continuing a given video sequence. These capabilities are all emergent properties of our vectorized timestep adaptation strategy and require no task-specific training, highlighting the versatility and power of our approach.

## 4 RELATED WORK

The field of video generation has rapidly evolved, driven by the success of diffusion models in image synthesis. Our work builds upon and extends several key research threads, positioning itself as a highly efficient and versatile solution for the next paradigm of video diffusion models.

### 4.1 CONVENTIONAL VIDEO DIFFUSION MODELS

The initial extension of image diffusion models to video generation established a foundational paradigm. Seminal works like VDM Ho et al. (2022) and subsequent large-scale models such as Latent Video Diffusion Models (LVDM) He et al. (2022), VideoCrafter1 Chen et al. (2023), and others Wang et al. (2023); Ma et al. (2024); Wan et al. (2025); Kong et al. (2024) all adopted the same diffusion framework. A core characteristic of these conventional models is their reliance on a scalar timestep variable. This single variable governs the noise level and evolution trajectory uniformly across all frames of a video clip during the diffusion process.

While this synchronized-frame approach proved effective for generating short, self-contained clips, particularly for T2V tasks, it imposes a rigid temporal structure. The uniform noise schedule inherently limits the model’s ability to handle tasks requiring asynchronous frame evolution, such as I2V generation, where the first frame or condition image is given, or complex editing tasks like video interpolation. Recognizing the limitations of conventional VDMs in temporal modeling, the research community has developed numerous extensions targeting specific video generation tasks Xing et al. (2023b). These approaches predominantly focus on adapting existing scalar timestep based models through fine-tuning strategies or zero-shot techniques to handle domain-specific challenges such

as image-to-video generation Xing et al. (2023a); Guo et al. (2023); Zhang et al. (2023); Li et al. (2024); Ni et al. (2024), video interpolation Wang et al. (2024b;a); Wan et al. (2025), and long video synthesis Duan et al. (2024); Henschel et al. (2024); Kim et al. (2024); Lu et al. (2024); Dalal et al. (2025); Zhao et al. (2025) These methods typically involve extensive fine-tuning of a large, pre-trained T2V model on task-specific data or employing zero-shot domain transfer techniques.

I2V generation has emerged as a particularly active area. For example, the Wan-I2V model presented in the Wan paper required fine-tuning Wan T2V model on its T2V pretraining dataset to achieve its SOTA I2V capabilities and can only do I2V after this process Wan et al. (2025). Overall, these extensions reveal fundamental challenges in balancing flexibility, generalization, and the retention of original model capabilities. Fine-tuning approaches often suffer from catastrophic forgetting, where adaptation to specific tasks severely degrades performance on its original capabilityPan et al. (2024); Ramasesh et al. (2021). Zero-shot methods, exemplified by TI2V-Zero introduces a zero-shot method for conditioning T2V diffusion models on images Ni et al. (2024). However, its generalization and generation quality are limited by potential visual artifacts and reduced robustness, as its simple "repeat-and-slide" strategy struggles with diverse input and can produce blurry or flickering videos. The reliance on task-specific architectures and training procedures highlights the need for a more unified and general approach that can handle diverse video generation scenarios without requiring extensive finetuning. Our work departs from these limitations by adopting the VTV introduced by FVDM Liu et al. (2024b), enabling fine-grained control over the generative process.

#### 4.2 AUTOREGRESSIVE VIDEO DIFFUSION MODELS

Recently, people have explored autoregressive paradigms for video diffusion models, where frames are generated sequentially rather than simultaneously (Chen et al., 2024; Sun et al., 2025; Teng et al., 2025; Chen et al., 2025; Huang et al., 2025). This direction includes methods like Diffusion Forcing, which trains a causal next-token model to predict one or multiple future tokens without fully diffusing past ones, enabling variable-length generation capabilities. CauseVid Yin et al. (2024) represents another significant advancement in this direction, proposing fast autoregressive video diffusion models that can generate frames on-the-fly with streaming capabilities.

Large-scale autoregressive models such as MAGI-1 Teng et al. (2025) and SkyReels-V2 Chen et al. (2025) have demonstrated the potential for scalable video generation through chunk-by-chunk processing, where each segment is denoised holistically before proceeding to the next. Self-Forcing Huang et al. (2025) addresses the critical issue of exposure bias in autoregressive video diffusion by introducing a training paradigm where models condition on their own previously generated outputs rather than ground-truth frames. This approach enables real-time streaming video generation while maintaining temporal coherence through innovative key-value caching mechanisms.

Despite these advances, autoregressive video diffusion models face inherent limitations that constrain their applicability. The sequential nature of generation restricts these models to unidirectional tasks, making them inadequate for many scenarios, such as start-end frames, video transitions, and keyframe interpolation. Moreover, error accumulation and drift issues represent persistent challenges in autoregressive approaches, where small prediction errors compound over time, leading to quality degradation in longer sequences Huang et al. (2025). Recent theoretical analyses have also identified both error accumulation and memory bottlenecks as fundamental phenomena in autoregressive video diffusion models, revealing a Pareto frontier between these competing constraintsWang et al. (2025).

#### 4.3 FRAME-AWARE VIDEO DIFFUSION MODEL

Frame-aware video diffusion model (FVDM) Liu et al. (2024b) is a parallel line of research to reconstruct the paradigm for video diffusion models, by enabling independent temporal evolution for each frame. This paradigm shift addresses fundamental shortcomings in conventional VDMs by enhancing the model's capacity to capture fine-grained temporal dependencies without the constraints of synchronized evolution.

The vectorized timestep approach enables unprecedented flexibility across multiple video generation tasks, including T2V, I2V, start-end frames, video extension, and so on, all within a single unified

framework. Unlike conventional approaches that require extensive fine-tuning and destructive architectural modifications, the FVDM-based paradigm demonstrates strong zero-shot capabilities across diverse scenarios, while only using minor finetuning to adapt the model to support VTV. Project Pusa-Mochi (V0.5) Liu & Liu (2025); Team (2024) have initially demonstrated the practical viability of this approach. Further in this work, Pusa-Wan (V1.0) achieves remarkable efficiency gains with training costs reduced to mere 500 dollars from above 100K of Wan-I2V, while outperforming it on Vbench-I2V.

