Transition Models: Rethinking the Generative Learning Objective

Zidong Wang[1,2,*], Yiyuan Zhang[1,2,*,‡], Xiaoyu Yue[2,3], Xiangyu Yue[1,†],
Yangguang Li[1,†], Wanli Ouyang[1,2], Lei Bai[2,†]
[1]MMLab, CUHK [2]Shanghai AI Lab [3]USYD
{wangzd2022, yiyuanzhang.ai}@gmail.com, {xyyue, wlouyang}@ie.cuhk.edu.hk
Code: https://github.com/WZDTHU/TiM

Abstract

A fundamental dilemma in generative modeling persists: iterative diffusion models achieve outstanding fidelity, but at a significant computational cost, while efficient few-step alternatives are constrained by a hard quality ceiling. This conflict between generation steps and output quality arises from restrictive training objectives that focus exclusively on either infinitesimal dynamics (PF-ODEs) or direct endpoint prediction. We address this challenge by introducing an exact, continuous-time dynamics equation that analytically defines state transitions across any finite time interval Δt. This leads to a novel generative paradigm, Transition Models (TiM), which adapt to arbitrary-step transitions, seamlessly traversing the generative trajectory from single leaps to fine-grained refinement with more steps. Despite having only 865M parameters, TiM achieves state-of-the-art performance, surpassing leading models such as SD3.5 (8B parameters) and FLUX.1 (12B parameters) across all evaluated step counts. Importantly, unlike previous few-step generators, TiM demonstrates monotonic quality improvement as the sampling budget increases. Additionally, when employing our native-resolution strategy, TiM delivers exceptional fidelity at resolutions up to 4096×4096.

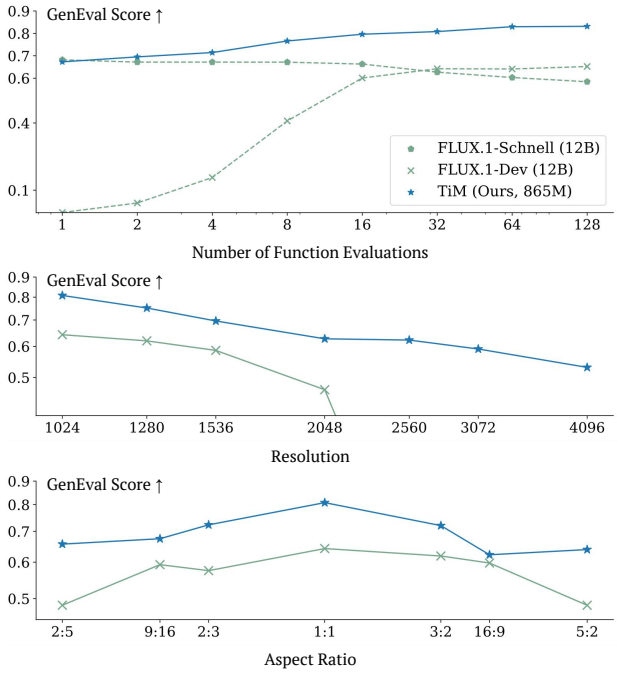


Figure 1. **TiM's superior performance across different NFEs, resolutions, and aspect ratios.** On the GenEval [27] benchmark, TiM outperforms Flux.1 models [5, 6] at different NFEs (top, 1024×1024), at higher resolutions (middle, 1024×1024 to 4096×4096), and diverse aspect ratios (bottom, $2:5$ to $5:2$).

1. Introduction

Diffusion models have emerged as the dominant paradigm in visual content generation, producing state-of-the-art results in various domains [9, 19, 37, 54, 56, 83]. They generate samples from noise by iterative denoising, a process that can be formulated as numerical integration of either the reverse-time Stochastic Differential Equation (SDE) or the corresponding Probability-Flow Ordinary Differential Equation (PF-ODE), with related discrete-time solvers [48, 68, 71]. Despite its effectiveness, iterative denoising entails a large Number of Function Evaluations (NFEs), approximately proportional to the number of integration steps– leading to increased inference latency and compute cost.

In contrast, recent approaches reduce step counts by avoiding explicit multi-step integration. Consistency models [47, 69, 72] impose PF-ODE self-consistency across different noise levels, while distribution-distillation methods [44, 63, 64, 84, 91] train students to approximate teacher distributions with fewer denoising steps. Shortcut [20], FlowMap [7, 59], and MeanFlow [26, 55] learn the average velocity (*i.e.*, shortcut) along the flow-matching trajectory via a self-consistency objective. The principle is that a single large step should approximate the integral of multiple smaller instantaneous steps. However, by averaging the entire trajectory on the linear transport, *they irre-*

*: Equal contribution. ‡: Project lead. †: Corresponding authors: Lei Bai, Yangguang Li, and Xiangyu Yue.

vocably discard the fine-grained local dynamics necessary for high-fidelity refinement. This leads to performance saturation: while effective for few-step generation, it offers no gains from additional sampling budget. Moreover, despite strong few-step results, their compute–quality scaling is weaker than that of high-NFE diffusion models: *quality gains plateau after only a few steps, and asymptotic performance remains below traditional multi-step diffusion*.

Thus, the entire field converges on a fundamental, yet flawed, compromise [26, 32, 46, 72, 84]: models either achieve high fidelity at the cost of computational efficiency (*e.g.*, diffusion models), or they gain efficiency by sacrificing the very dynamics needed for high-fidelity refinement (*e.g.*, few-step models). The root of this dilemma is not architectural, but a learning objective. It stems from a foundational choice in the way these models are taught to generate. This trade-off is a direct and inevitable consequence of the chosen *granularity of supervision*. On the one hand, local supervision methods that model instantaneous dynamics (such as those consistent with PF-ODE [32, 46, 71]) achieve high accuracy with small step sizes (Δt) and scale well to many-step generation. However, their performance significantly degrades in the few-step regimes. On the other hand, finite-horizon supervision, which models direct mapping over a fixed interval (such as consistency models [47, 69, 72]), excels at generating in a few-steps. Yet, these models see diminishing returns from additional intermediate steps unless specifically trained with complex, multi-interval objectives. This reveals a persistent dilemma: *objectives that model instantaneous dynamics and those that learn finite-interval mappings, each entail inherent limitations*. This motivates the question: **What is an appropriate learning objective for generative models?**

We attempt to answer this question from the following perspectives:

1) Diffusion training learns a local PF-ODE field whose numerical integration is accurate only in the small-step limit $\Delta t \to 0$. As illustrated in Fig. 2 (a), with large steps, the discretization error dominates; therefore, *the objective should be flexible in terms of step sizes*.

