

Appendix

A Proofs	2
A.1. Multi-Step Solving Method	2
B Discussion	3
B.1. Discussion on LGP and ACD	3
B.2. Contributions	4
B.3. Removing CFG	4
C Additional Experimental Settings	4
D Upsampling Module	4

A. Proofs

The following is based on consistency distillation [30].

A.1. Multi-Step Solving Method

Theorem A.1. Let $\Delta t := \max_{n \in \llbracket 1, N-1 \rrbracket} \{|t_{n+1} - t_n|\}$, and $f(\cdot, \cdot; \phi)$ be the target phased consistency function induced by the pre-trained diffusion model (empirical PF-ODE). Assume f_θ satisfies the Lipschitz condition: there exists $L > 0$ such that for all $t \in [\epsilon, T]$, x , and y , we have $\|f_\theta(x, t) - f_\theta(y, t)\|_2 \leq L\|x - y\|_2$. Assume further that for all $n \in \llbracket 1, N-1 \rrbracket$, the ODE solver called at t_{n+1} has local error uniformly bounded by $O((t_{n+1} - t_n)^{p+1})$ with $p \geq 1$. Then, if $\text{Dis}(f_\theta(x_{t_{n+m}}, t_{n+m}), f_\theta(\hat{x}_{t_n}^\phi, t_n)) = 0$, we have

$$\sup_{n,x} \|f_\theta(x, t_n) - f(x, t_n; \phi)\|_2 = O((\Delta t)^p).$$

Proof. From the loss $\text{Dis}(f_\theta(x_{t_{n+m}}, t_{n+m}), f_\theta(\hat{x}_{t_n}^\phi, t_n)) = 0$, we have:

$$f_\theta(x_{t_{n+m}}, t_{n+m}) \equiv f_\theta(\hat{x}_{t_n}^\phi, t_n). \quad (14)$$

Let $e_n := f_\theta(x_{t_n}, t_n) - f(x_{t_n}, t_n; \phi)$. We obtain the subsequent recursive formula:

$$\begin{aligned} e_{n+m} &= f_\theta(x_{t_{n+m}}, t_{n+m}) - f(x_{t_{n+m}}, t_{n+m}; \phi) \\ &\stackrel{(i)}{=} f_\theta(\hat{x}_{t_n}^\phi, t_n) - f(x_{t_n}, t_n; \phi) \\ &= f_\theta(\hat{x}_{t_n}^\phi, t_n) - f_\theta(x_{t_n}, t_n) + f_\theta(x_{t_n}, t_n) - f(x_{t_n}, t_n; \phi) \\ &= f_\theta(\hat{x}_{t_n}^\phi, t_n) - f_\theta(x_{t_n}, t_n) + e_n, \end{aligned} \quad (15)$$

where (i) is due to Eq. (14) and $f(x_{t_{n+m}}, t_{n+m}; \phi) = f(x_{t_n}, t_n; \phi)$. Considering $f_\theta(\cdot, t_n)$ has Lipschitz constant L , we have:

$$\|e_{n+m}\|_2 \leq \|e_n\|_2 + L\|\hat{x}_{t_n}^\phi - x_{t_n}\|_2 \quad (16)$$

$$\stackrel{(i)}{=} \|e_n\|_2 + L \cdot O\left(\max_{k \in \llbracket n, n+m-1 \rrbracket} (t_{k+1} - t_k)^{p+1}\right) \quad (17)$$

$$= \|e_n\|_2 + O\left(\max_{k \in \llbracket n, n+m-1 \rrbracket} (t_{k+1} - t_k)^{p+1}\right). \quad (18)$$

Considering the definition of f , we have:

$$e_0 = f_\theta(x_{t_0}, t_0) - f(x_{t_0}, t_0; \phi) \quad (19)$$

$$\stackrel{(ii)}{=} x_{t_0} - x_{t_0} \quad (20)$$

$$= \mathbf{0}. \quad (21)$$

Let $j * m == N$, we have:

$$\|e_{m*j}\|_2 \leq \|e_0\|_2 + \sum_{k=0}^{j-1} O\left(\max_{l \in \llbracket k*m, (k+1)*m-1 \rrbracket} (t_{l+1} - t_l)^{p+1}\right) \quad (22)$$

$$= \sum_{k=0}^{j-1} O\left(\max_{l \in \llbracket k*m, (k+1)*m-1 \rrbracket} (t_{l+1} - t_l)^{p+1}\right) \quad (23)$$

$$= \sum_{k=0}^{j-1} \left(\max_{l \in \llbracket k*m, (k+1)*m-1 \rrbracket} (t_{l+1} - t_l) \right) O\left(\max_{l \in \llbracket k*m, (k+1)*m-1 \rrbracket} (t_{l+1} - t_l)^p\right) \quad (24)$$

$$\leq \sum_{k=1}^{j-1} (T - \epsilon) O\left(\max_{l \in \llbracket k*m, (k+1)*m-1 \rrbracket} (t_{l+1} - t_l)^p\right) \quad (25)$$

$$\leq \sum_{k=1}^{j-1} (T - \epsilon) O((\Delta t)^p) \quad (26)$$

$$= O((\Delta t)^p) \quad (27)$$

which completes the proof. Eq. 24 and Eq. 25 demonstrate that our method has a smaller error upper bound. \square

Table 3. MSE Loss of Feature Extracted by DINOv2 During LGP and ACD Stages.