The FVDM framework represents a fundamental departure from previous temporal modeling approaches, offering a solution that can perform both directional generation tasks (like autoregressive models) and bidirectional temporal reasoning tasks that autoregressive approaches cannot handle. This unified capability, combined with the demonstrated computational efficiency and strong empirical results, positions this approach as a promising direction for next-generation video diffusion models.

## 5 CONCLUSION

This work introduces a transformative approach to video diffusion models through the FVFM framework, culminating in Pusa V1.0. By decoupling temporal dynamics from content generation, our method achieves SOTA level I2V performance with unprecedented efficiency, requiring only \$500 and 4K samples to surpass Wan-I2V. The key innovation lies in our non-destructive adaptation strategy: preserve the foundation model’s robust priors while enabling frame-specific evolution via VTV. This unlocks many zero-shot generalization to diverse tasks—a property unmatched by conventional or autoregressive VDMs.

Our mechanistic studies (e.g., attention maps, parameter drift) demonstrate that Pusa’s success stems from its minimal, targeted modifications to the base model’s temporal attention mechanisms, avoiding the catastrophic forgetting observed in task-specific fine-tuning. The implications are profound: Pusa redefines the efficiency-quality tradeoff in video generation, enabling high-fidelity, multi-task generation at a fraction of traditional costs. Future directions include extending VTV to long-form video generation and dynamic scene editing. By bridging theoretical innovation with practical deployment, this work paves the way for a new era of scalable, general-purpose video diffusion models, with far-reaching applications in creative industries, education, and beyond.

## REFERENCES

- Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, et al. Lora learns less and forgets less. *arXiv preprint arXiv:2405.09673*, 2024.
- 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.
- 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*, 2025.
- 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.
- Ricky T. Q. Chen. torchdiffeq, 2018. URL <https://github.com/rtqichen/torchdiffeq>.
- Karan Dalal, Daniel Koceja, Jiarui Xu, Yue Zhao, Shihao Han, Ka Chun Cheung, Jan Kautz, Yejin Choi, Yu Sun, and Xiaolong Wang. One-minute video generation with test-time training. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 17702–17711, 2025.
- Zhongjie Duan, Wenmeng Zhou, Cen Chen, Yaliang Li, and Weining Qian. Exvideo: Extending video diffusion models via parameter-efficient post-tuning. *arXiv preprint arXiv:2406.14130*, 2024.

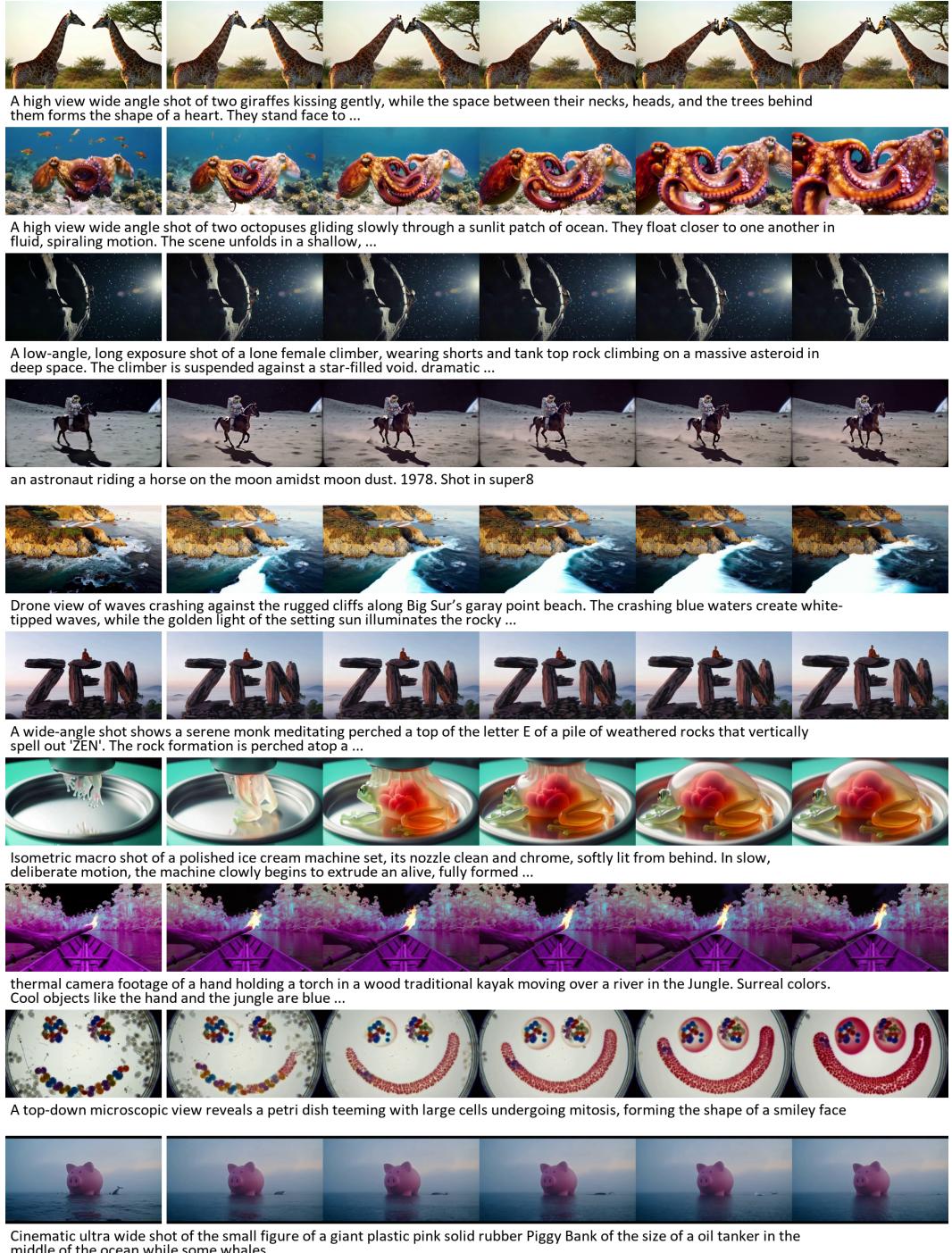


Figure 6: More image-to-video results. The first frames of each row are the given condition images extracted from Veo2 & Sora demos. Each generated video has 81 frames in total.