2) Few-step objectives supervise an endpoint map but do not learn a compositional flow: without an approximate semigroup over time, extra steps induce accumulated errors rather than refine the generated samples, as in Fig. 2 (b), causing schedule sensitivity and early saturation. Therefore, *the objective requires consistency along the entire trajectory, where intermediate steps act as refinements along a single trajectory rather than as deviations onto new ones*.

Consequently, we argue that a generative model should learn a *versatile denoising operator*, parameterized by the desired interval Δt. By learning the transitions between any state $\mathbf{x}_t$ to a previous state $\mathbf{x}_{t-\Delta t}$ for an *arbitrary* Δt, *the generative models are learning the solution manifold of the generative process* [1] *itself*. This approach is fundamentally distinct from approximating a differential equation or a statistical mapping. Inherently, it unifies local and finite-horizon supervisions, yielding a model that is both a powerful few-step generator and an accurate and refinable integrator. Since the training objective is to learn the transitions between any state to a previous state, it is named Transition Models (TiM), which parameterize state-to-state transitions along the PF-ODE trajectory for arbitrary time intervals.

We validate TiM's effectiveness through extensive experiments on text-to-image and class-guided image generation. As shown in Figure 1, TiM shows superior performance across different NFEs, resolutions, and aspect ratios. On the GenEval [27] benchmark, our compact 865M parameter model, TiM, establishes a new state-of-the-art. It achieves a score of 0.67 with a single function evaluation (1-NFE) and scales to 0.83 at 128-NFE, outperforming billion-scale industrial models including SD3.5-Large [19] (8B).

2. Related Work

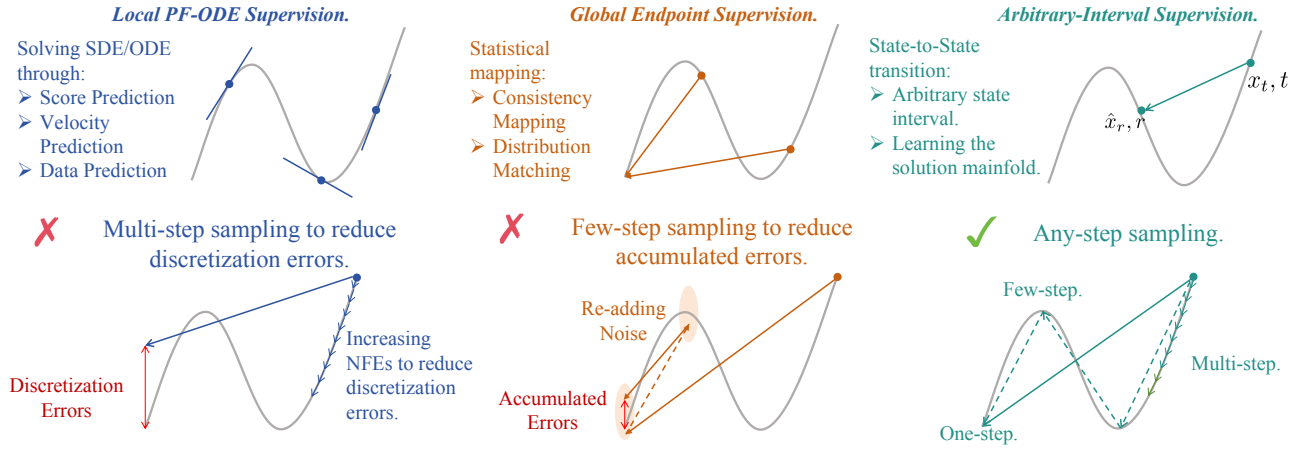
Diffusion and Consistency Models. Continuous generative modeling has seen two dominant paradigms. Diffusion models [32, 37, 46] iteratively solve a PF-ODE/SDE, achieving high quality but requiring many function evaluations (NFEs). In contrast, Consistency Models [72] learn a direct mapping for few-step generation, but suffer from performance saturation and complex training requirements (e.g., pre-training and stabilization [25, 47, 69]). Although recent methods such as FlowMap [7] and MeanFlow [26] enable training CM-like models from scratch, they inherit the same limitation of stagnant quality with more steps.

To break this impasse, we propose a new learning principle: mastering state transitions over arbitrary time intervals. This enables the model to function as a robust navigator on the data manifold, *preserving a few-step efficiency while supporting monotonic refinement by using more steps*.

Text-to-Image Generation with Few-steps. Efficient text-to-image (T2I) sampling is currently dominated by distillation. These methods fall into two main camps: distribution distillation (e.g. SD-Turbo [63, 64], DMD [84, 85]), which matches the teacher's output distribution, and trajectory distillation (e.g., LCM [50], PCM [78]), which mimics the generation path. Hybrid methods [57] combine both.

All these methods suffer from inherent flaws: 1) reliance on large, pre-trained teachers, resulting in costly pipelines; and 2) brittle performance that stagnates or degrades as sampling steps increase. We bypass these by introducing TiM, **the first T2I generator** trained from scratch that masters arbitrary-step sampling, *delivering strong few-step results that monotonically improve with more computation*.

[1] solution manifold of a generative process is the high-dimensional geometric surface formed by the collection of all possible generative trajectories that lead from noise to data.



(a) Diffusion Models: $f_\theta(x_t, t)$ (b) Few-step Models: $f_\theta(x_t, t)$ (c) Transition Models: $f_\theta(x_t, t, r)$

Figure 2. **Illustration of Different Generative Paradigms**. While conventional diffusion models learn the local PF-ODE field and few-step models learn a fixed endpoint map (a single large step), our Transition Models (TiM) are trained to master arbitrary state-to-state transitions. This approach allows TiM to learn the entire solution manifold of the generative process, unifying the one-step, few-step and many-step regimes within a single, powerful model.