Stage	MSE(DINOv2(x_{in}^{Image}), DINOv2(x^{Predict}))	MSE(DINOv2($f_{\theta}(x_{t_{n+m}}, t_{n+m})$), DINOv2(x^{Predict}))
LGP	0.21	0.26
ACD	0.0022	4.09e-5

B. Discussion

B.1. Discussion on LGP and ACD

We demonstrate the convergence of training at different stages based on PCM [32]. Let the data distribution used in the LGP and ACD phases be denoted as p_0 , and the forward conditional probability path is defined as $\alpha_t \mathbf{x}_0 + \sigma_t \epsilon$. The intermediate distribution is then defined as $p_t(\mathbf{x}) = (p_0(\frac{\mathbf{x}}{\alpha_t}) \cdot \frac{1}{\alpha_t}) * \mathcal{N}(0, \sigma_t)$. Similarly, the data distribution used for pretraining the diffusion model is denoted as $p_0^{\text{pretrain}}(\mathbf{x})$, and the corresponding intermediate distribution during the forward process is $p_t^{\text{pretrain}}(\mathbf{x}) = (p_0^{\text{pretrain}}(\frac{\mathbf{x}}{\alpha_t}) \cdot \frac{1}{\alpha_t}) * \mathcal{N}(0, \sigma_t)$. This is reasonable because current large diffusion models are typically trained with more resources on larger datasets compared to those used for consistency distillation. We denote $\mathcal{T}_{t \rightarrow s}^{\phi}$, $\mathcal{T}_{t \rightarrow s}^{\theta}$, and $\mathcal{T}_{t \rightarrow s}^{\phi'}$ as the flow operators corresponding to the pre-trained diffusion model, the flow operators corresponding to our consistency model, and the PF-ODE of the data distribution used for consistency distillation, respectively.

We first discuss the convergence of $\mathcal{L}_{\text{ACD}}^{\text{adv}}$. We have $f_{\theta}(x_{t_{n+m}}, t_{n+m}) \equiv f_{\theta}(\hat{x}_{t_n}^{\phi}, t_n)$, where $x_{t_{n+m}} \in p_{n+m}$ and $x_{t_n} \in p_n$. Consequently, we obtain:

$$\mathcal{T}_{t_{n+m} \rightarrow \epsilon}^{\theta} \# \mathbb{P}_{t_{n+m}} \equiv \mathcal{T}_{t_n \rightarrow \epsilon}^{\theta} \mathcal{T}_{t_{n+m} \rightarrow t_n}^{\phi} \# \mathbb{P}_{t_{n+m}}. \quad (28)$$

Therefore, if $\text{Dis}(f_{\theta}(x_{t_{n+m}}, t_{n+m}), f_{\theta}(\hat{x}_{t_n}^{\phi}, t_n)) = 0$, we have $\mathcal{L}_{\text{ACD}}^{\text{adv}} = 0$.

We discuss the convergence of $\mathcal{L}_{\text{LGP}}^{\text{adv}}$. We have:

$$p_0 \equiv \mathcal{T}_{t_{n+m} \rightarrow 0}^{\phi'} \# p_{t_{n+m}}. \quad (29)$$

Therefore, we have

$$\text{Dis}(\mathcal{T}_{t_{n+m} \rightarrow \epsilon}^{\theta} \# p_{t_{n+m}} \| p_0) \quad (30)$$

$$= \text{Dis}(\mathcal{T}_{t_{n+m} \rightarrow \epsilon}^{\theta} \# p_{t_{n+m}} \| \mathcal{T}_{t_{n+m} \rightarrow 0}^{\phi'} \# p_{t_{n+m}}) \quad (31)$$

Because $f_{\theta}(x_{t_{n+m}}, t_{n+m}) \equiv f_{\theta}(\hat{x}_{t_n}^{\phi}, t_n)$, we have:

$$\text{Dis}(\mathcal{T}_{t_{n+m} \rightarrow \epsilon}^{\theta} \# p_{t_{n+m}} \| \mathcal{T}_{t_{n+m} \rightarrow 0}^{\phi'} \# p_{t_{n+m}}) \quad (32)$$

$$= \text{Dis}(\mathcal{T}_{t_{n+m} \rightarrow \epsilon}^{\phi} \# p_{t_{n+m}} \| \mathcal{T}_{t_{n+m} \rightarrow 0}^{\phi'} \# p_{t_{n+m}}) \quad (33)$$

$$= \text{Dis}(p_0^{\text{pretrain}} \| p_0) \quad (34)$$

Because $p_0^{\text{pretrain}} \neq p_0$, we have $\mathcal{L}_{\text{LGP}}^{\text{adv}} > 0$.