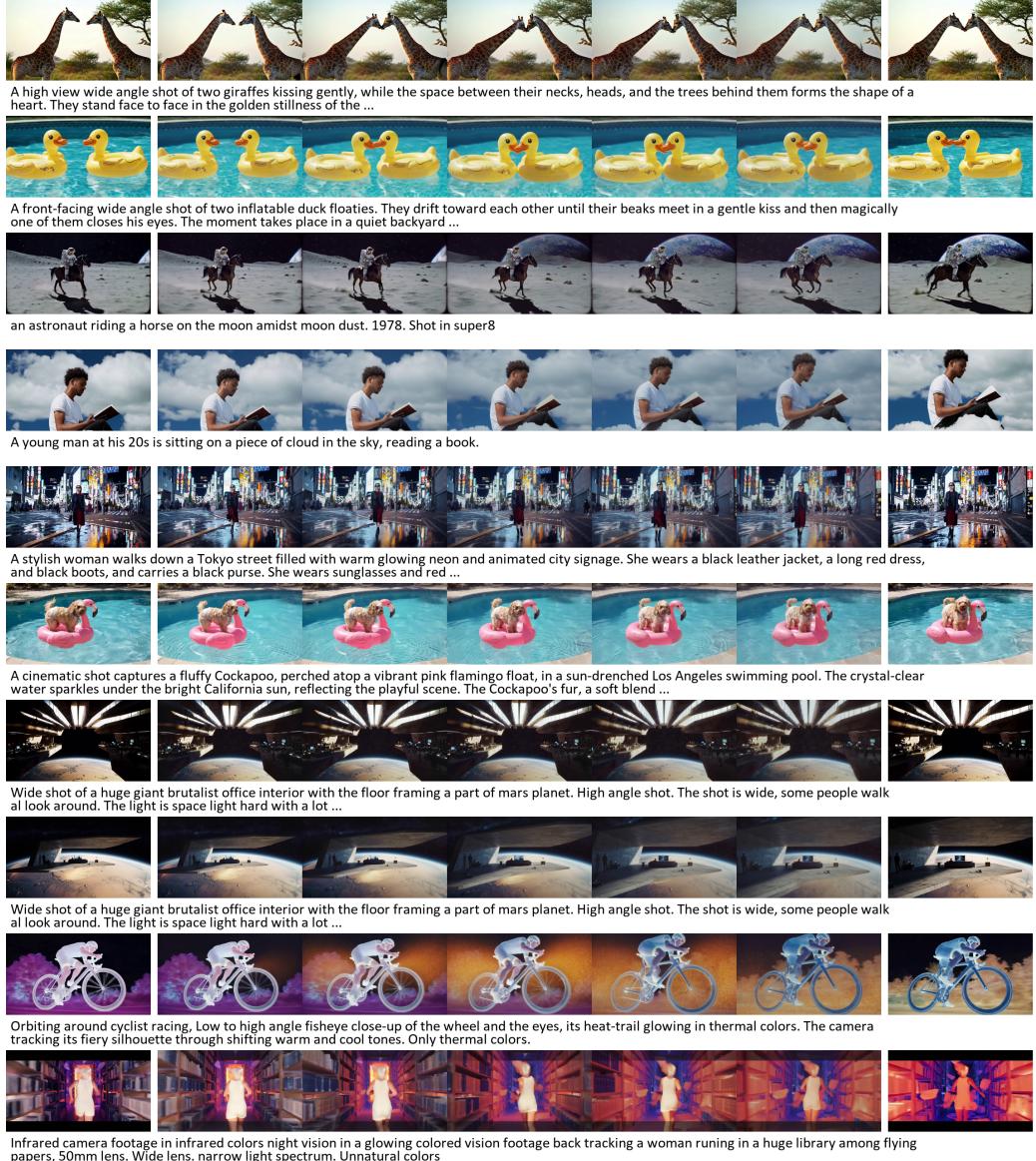


Figure 7: Zero-shot results w.r.t. start & end frames to video. The first and last frames are given condition frames extracted from Veo2 & Sora demos. Each generated video has 81 frames in total.