3. Transition Models

In this section, we first analyze the limitations of PF-ODE supervision in diffusion models, which constrain the state transition to a local, infinitesimal interval. To address the limitations, we generalize diffusion's local state transition to an arbitrary-interval state transition, as illustrated in Fig. 2, from which we derive a novel mathematical identity that links the state $\mathbf{x}_t$, the interval Δt, and the network $\boldsymbol{f}_\theta$. From this identity, we formulate a training objective that governs the state transition over any time interval: for randomly sampled pairs of $(t, \Delta t)$, we train $\boldsymbol{f}_\theta$ to predict the target state $\mathbf{x}_{t-\Delta t}$ (or an equivalent representation) of $\mathbf{x}_t$. We further propose two theoretically motivated improvements for stable and scalable training.

3.1. Limitation of PF-ODE Supervision

Given the noise from the Gaussian distribution $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and the clean data from the data distribution $\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})$, the diffusion models learn to map the noise distribution to the data distribution. Given the time range $t \in [0, T]$, the forward process utilizes the coefficients α_t and σ_t, such that $\boldsymbol{x}_t = \alpha_t \boldsymbol{x} + \sigma_t \boldsymbol{\varepsilon}$. Song et al. [71] has proven that the forward process can be described by a Stochastic Differential Equation (SDE):

$$\mathrm{d}\boldsymbol{x}_t = \mathrm{f}(\boldsymbol{x}_t, t)\mathrm{d}t + \mathrm{g}(t)\mathrm{d}\mathbf{w}, \tag{1}$$

where $\mathbf{w}$ is the standard Wiener process, $\mathrm{f}(\boldsymbol{x}_t, t) = \frac{\dot{\alpha}_t}{\alpha_t}\boldsymbol{x}_t$ is the drift coefficient, and $\mathrm{g}(t) = 2\sigma_t\dot{\sigma}_t - 2\frac{\dot{\alpha}_t}{\alpha_t}\sigma_t^2$ is the diffusion coefficient [37, 47, 71, 74]. Anderson [3] and Song et al. [71] have shown that the forward process can be reversed by solving the reverse-time SDE from or equivalently the probability flow ODE (PF-ODE)[2]:

$$\begin{aligned}\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} &= \mathrm{f}(\boldsymbol{x}_t, t) - \frac{1}{2}\mathrm{g}(t)^2\nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{x}_t) \\ &= \frac{\mathrm{d}\alpha_t}{\mathrm{d}t}\boldsymbol{x} + \frac{\mathrm{d}\sigma_t}{\mathrm{d}t}\boldsymbol{\varepsilon},\end{aligned} \tag{2}$$

where $\nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{x}_t) = -\frac{\boldsymbol{\varepsilon}}{\sigma_t}$ denotes the score function.

Thus, a diffusion model can be parameterized as $\boldsymbol{f}_\theta(\boldsymbol{x}_t, t) = \boldsymbol{F}_\theta(\boldsymbol{x}_t, c_{\text{noise}}(t))$, where θ denotes the parameters of the neural network and $c_{\text{noise}}(t)$ is the time scaling function. The training objective can be given by:

$$\mathbb{E}_{\boldsymbol{x},\boldsymbol{\varepsilon},t}[w(t)d\left(\boldsymbol{f}_\theta(\boldsymbol{x}_t, t) - (\hat{\alpha}_t\boldsymbol{x} + \hat{\sigma}_t\boldsymbol{\varepsilon})\right)], \tag{3}$$

where $\hat{\alpha}_t$ and $\hat{\sigma}_t$ are the diffusion target coefficients, $w(t)$ is a weighting function, $d(\cdot,\cdot)$ is a metric function such as L2 loss $d(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|_2^2$.

Despite different transports [3] have instantiate target coefficients $\hat{\alpha}_t$ and $\hat{\sigma}_t$, the training objectives are equivalent to supervising the PF-ODE field [4]. During sampling, diffusion models solve this PF-ODE, integrated from $t = T$ to $t = 0$ using numerical solvers. To reduce discretization errors and preserve the learned continuous-time dynamics, practical solvers [49, 68] typically require a small step size (i.e., $\Delta t \to 0$) or many sub-steps per interval (i.e., high-order solvers), thus inducing huge NFEs.

[2] Song et al. [71] have shown that the PF-ODE trajectory has the same marginal probability as the reverse-time SDE: $\mathrm{d}\boldsymbol{x}_t = [\mathrm{f}(\boldsymbol{x}_t, t) - \frac{1}{2}\mathrm{g}(t)^2\nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{x}_t)]\mathrm{d}t + \mathrm{g}(t)\mathrm{d}\mathbf{w}$.

[3] For convenience, we elaborate the coefficients α_t, σ_t, $\hat{\alpha}_t$, and $\hat{\sigma}_t$ of different diffusion transports in Tab. 1

[4] For example, in VE-SDE [71],with coefficients $\alpha_t = 1, \sigma_t = t$, the PF-ODE is: $\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} = \boldsymbol{\varepsilon}$, and the training objective is $-\boldsymbol{\varepsilon}$. In OT-FM [45], with $\alpha_t = 1 - t, \sigma_t = t$, the PF-ODE is: $\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} = \boldsymbol{\varepsilon} - \boldsymbol{x}$, which directly matches the training objective.