We consider the input condition x_{in}^{Image} for the diffusion model, which involves replicating the image condition across multiple frames to align with the frame count of the original video. The output of our consistency model is $f_{\theta}(x_{t_{n+m}}, t_{n+m})$. During the LGP phase, our prediction target is $x^{\text{Predict}} = x_0$. During the ACD phase, our prediction target is $x^{\text{Predict}} = f_{\theta}(\hat{x}_{t_n}^{\phi}, t_n)$.

We extract features from these data using DINOv2 and compute the MSE loss of these features. As shown in Table 3, during the LGP phase, the difference between x_{in}^{Image} and x^{Predict} is minimal, indicating that our consistency model tends to predict multiple static images. During the ACD phase, the difference between $f_{\theta}(x_{t_{n+m}}, t_{n+m})$ and x^{Predict} is minimal,

indicating that our consistency model tends to predict data generated by the pre-trained model. Although random noise is added to x_{in}^{Image} in actual training, this does not fundamentally solve the issue. However, fortunately, using LGP in the early stage of model training can accelerate the convergence of our distillation model. Figure 1 demonstrates the effectiveness of using LGP initially.

B.2. Contributions

Here, we re-emphasize the key components of OSV and summarize the contributions of our work.

The primary motivation of this research is to expedite the sampling process for high-resolution image-to-video generation by leveraging the consistency model training paradigm. Previous methods, including Animate-LCM and SF-V, sought to harness the potential of consistency models in this demanding scenario but failed to deliver satisfactory outcomes. We systematically examine and dissect the limitations of these approaches from three distinct perspectives. Crucially, these methods largely represent direct extensions of techniques originally devised to accelerate text-to-image sampling, and their straightforward adaptation to image-to-video sampling introduces significant challenges. To address these issues, we broaden the design space and propose comprehensive solutions to overcome these limitations.

The OSV framework is built upon the decomposition of the training process into two distinct stages, each utilizing a tailored distillation method to ensure efficient and effective model training. In the second stage, we introduce a multi-step solving method that capitalizes on the teacher model to execute multiple reverse ODE processes, thereby enhancing prediction accuracy. As illustrated in Figure 5, this multi-step solving method not only accelerates training but also significantly improves the performance of the consistency model.

Furthermore, inspired by the inherent properties of consistency models, we propose a novel higher-order solver, termed TTS, to replace the conventional CFG method. Experimental evaluations substantiate the efficacy of TTS, with results demonstrating state-of-the-art image-to-video generation performance. Remarkably, our approach achieves this using only 8 H800 GPUs (with merely 2 H800 GPUs required in the second stage), underscoring the efficiency and effectiveness of the proposed method.

B.3. Removing CFG

We introduce CFG into the distilled model: $\hat{\Phi}(\mathbf{x}_{t_{n+m}}, t_{n+m}, c; \phi) = \Phi(\mathbf{x}_{t_{n+m}}, t_{n+m}, c_{zero}; \phi) + w * (\Phi(\mathbf{x}_{t_{n+m}}, t_{n+m}, c; \phi) - \Phi(\mathbf{x}_{t_{n+m}}, t_{n+m}, c_{zero}; \phi))$. This means the model already has CFG during inference, and using the same CFG scale again during inference leads to exposure issues in the generated videos. Table 2f also shows that a smaller CFG scale does not significantly improve the video quality. Removing CFG not only speeds up the model generation but also improves the overall quality of the generated videos.

C. Additional Experimental Settings

λ^{LGP} and λ^{ACD} are set to 0.1. In the Huber Loss, we set $c = 0.001$.

We train the model with videos of 14 frames, and the test videos also consist of 14 frames.

We use TTS only when the step equals 1.

Table 4. Effect of Upsampling Module.

Upsampling	FVD↓
✓	171.15
×	194.83

D. Upsampling Module

As shown in Figure 9, the upsampling module is displayed. First, we increase the number of channels of the latent space features, and then upsample the latent space features using the PixelShuffle operation. We set $r = 4$.

As shown in Table 4, using the upsampling module helps reduce the information loss in videos after they pass through the VAE Encoder.

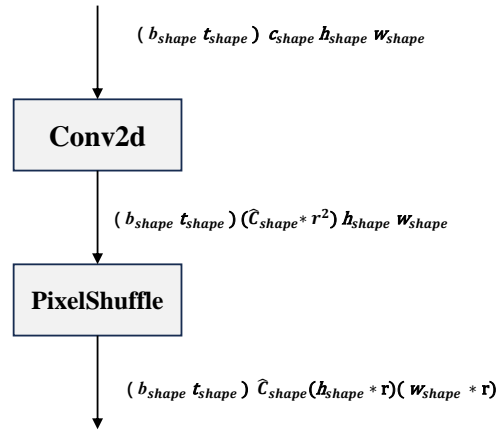


Figure 9. Upsampling Module. We design the upsampling module inspired by sub-pixel convolution [25].