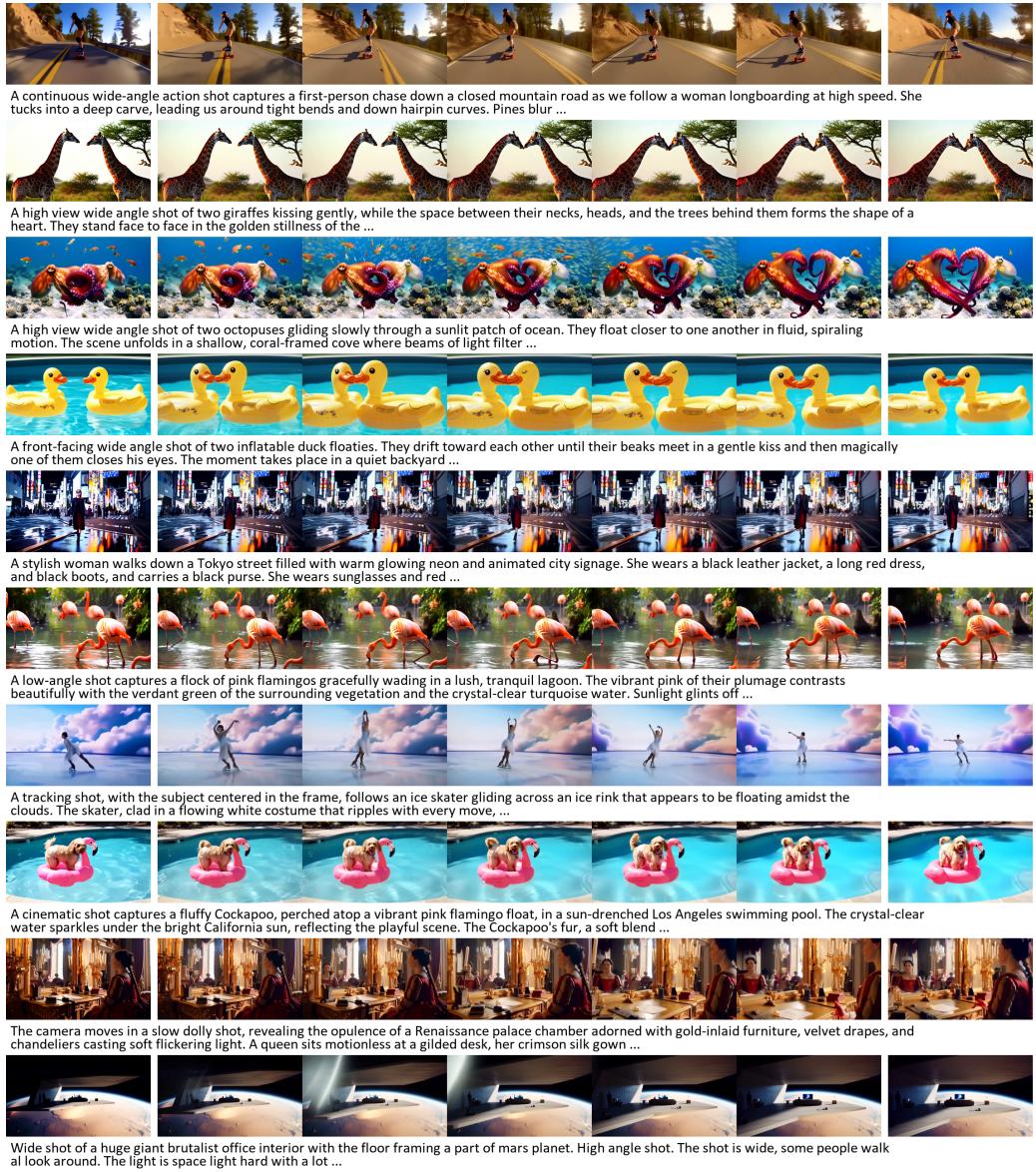


Figure 8: Zero-shot results w.r.t. start & end frames with noise. The first and last frames are given conditions and added 30% and 70% noise during sampling to make the generated video more coherent. Condition frames are extracted from Veo2 & Sora demos. Each generated video has 81 frames in total.