Transport	*Diffusion Parameterization*					*Transition Parameterization*			
	$c_{\text{noise}}(t) =$	$\alpha_t =$	$\sigma_t =$	$\hat{\alpha}_t =$	$\hat{\sigma}_t =$	$\frac{\mathrm{d}\hat{\alpha}_t}{\mathrm{d}t} =$	$\frac{\mathrm{d}\hat{\sigma}_t}{\mathrm{d}t} =$	$B_{t,r} =$	$\frac{\mathrm{d}B_{t,r}}{\mathrm{d}t} =$
OT-FM [45]	t	$1-t$	t	-1	1	0	0	$r-t$	-1
TrigFlow [47]	t	$\cos(t)$	$\sin(t)$	$-\sin(t)$	$\cos(t)$	$-\cos(t)$	$-\sin(t)$	$\sin(r-t)$	$-\cos(r-t)$
EDM [37]	$\frac{1}{4}\ln(t)$	$\frac{1}{t^2+\sigma_{\mathrm{d}}^2}$	$\frac{t}{\sqrt{t^2+\sigma_{\mathrm{d}}^2}}$	$\frac{t}{\sigma_{\mathrm{d}}\sqrt{t^2+\sigma_{\mathrm{d}}^2}}$	$-\frac{\sigma_{\mathrm{d}}}{\sqrt{t^2+\sigma_{\mathrm{d}}^2}}$	Eq. (33)	Eq. (34)	Eq. (35)	Eq. (36)
VP-SDE [32]	$(T-1)t$	$\frac{1}{\beta_t^2+1}$	$\frac{\beta_t}{\sqrt{\beta_t^2+1}}$	0	1	0	0	$\frac{\beta_r-\beta_t}{\sqrt{\beta_r^2+1}}$	$\frac{-1}{\sqrt{\beta_r^2+1}}\cdot\frac{\mathrm{d}\beta_t}{\mathrm{d}t}$
VE-SDE [71]	$\ln(\frac{1}{2}t)$	1	t	0	-1	0	0	$t-r$	1
TiM (Ours)	t	$\cos(\frac{\pi}{2}t)$	$\sin(\frac{\pi}{2}t)$	-1	1	0	0	$\frac{\sin(\frac{\pi}{2}(r-t))}{\sin(\frac{\pi}{2}t)+\cos(\frac{\pi}{2})t}$	$-\frac{\sqrt{2}\sin(\frac{\pi}{2}r+\frac{\pi}{4})}{2\sin(\pi t)+2}$

Table 1. **Transition parameterization for different diffusion transports**. For VP-SDE, T is set to 1000, and $\beta_t = \sqrt{e^{\frac{1}{2}\beta_{\mathrm{d}}t^2+\beta_{\min}t}-1}$, where $\beta_{\mathrm{d}} = 19.9$ and $\beta_{\min} = 0.1$ by default. Song et al. [71] proves that VP-SDE equals DDPM [32] while VE-SDE equals score matching [70]. The EDM transition parameterization is too complex, so we provide them in Eqs. (33) to (36) in the Appendix.

3.2. State Transition

The derivation begins with the general mathematical form for a state transition between points $(\boldsymbol{x}_t, \boldsymbol{x}_r)$ on a PF-ODE trajectory, as given in Eq. (6). *The central principle is to treat this form not as a numerical approximation but as an exact identity that must hold for any interval* $\Delta t = t - r$. Therefore, it allows us to formulate a *general state transition dynamic* (Eq. (8)) that is valid across any interval. Consequently, the model's training objective is no longer constrained to approximating a local solution of the PF-ODE. Instead, it is trained to learn the *entire solution manifold of the generative process*. By internalizing this global structure, the model inherently acquires the ability to perform inference over arbitrary step sizes, from large, single leaps to fine-grained, iterative refinement. We illustrate our derivation process step-by-step as follows.

State Transition Approximation. Given $\boldsymbol{x}_t = \alpha_t\boldsymbol{x} + \sigma_t\boldsymbol{\varepsilon}$, a diffusion model $\boldsymbol{f}_\theta(\boldsymbol{x}_t, t)$ is optimized towards the target $\hat{\alpha}_t\boldsymbol{x} + \hat{\sigma}_t\boldsymbol{\varepsilon}$, leading to the x-prediction and $\boldsymbol{\varepsilon}$-prediction:

$$\hat{x} = \frac{\hat{\sigma}_t\boldsymbol{x}_t - \sigma_t\boldsymbol{f}_\theta(\boldsymbol{x}_t, t)}{\hat{\sigma}_t\alpha_t - \hat{\alpha}_t\sigma_t}, \quad \hat{\boldsymbol{\varepsilon}} = \frac{\alpha_t\boldsymbol{f}_\theta(\boldsymbol{x}_t, t) - \hat{\alpha}_t\boldsymbol{x}_t}{\hat{\sigma}_t\alpha_t - \hat{\alpha}_t\sigma_t}. \tag{4}$$

Using the prediction $\hat{x}$ and $\hat{\boldsymbol{\varepsilon}}$, the arbitrary previous state $\boldsymbol{x}_r$ $(r < t)$ can be approximated as:

$$\begin{aligned}\boldsymbol{x}_r &= \alpha_r\hat{\boldsymbol{x}} + \sigma_r\hat{\boldsymbol{\varepsilon}} \\ &= \frac{(\alpha_r\hat{\sigma}_t - \sigma_r\hat{\alpha}_t)\boldsymbol{x}_t + (\sigma_r\alpha_t - \alpha_r\sigma_t)\boldsymbol{f}_\theta(\boldsymbol{x}_t, t)}{\hat{\sigma}_t\alpha_t - \hat{\alpha}_t\sigma_t}.\end{aligned} \tag{5}$$

This represents the general form of a first-order state transition approximation on the PF-ODE Trajectory.

State Transition Identity. In contrast to diffusion models, which approximate Eq. (5) in the limit as $\Delta t \to$, our transition models learn an exact state transition function $\boldsymbol{f}_\theta(\boldsymbol{x}_t, t, r) = \boldsymbol{F}_\theta(\boldsymbol{x}_t, c_{\text{noise}}(t), c_{\text{noise}}(r))$ between any two states $\boldsymbol{x}_t$ and $\boldsymbol{x}_r$. Introducing $\boldsymbol{f}_\theta(\boldsymbol{x}_t, t, r)$ to Eq. (5), we obtain the following equation:

$$\boldsymbol{x}_r = \frac{\alpha_r\hat{\sigma}_t - \sigma_r\hat{\alpha}_t}{\hat{\sigma}_t\alpha_t - \hat{\alpha}_t\sigma_t}\boldsymbol{x}_t + \frac{\sigma_r\alpha_t - \alpha_r\sigma_t}{\hat{\sigma}_t\alpha_t - \hat{\alpha}_t\sigma_t}\boldsymbol{f}_\theta(\boldsymbol{x}_t, t, r). \tag{6}$$

Here, we define $A_{t,r} := \frac{\alpha_r\hat{\sigma}_t - \sigma_r\hat{\alpha}_t}{\hat{\sigma}_t\alpha_t - \hat{\alpha}_t\sigma_t}$, $B_{t,r} := \frac{\sigma_r\alpha_t - \alpha_r\sigma_t}{\hat{\sigma}_t\alpha_t - \hat{\alpha}_t\sigma_t}$, and $\boldsymbol{f}_{\theta,t,r} := \boldsymbol{f}_\theta(\boldsymbol{x}_t, t, r)$ for simplicity. Our target is to make Eq. (6) hold for any Δt, so the case is: given r, $\boldsymbol{f}_{\theta,t,r}$ can transit any state $\boldsymbol{x}_t$ to the same state $\boldsymbol{x}_r$. Considering that r is independent of t, we differentiate both sides with respect to t and rearranging the equation, which yields:

$$\begin{aligned}&\frac{\mathrm{d}\boldsymbol{x}_r}{\mathrm{d}t} = \frac{\mathrm{d}}{\mathrm{d}t}(A_{t,r}\boldsymbol{x}_t + B_{t,r}\boldsymbol{f}_{\theta,t,r}) \Longrightarrow \\ &\boldsymbol{x}_t\frac{\mathrm{d}A_{t,r}}{\mathrm{d}t} + A_{t,r}\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} = -\boldsymbol{f}_{\theta,t,r}\frac{\mathrm{d}B_{t,r}}{\mathrm{d}t} - B_{t,r}\frac{\mathrm{d}\boldsymbol{f}_{\theta,t,r}}{\mathrm{d}t},\end{aligned} \tag{7}$$

which can be further simplified as follows[5]:

$$\begin{aligned}&\frac{\mathrm{d}(B_{t,r}\cdot(\hat{\alpha}_t\boldsymbol{x} + \hat{\sigma}_t\boldsymbol{\varepsilon} - \boldsymbol{f}_{\theta,t,r}))}{\mathrm{d}t} = 0 \Longrightarrow \\ &\underbrace{(\hat{\alpha}_t\boldsymbol{x} + \hat{\sigma}_t\boldsymbol{\varepsilon} - \boldsymbol{f}_{\theta,t,r})}_{\text{PF-ODE supervision}}\frac{\mathrm{d}B_{t,r}}{\mathrm{d}t} + B_{t,r}\underbrace{\frac{\mathrm{d}(\hat{\alpha}_t\boldsymbol{x} + \hat{\sigma}_t\boldsymbol{\varepsilon} - \boldsymbol{f}_{\theta,t,r})}{\mathrm{d}t}}_{\text{time-slope matching}} = 0.\end{aligned} \tag{8}$$

We denote Equation (8) as the State Transition Identity, a product-derivative invariant. The State Transition Identity, $\frac{\mathrm{d}}{\mathrm{d}t}(B_{t,r}\cdot h(t)) = 0$, where $h(t) = \hat{\alpha}_t\boldsymbol{x} + \hat{\sigma}_t\boldsymbol{\varepsilon} - \boldsymbol{f}_{\theta,t,r}$ is the instantaneous residual, imposes a powerful three-fold constraint on the generative model $\boldsymbol{f}_\theta$.

- **Implicit Trajectory Consistency:** Identity dictates that the weighted residual $B_{t,r}h(t)$ must be constant for any starting time t satisfying $t > r$, leading to the same target $\boldsymbol{x}_r$. This directly enforces trajectory consistency: the direct map $(t \to r)$ must be equivalent to any composition of intermediate steps, such as $(t \to s) \circ (s \to r)$. This property (Eq. (8)), absent in standard consistency models, is the core mechanism that makes TiM robust to sampling schedules and enables monotonic refinement.
- **Time-Slope Matching:** Unpacking the product rule reveals that $(\frac{\mathrm{d}}{\mathrm{d}t}B_{t,r})h(t) + B_{t,r}(\frac{\mathrm{d}}{\mathrm{d}t}h(t)) = 0$. Unlike conventional diffusion training, which only minimizes the

[5] As r is independent of t, we have $\frac{\mathrm{d}\boldsymbol{x}_r}{\mathrm{d}t} = 0$. The complete derivation is detailed in the Appendix A.1.

residual's value ($h(t) \to 0$), our objective incorporates a time-slope term $\frac{\mathrm{d}}{\mathrm{d}t}h(t)$ and minimizes the joint term. This higher-order supervision compels the model to learn a smoother solution manifold, preserving coherence during large-step sampling and ensuring stable refinement with smaller steps.

- **Boundary Condition:** When $r \to t$, the transition weight vanishes: $B_{t,r} \to 0$. the transition degenerates to the PF-ODE supervision case, i.e., $\lim_{t\to r} \boldsymbol{f}_{\theta,t,r} \to \hat{\alpha}_t \boldsymbol{x} + \hat{\sigma}_t \boldsymbol{\varepsilon}$. This boundary case ensures that TiM remains fully compatible with conventional PF-ODE objectives while naturally extending them to finite time intervals. Consequently, the proposed formulation bridges infinitesimal supervision and large-step consistency under a unified identity constraint.

Derived from State Transition Identity (Eq. (8)), we obtain the learning target $\hat{\boldsymbol{f}}$:

$$\hat{\boldsymbol{f}} = \hat{\alpha}_t \boldsymbol{x} + \hat{\sigma}_t \boldsymbol{\varepsilon} + \frac{B_{t,r}}{\frac{\mathrm{d}B_{t,r}}{\mathrm{d}t}} \left(\frac{\mathrm{d}\hat{\alpha}_t}{\mathrm{d}t} \boldsymbol{x} + \frac{\mathrm{d}\hat{\sigma}_t}{\mathrm{d}t} \boldsymbol{\varepsilon} - \frac{\mathrm{d}\boldsymbol{f}_{\theta^-,t,r}}{\mathrm{d}t} \right), \quad (9)$$

where θ^- is the fixed network parameter θ without gradient and $\frac{\mathrm{d}\boldsymbol{f}_{\theta^-,t,r}}{\mathrm{d}t}$ is the time derivative of the network.

Transition Parameterization. This derivation culminates in a universal parameterization framework. Its universality is directly derived from the State Transition Identity (Eq. (8)). To ensure that this identity is theoretically valid and leads to a well-defined learning target (Eq. (9)), we assume that: **1)** the transport coefficients $(\alpha_t, \sigma_t, \hat{\alpha}_t, \hat{\sigma}_t)$ are continuously differentiable, i.e., $\alpha_t, \sigma_t, \hat{\alpha}_t, \hat{\sigma}_t \in C^1(0,T)$; **2)** the transition network $\boldsymbol{f}_\theta(\boldsymbol{x}_t, t, r)$ is differentiable with respect to t for any $(\boldsymbol{x}_t, r)$; **3)** the scalar coefficient $B_{t,r}$ satisfies $\frac{\mathrm{d}B_{t,r}}{\mathrm{d}t} \neq 0$ for all $(t,r) \in (0,T)^2$.