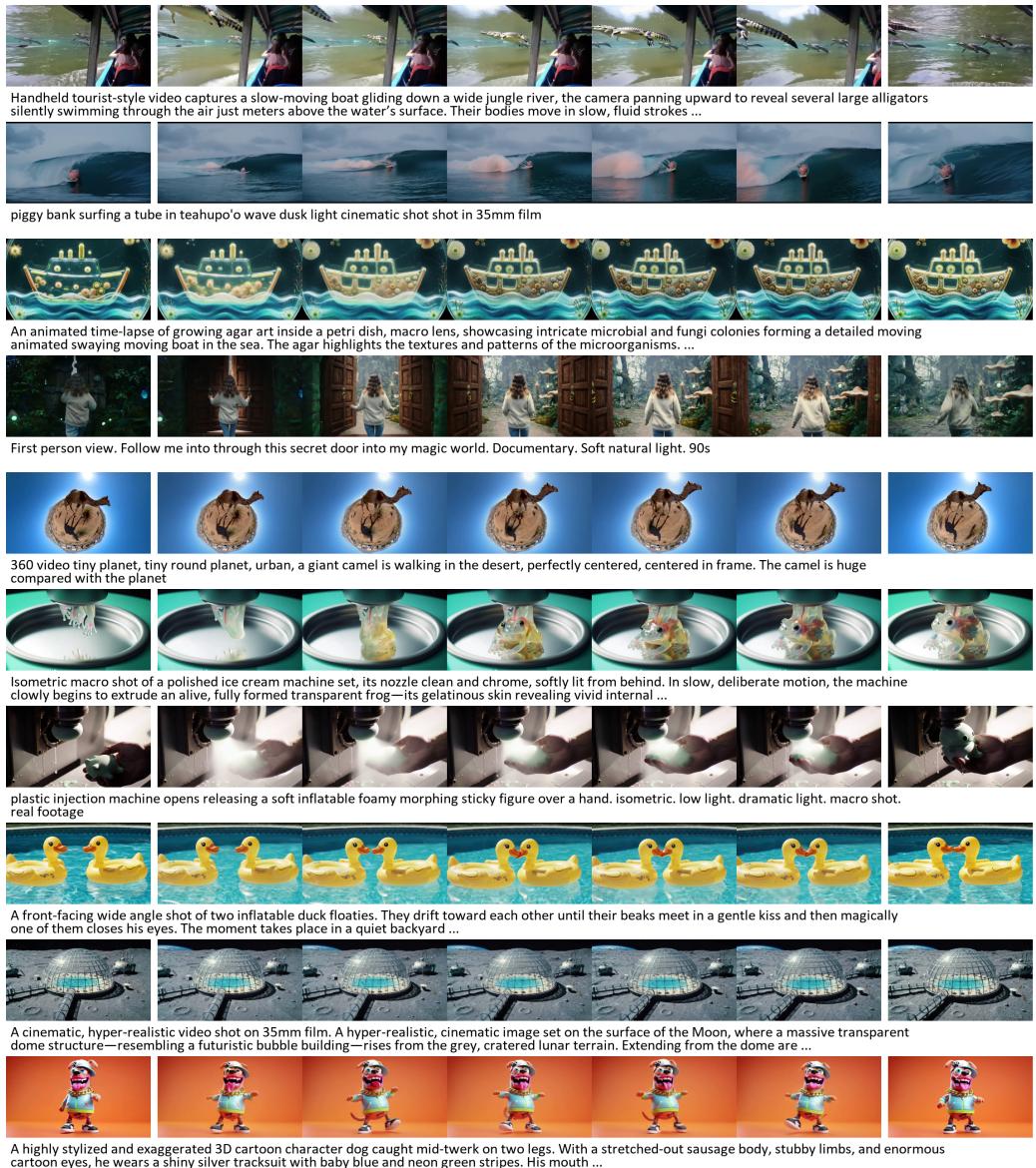


Figure 9: Zero-shot results w.r.t. start & end frames to video. The first and last 4 frames (encoded to one latent frame) are given condition frames extracted from Veo2 & Sora demos. Each generated video has 81 frames/21 latent frame in total.

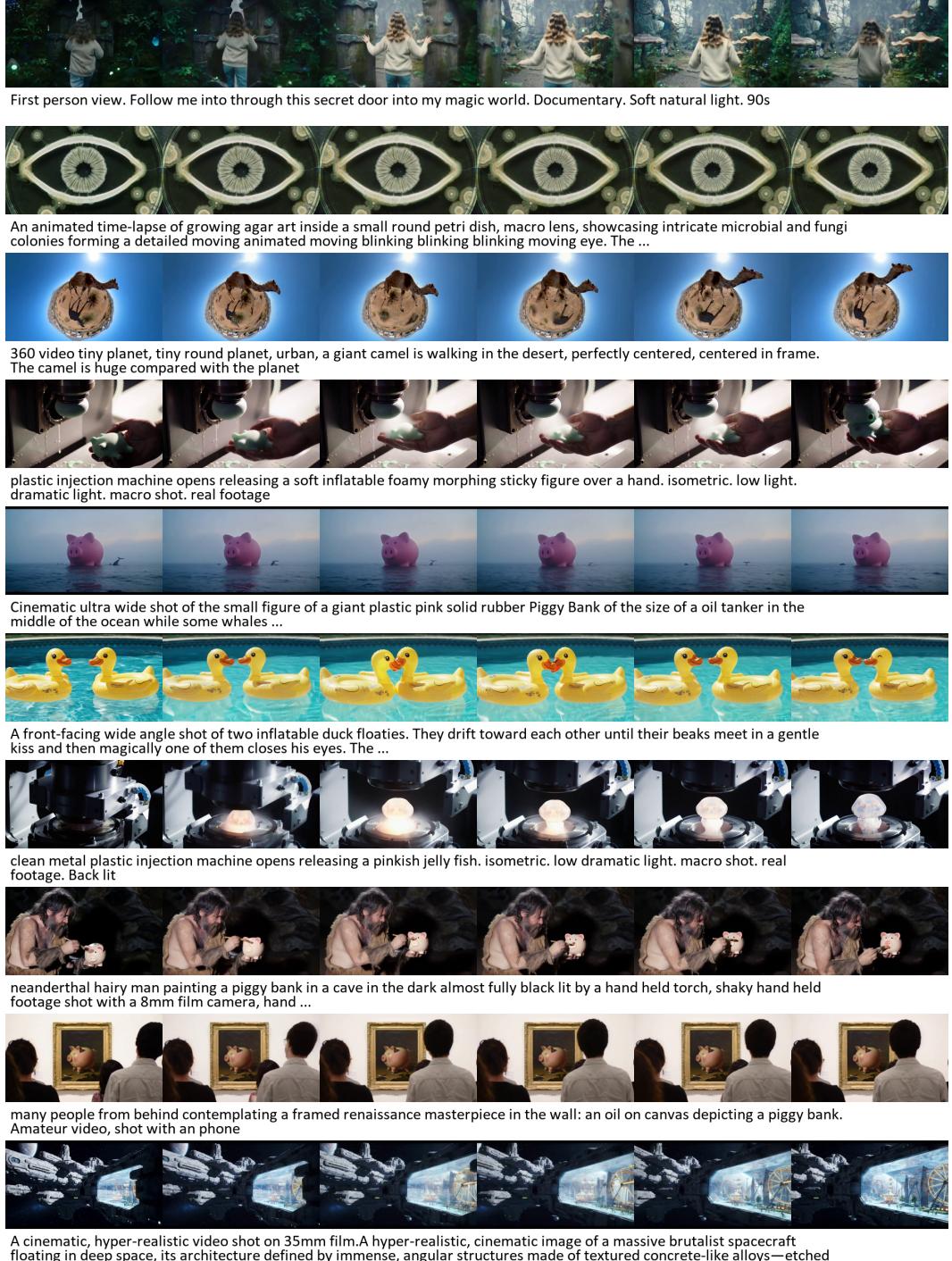


Figure 10: Zero-shot results w.r.t. video completion/transition. The first 9 frames and the last 12 frames extracted from Veo2 demos are given as conditions and encoded to the first 3 latent frames and the last 3 latent frames, respectively. Each generated video has 81 frames/21 latent frames in total.