Under these assumptions, the identity holds, establishing a theoretically complete formulation that is not an infinitesimal approximation but an exact identity valid for any arbitrary time interval (t,r). This formulation decouples the learning of the transition function $\boldsymbol{f}_\theta$ from the specific choice of transport coefficients. This decoupling liberates the model from learning a single, fixed generative dynamic. Instead, *it provides the fundamental, transport-agnostic foundation necessary to enable the design of novel, more expressive transports for complex dynamics where standard PF-ODEs may falter*. To demonstrate this generality, Tab. 1 summarizes how our universal formulation encapsulates and extends existing diffusion transports. We perform a holistic analysis of the effects of transports on the modeling of generative dynamics in the Appendix Tab. 11.

3.3. Scalability and Stability in TiM Training

Remark 1: Making TiM Training Scalable.

A critical challenge in implementing our training target (Eq. (9)) is the computation of the network's time derivative, $\frac{\mathrm{d}\boldsymbol{f}_{\theta^-,t,r}}{\mathrm{d}t}$. Prior work, such as MeanFlow [26, 55, 59] and sCM [47], relies on the Jacobian-Vector Product (JVP) for this task. However, JVP presents a *fundamental roadblock to scalability*. It is not only compute-intensive but, more cripplingly, its reliance on backward-mode automatic differentiation is **incompatible with essential training optimizations**, including FlashAttention [16] and distributed frameworks of FSDP [88]. This incompatibility has effectively rendered JVP-based methods impractical for training billion-parameter foundation models.

Method	Operator		Training		FID		
	FLOPs (G)	Latency (ms)	Throughput (/s)	Memory (GiB)	NFE=1	NFE=8	NFE=50
JVP	48.29	213.14	1.80	**14.89**	**49.75**	26.22	18.11
DDE	**24.14**	**110.08**	**2.40**	15.23	49.91	**26.09**	**17.99**

Table 2. **Derivative Calculation Comparison.** We utilize a TiM-B/4 model for latency, throughput, and memory measurement, with a batch size of 256 on a NVIDIA-A100-40G GPU using BF16 precision.

We break this barrier with the **Differential Derivation Equation (DDE)**, a principled and highly efficient finite-difference approximation:

$$\frac{\mathrm{d}\boldsymbol{f}_{\theta^-,t,r}}{\mathrm{d}t} \approx \frac{\boldsymbol{f}_{\theta^-}(\boldsymbol{x}_{t+\epsilon}, t+\epsilon, r) - \boldsymbol{f}_{\theta^-}(\boldsymbol{x}_{t-\epsilon}, t-\epsilon, r)}{2\epsilon}. \quad (10)$$

As shown in Tab. 2, DDE is not only $\sim 2\times$ faster than JVP but, crucially, its forward-pass-only structure is *natively compatible with FSDP*. This compatibility transforms a previously unscalable training process into one ready for large-scale deployment, making TiM the first model of its kind practical for from-scratch, billion-parameter pre-training.

Remark 2: Making TiM Training Stable.

In addition to scalability, a key challenge in training with arbitrary intervals is managing gradient variance. For example, transitions over very large intervals ($\Delta t \to t$) are easier to make loss spikes. To mitigate this, we introduce a loss weighting scheme that prioritizes short-interval transitions, providing a more stable learning signal.

The weighting function, $w(t,r)$, is a composition of a time-warping function $\tau(\cdot)$ and a kernel function $k(\cdot,\cdot)$:

$$w(t,r) = k(\tau(t), \tau(r)). \quad (11)$$

Here, $\tau(\cdot)$ is a monotonic function that re-parameterizes the time axis. For our final model, we use a tangent space transformation, which effectively stretches the time domain, yielding the specific weighting:

$$w(t,r) = (\sigma_{\text{data}} + \tan(t) - \tan(r))^{-\frac{1}{2}}, \quad (12)$$

where σ_{data} is the standard deviation of the clean data.

Final Learning Objective. Our framework culminates in a scalable and stable learning objective. We train the network $\boldsymbol{f}_\theta$ to predict the dynamic target $\hat{\boldsymbol{f}}$ in Eq. (9). Weighted by the interval function $w(t,r)$, the final objective is:

$$\mathbb{E}_{\boldsymbol{x},\boldsymbol{\varepsilon},t,r} \left[w(t,r) \cdot d\left(\boldsymbol{f}_\theta(\boldsymbol{x}_t, t, r) - \hat{\boldsymbol{f}} \right) \right]. \quad (13)$$

Model	Param.	NFE	GenEval							DPGBench					
			Overall↑	1-Obj.	2-Obj.	Count.	Colors	Pos.	Attr.	Overall↑	Global	Entity	Attribute	Relation	Other
Autoregressive Models															
GPT-4o [1]	-	-	0.84	0.99	0.92	0.85	0.92	0.75	0.61	-	-	-	-	-	-
Emu3-Gen [80]	-	-	0.54	0.98	0.71	0.34	0.81	0.17	0.21	80.60	85.21	86.68	86.84	90.22	83.15
Multi-step Diffusion Models															
Seedream2.0 [28]	-	-	0.84	1.0	0.98	0.91	0.94	0.47	0.75	-	-	-	-	-	-
SD2.1 [58]	865M	100	0.50	0.98	0.51	0.44	0.85	0.07	0.17	63.18	74.63	74.23	75.39	73.49	67.81
SD-XL [56]	2.6B	100	0.55	0.98	0.74	0.39	0.85	0.15	0.23	74.65	83.27	82.43	80.91	86.76	80.41
SD3.5-Medium [19]	2B	100	0.63	0.98	0.78	0.50	0.81	0.24	0.52	84.08	87.90	91.01	88.83	80.70	88.68
SD3.5-Large [19]	8B	128	0.69	0.99	0.89	0.67	0.81	0.24	0.56	83.99	85.62	89.36	87.57	93.00	81.30
SANA-1.5 [82]	4.6B	40	0.81	0.99	0.93	0.86	0.84	0.59	0.65	83.44	84.49	89.32	87.58	93.19	79.60
FLUX.1-Dev [5]	12B	128	0.65	0.98	0.79	0.69	0.76	0.21	0.48	83.57	82.26	90.40	87.33	92.49	78.94
Few-step Distilled Diffusion Models															
SDXL-LCM [50]	2.6B	8	0.40	0.97	0.50	0.12	0.67	0.09	0.07	-	-	-	-	-	-
SDXL-Turbo [64]	2.6B	8	0.50	0.99	0.75	0.07	0.89	0.11	0.20	-	-	-	-	-	-
SANA-Sprint [15]	1.6B	8	0.72	1.0	0.88	0.56	0.87	0.56	0.47	81.55	86.39	88.03	86.21	93.00	74.68
SD3.5-Turbo [63]	8B	8	0.66	0.99	0.81	0.62	0.79	0.25	0.48	79.03	80.12	86.13	84.73	91.86	78.29
FLUX.1-Schnell [5]	12B	8	0.67	0.99	0.90	0.60	0.75	0.27	0.50	84.94	86.62	90.82	88.35	93.45	82.00
Transition Models															
		1	0.67	0.98	0.75	0.52	0.80	0.54	0.44	74.93	82.98	83.64	83.54	91.99	63.20
TiM	865M	8	0.76	0.99	0.87	0.61	0.88	0.63	0.61	81.30	82.01	88.31	87.81	93.37	70.80
		128	0.83	1.0	0.91	0.73	0.91	0.73	0.71	81.62	82.37	88.78	88.54	93.31	76.40