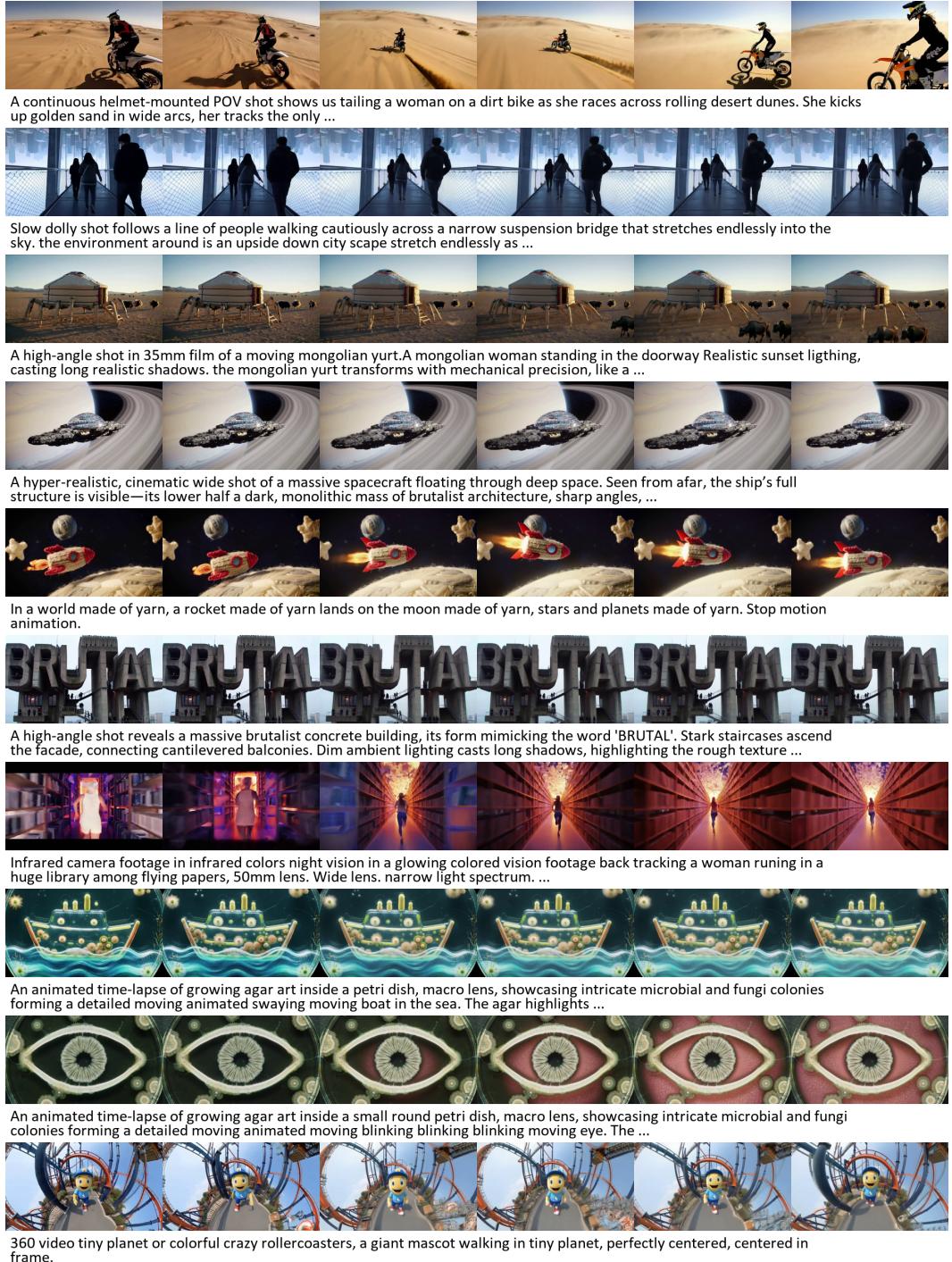


Figure 11: Zero-shot results w.r.t. video extension. The first 13 frames extracted from Veo2 demos are given as conditions and encoded to the first 4 latent frames. Each generated video has 81 frames/21 latent frames in total.

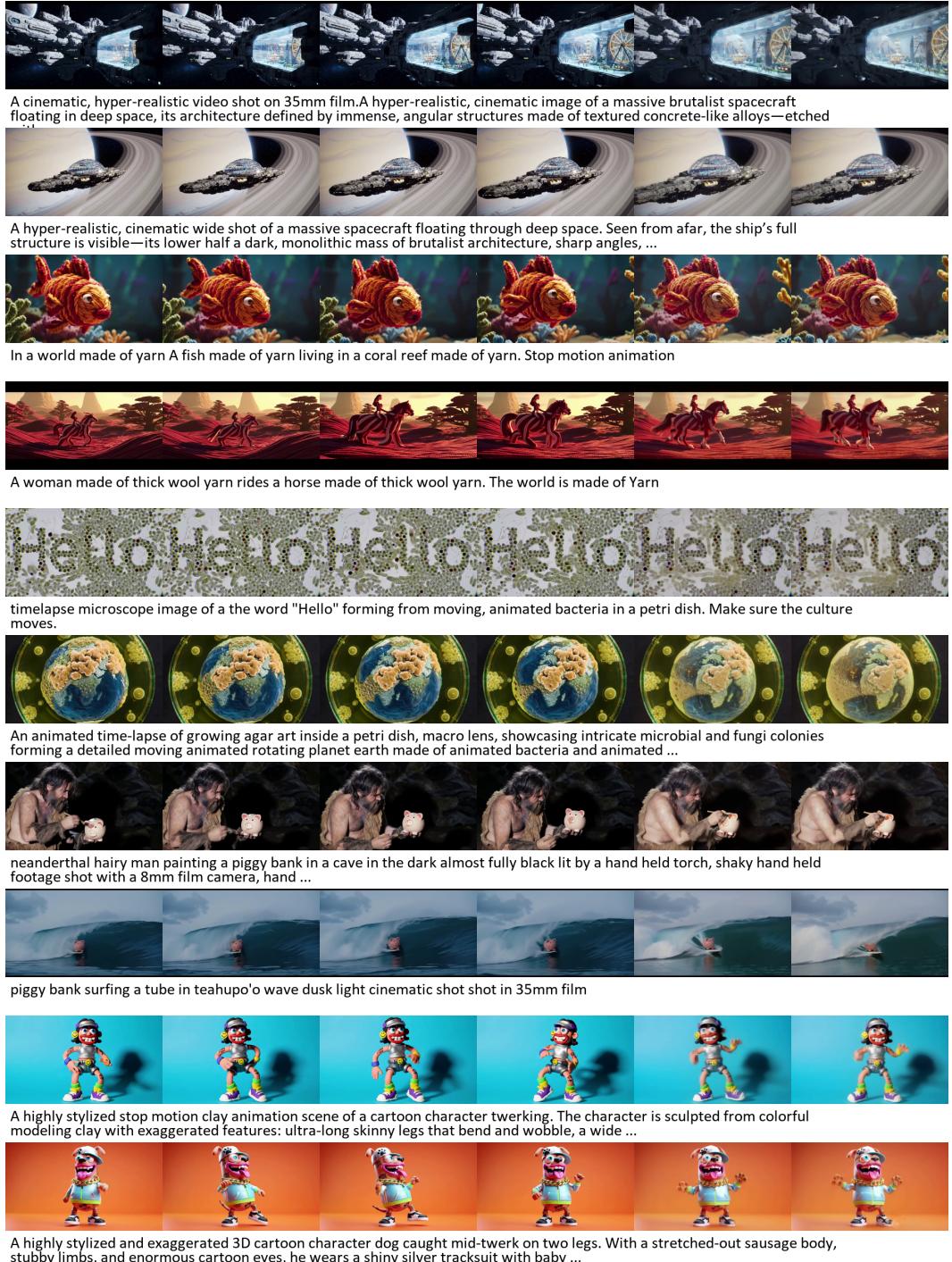


Figure 12: Zero-shot results w.r.t. video extension. The first 41 frames extracted from Veo2 demos are given as conditions and encoded to the first 11 latent frames. Each generated video has 81 frames/21 latent frames in total.

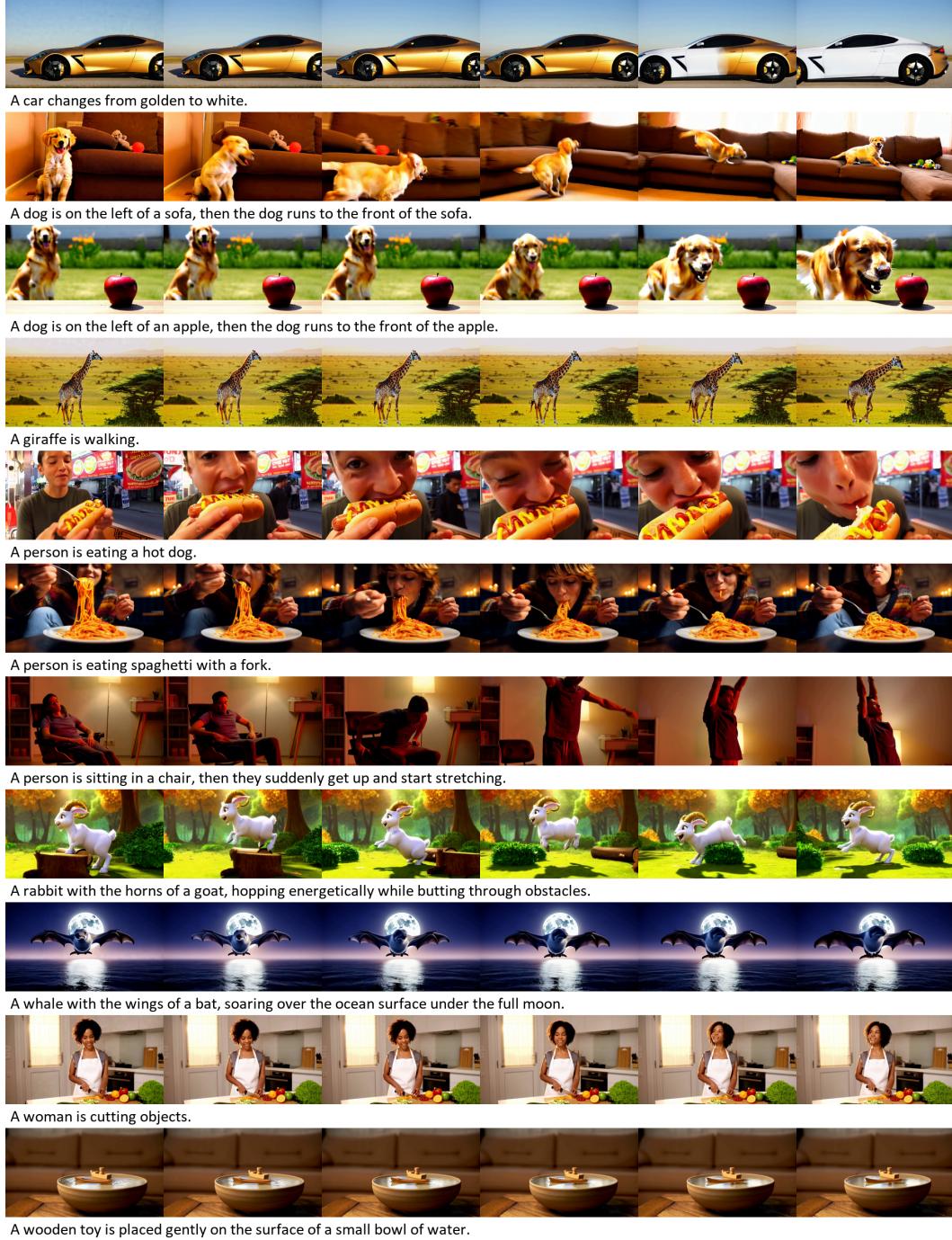


Figure 13: Text-to-video results. Prompts all from Vbench2.0.