Table 3. **System-level quality comparison of TiM and SOTA methods on GenEval and DPGBench benchmarks.** In the table, 1-NFE denotes a single sampling step; 8-NFE corresponds to four sampling steps with CFG, and other multi-NFE follow the same convention. Compared with multi-step diffusion models and few-step distilled models, TiM offers any-step generation, delivering strong few-step performance and exhibiting consistent, stable improvements as NFE increases.

Method	NFE	Aspect Ratio						Resolution					
		2 : 5	9 : 16	2 : 3	3 : 2	16 : 9	5 : 2	1280	1536	2048	2560	3072	4096
SD3.5-Turbo [63]	8	✗	0.53	0.60	0.58	0.30	✗	0.61	✗	✗	✗	✗	✗
FLUX.1-Schnell [6]	8	0.57	0.61	0.63	0.62	0.59	0.57	0.64	0.58	0.46	0.14	✗	✗
TiM	8	0.55	0.58	0.63	0.64	0.58	0.56	0.70	0.61	0.49	0.48	0.45	0.39
SD3.5-Large [19]	32	0.25	0.48	0.60	0.57	0.16	✗	0.63	✗	✗	✗	✗	✗
FLUX.1-Dev [5]	32	0.48	0.59	0.62	0.60	0.59	0.57	0.62	0.58	0.49	0.27	✗	✗
TiM	32	0.66	0.67	0.72	0.72	0.62	0.64	0.75	0.69	0.63	0.62	0.59	0.53

Table 4. **Benchmarking resolution generation capabilities on GenEval Benchmark.** For aspect ratio generalization, the exact resolutions are: $\{1024 \times 2560, 1024 \times 1856, 1024 \times 1536, 1536 \times 1024, 1856 \times 1024, 2560 \times 1024\}$. ✗: when GenEval score falls below 0.10, we interpret it as evidence that the model fails to generalize to that resolution.

Model	Param.	NFE	FID↓	CLIP↑
PixArt-α [12]	610M	100	6.14	27.55
SDXL [56]	2.6B	100	6.63	29.03
Playground v2.5 [41]	2.6B	100	6.09	29.13
Hunyuan-DiT [43]	1.5B	100	6.54	28.19
SD3.5-Medium [19]	2B	100	11.92	27.83
SD3.5-Turbo [63]	8B	8	11.97	27.35
SD3.5-Large [19]	8B	32	14.68	27.88
FLUX.1-Schnell [6]	12B	8	7.94	28.14
FLUX.1-dev [5]	12B	32	9.19	27.27
		1	6.68	24.80
TiM	865M	8	5.28	26.10
		32	5.65	26.31

Table 5. **Quality comparison on MJHQ30K benchmark.**

This learning objective generalizes the standard PF-ODE supervision to arbitrary state-to-state transitions. The practical implementation, enabled by our efficient DDE calculation, is detailed in Algorithm 1 in the Appendix.

Method	NFE=1	NFE=8	NFE=32	NFE=128
SD3.5-Turbo [63]	0.50	0.66	0.70	0.70
FLUX.1-Schnell [6]	0.68	0.67	0.63	0.58
SD3.5-Large [19]	0.00	0.50	0.69	0.70
FLUX.1-Dev [5]	0.00	0.40	0.64	0.65
TiM	0.67	0.76	0.80	0.83

Table 6. **Benchmarking generation quality across NFEs on the GenEval benchmark (score↑).** We compare a single TiM model against diffusion models (i.e., SD3.5-Large and FLUX.1-Dev) and distilled models (i.e., SD3.5-Turbo and FLUX.1-Schnell).

4. Experiments

4.1. Setup

We use SD-VAE [58] for ImageNet-256 experiments and DC-AE [13] for text-to-image (T2I) experiments. For T2I generation, we use $33M$ images from public datasets [2, 11, 14, 17, 35, 65–67]. We train the model with $865M$ parameters for about 30 days using 16 NVIDIA-A100 GPUs.

Figure 3. **Qualitative Analysis between TiM and existing methods under different NFEs.** TiM delivers superior fidelity and text alignment across all NFEs. In contrast, multi-step diffusion and few-step distilled models exhibit pronounced step–quality trade-offs: SDXL, SD3.5-Large, and FLUX.1-Dev fail to generate images at low NFEs, while SDXL-Turbo, SD3.5-Turbo, and FLUX.1-Schnell produce over-saturated outputs at high NFEs.