- Xun Guo, Mingwu Zheng, Liang Hou, Yuan Gao, Yufan Deng, Chongyang Ma, Weiming Hu, Zhengjun Zha, Haibin Huang, Pengfei Wan, et al. I2v-adapter: A general image-to-video adapter for video diffusion models. *arXiv preprint arXiv:2312.16693*, 2023.
- Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. Latent video diffusion models for high-fidelity long video generation. 2022.
- Roberto Henschel, Levon Khachatryan, Daniil Hayrapetyan, Hayk Poghosyan, Vahram Tadevosyan, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Streaming2v: Consistent, dynamic, and extendable long video generation from text. *arXiv preprint arXiv:2403.14773*, 2024.
- 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.
- 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.
- 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, Fan Zhang, Xiaoje 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.
- Jihwan Kim, Junoh Kang, Jinyoung Choi, and Bohyung Han. Fifo-diffusion: Generating infinite videos from text without training. *arXiv preprint arXiv:2405.11473*, 2024.
- Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuandvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.
- Weijie Li, Litong Gong, Yiran Zhu, Fanda Fan, Biao Wang, Tiezheng Ge, and Bo Zheng. Tuning-free noise rectification for high fidelity image-to-video generation. *arXiv preprint arXiv:2403.02827*, 2024.
- Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.
- 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.
- Yaofang Liu and Rui Liu. Pusa: Thousands timesteps video diffusion model, 2025. URL <https://github.com/Yaofang-Liu/Pusa-VidGen>.
- Yaofang Liu, Xiaodong Cun, Xuebo Liu, Xintao Wang, Yong Zhang, Haoxin Chen, Yang Liu, Tieyong Zeng, Raymond Chan, and Ying Shan. Evalrafter: Benchmarking and evaluating large video generation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22139–22149, 2024a.
- Yaofang Liu, Yumeng Ren, Xiaodong Cun, Aitor Artola, Yang Liu, Tieyong Zeng, Raymond H Chan, and Jean-michel Morel. Redefining temporal modeling in video diffusion: The vectorized timestep approach. *arXiv preprint arXiv:2410.03160*, 2024b.
- Yu Lu, Yuanzhi Liang, Linchao Zhu, and Yi Yang. Freelong: Training-free long video generation with spectralblend temporal attention. *arXiv preprint arXiv:2407.19918*, 2024.
- Xin Ma, Yaohui Wang, Gengyun Jia, Xinyuan Chen, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *arXiv preprint arXiv:2401.03048*, 2024.

Haomiao Ni, Bernhard Egger, Suhas Lohit, Anoop Cherian, Ye Wang, Toshiaki Koike-Aokino, Sharon X Huang, and Tim K Marks. Ti2v-zero: Zero-shot image conditioning for text-to-video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9015–9025, 2024.

OpenAI. Sora: Creating video from text. <https://openai.com/sora>, 2024.

Jiadong Pan, Hongcheng Gao, Zongyu Wu, Taihang Hu, Li Su, Qingming Huang, and Liang Li. Leveraging catastrophic forgetting to develop safe diffusion models against malicious finetuning. *Advances in Neural Information Processing Systems*, 37:115208–115232, 2024.

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.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–16. IEEE, 2020.

Vinay Venkatesh Ramasesh, Aitor Lewkowycz, and Ethan Dyer. Effect of scale on catastrophic forgetting in neural networks. In *International conference on learning representations*, 2021.

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.

Mingzhen Sun, Weining Wang, Gen Li, Jiawei Liu, Jiahui Sun, Wanquan Feng, Shanshan Lao, SiYu Zhou, Qian He, and Jing Liu. Ar-diffusion: Asynchronous video generation with auto-regressive diffusion. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 7364–7373, 2025.

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

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.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *arXiv preprint arXiv:2302.00482*, 2023.

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.

Cong Wang, Jiaxi Gu, Panwen Hu, Haoyu Zhao, Yuanfan Guo, Jianhua Han, Hang Xu, and Xiaodan Liang. Easycontrol: Transfer controlnet to video diffusion for controllable generation and interpolation. *arXiv preprint arXiv:2408.13005*, 2024a.

Jing Wang, Fengzhuo Zhang, Xiaoli Li, Vincent YF Tan, Tianyu Pang, Chao Du, Aixin Sun, and Zhuoran Yang. Error analyses of auto-regressive video diffusion models: A unified framework. *arXiv preprint arXiv:2503.10704*, 2025.

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

Xiaojuan Wang, Boyang Zhou, Brian Curless, Ira Kemelmacher-Shlizerman, Aleksander Holynski, and Steven M Seitz. Generative inbetweening: Adapting image-to-video models for keyframe interpolation. *arXiv preprint arXiv:2408.15239*, 2024b.

Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Xintao Wang, Tien-Tsin Wong, and Ying Shan. Dynamiccrafter: Animating open-domain images with video diffusion priors. *arXiv preprint arXiv:2310.12190*, 2023a.

Zhen Xing, Qijun Feng, Haoran Chen, Qi Dai, Han Hu, Hang Xu, Zuxuan Wu, and Yu-Gang Jiang. A survey on video diffusion models. *arXiv preprint arXiv:2310.10647*, 2023b.

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 preprint arXiv:2412.07772*, 2024.

Shiwei Zhang, Jiayu Wang, Yingya Zhang, Kang Zhao, Hangjie Yuan, Zhiwu Qin, Xiang Wang, Deli Zhao, and Jingren Zhou. I2vgen-xl: High-quality image-to-video synthesis via cascaded diffusion models. *arXiv preprint arXiv:2311.04145*, 2023.

Min Zhao, Guande He, Yixiao Chen, Hongzhou Zhu, Chongxuan Li, and Jun Zhu. Reflex: A free lunch for length extrapolation in video diffusion transformers. *arXiv preprint arXiv:2502.15894*, 2025.

Dian Zheng, Ziqi Huang, Hongbo Liu, Kai Zou, Yinan He, Fan Zhang, Yuanhan Zhang, Jingwen He, Wei-Shi Zheng, Yu Qiao, et al. Vbench-2.0: Advancing video generation benchmark suite for intrinsic faithfulness. *arXiv preprint arXiv:2503.21755*, 2025.