Method	Epochs	Params	NFE	**FID↓**
Generative Adversarial Networks				
BigGAN [8]	-	112M	1	6.95
GigaGAN [36]	-	569M	2	3.45
Multi-step Diffusion Models				
Flag-DiT-3B [21]	200	4.23B	500	1.96
Large-DiT-3B [21]	340	4.23B	500	2.10
DiT-XL [54]	1400	675M	500	2.27
SiT-XL [51]	1400	675M	500	2.06
Few-step Consistency Models				
MeanFlow-XL [26]	240	675M	1	3.43
			2	2.93
iCT-XL [69]	-	675M	2	20.30
Shortcut-XL [20]	250	675M	2	10.60
IMM-XL [90]	3840	675M	2	7.77
Any-step Transition Models				
TiM-XL	300	675M	1	3.15
			2	2.47
			100	1.68

Table 7. **Comparison on ImageNet-256 generation.**

Method	NFE=1	NFE=8	NFE=50
Training Objective			
(a) Baseline (SiT-B/4 [51])	309.5	77.26	20.35
(b) TiM-B/4 (*w/* JVP)	49.75	26.22	18.11
(c) TiM-B/4 (*w/* DDE)	49.91	26.09	17.99
(d) + Improved Transport	**46.38**	**23.54**	**15.06**
Architecture			
(e) Shared Time Encoder	79.38	26.23	17.66
(f) Decoupled Time Encoder	**46.38**	**23.54**	**15.06**
Training Strategy (on top of (f))			
(g) + Time-weighting	**46.31**	**22.55**	**14.21**

Table 8. **Ablation studies of Transition Models on the standard ImageNet-256 benchmark (FID↓).** We analyze the effect of training objectives, architecture, and training strategies.

Gemma3-1B-it [76] is utilized as a text encoder. We report the Number of Function Evaluations (NFE) to quantify the sampling steps. Classifier-free guidance (CFG) doubles NFE because each step requires two model evaluations: one conditioned and one unconditioned. We report FID [30] in ImageNet experiments and report GenEval [27], DPG-Bench [34] and MJHQ-30K [41] in T2I experiments.

Model Architecture. The model architecture is based on SiT [51], employing a decoupled embedding strategy with separate time encoders, ϕ_t for absolute time t and $\phi_{\Delta t}$ for transition interval Δt. The final time-conditioning vector is $\mathbf{E}_{t,\Delta t} = \phi_t(t) + \phi_{\Delta t}(\Delta t)$. *For class-guided generation,* the class embedding $\mathbf{E}_c$ is added to the time embedding, jointly

modulating the AdaLN layers. *For text-to-image generation,* the pathways are separate: time embedding $\mathbf{E}_{t,\Delta t}$ modulates the AdaLN layers, while textual features from the prompt are injected via cross-attention mechanisms.

Native-Resolution Training. Previous methods [24, 28, 81] have shown the success of native-resolution training on resolution generalization; therefore, we adopt this strategy for *text-to-image generation*, which preserves the original image resolution and aspect ratio information. See *more details* on T2I training in the Appendix C.1.

Sampling. Since TiM learns the arbitrary state transition on the diffusion trajectory, it supports arbitrary-step sampling when producing images. Given a set of timesteps $\mathcal{T} = \{t_i\}_{i=N}^{0}$ where $t_N = T, t_0 = 0$, we obtain the next state $\boldsymbol{x}_{t_{n-1}}$ given the current state $\boldsymbol{x}_{t_n}$ based on Eq. (5), as illustrated in Algorithm 2 in the Appendix.

4.2. Main Results

Text-to-Image Generation. TiM establishes a new state-of-the-art in performance, efficiency, and flexibility on various benchmarks (Tabs. 3 to 6). It achieves a SOTA FID of 5.25 on MJHQ-30K with 8 NFEs. In GenEval, TiM's 1-NFE performance surpasses 8-NFE distilled models (e.g. SDXL-Turbo), while its 128-NFE quality rivals closed-source models. Also validated in Fig. 3, *TiM alone shows a monotonic quality improvement with NFE.*[6] This unique scalability of NFE contrasts starkly with competitors like SD3.5-Large, which collapse after a few steps and FLUX.1-Schnell, which degrades at many steps. This efficiency is further proven on DPGBench, where 8-NFE TiM outperforms 100-NFE baselines like SDXL. Finally, TiM demonstrates superior generalization across diverse resolutions and aspect ratios, validating its more robust design.

Class-Conditioned Image Generation. Our analysis on the ImageNet-256 evaluates models for quality (FID) and speed (NFE).[7] Tab. 7 shows that multi-step models like Flag-DiT offer high quality (1.96 FID) but are slow (500 NFE), while few-step methods like Shortcut model sacrifice quality (10.60 FID at 2 NFE). TiM excels in both areas, achieving a leading 1.68 FID at 100 NFE and a 2.47 FID at 2 NFE, resolving the speed-quality trade-off.

4.3. Ablation Studies

We conduct a series of ablation studies to validate our design choices, building from a standard diffusion baseline (SiT-B/4 [51] here) to our final TiM configuration. We train a 131M parameter model on ImageNet-256 for 80 epochs. In Tab. 8, we report FID at 1, 8, and 50 NFEs, corresponding to single-step, few-step, and multi-step generation[8].

Transition Objective. As shown in Table 8 (a vs. c), switching from the standard SiT objective to our TiM objective delivers a dramatic improvement in few-step performance, *reducing the 1-NFE FID by over 6×* (309.5 → 49.91) while maintaining strong many-step quality. Our proposed DDE method[9] (c) achieves this performance while being far more scalable than JVP (b), making large-scale training practical. Furthermore, with the improved transport[10] in Tab. 1, TiM (d) reduces the FID in all three NFEs, *showing a more effective dynamic modeling capability*.

Architectural Design. We next analyze the impact of our architectural design on the time encoder. Compared with *Shared Time Encoder*[11] (e), the *Decoupled Time Embedding* (f) provides substantial gains across all sampling steps, lowering the 1-NFE FID from 79.38 to 46.38. This demonstrates that enabling the model to explicitly reason about both absolute time and the transition interval is *complementary and essential* for optimal performance.

Training Strategy. Building on our best architecture (f), we apply our proposed *interval weighting* scheme. This final step provides a consistent boost across the board (f), further refining the model and achieving our best FID scores of 46.31 / 22.55 / 14.21. We conduct an in-depth comparison of alternative weighting schemes in the Appendix Tab. 12.

5. Conclusion and Limitations

Conclusion. This paper introduces Transition Models (TiM), a novel generative model that learns to navigate the entire generative trajectory with unprecedented flexibility. The success of our compact 865M model in outperforming multi-billion parameter giants is not just a new state-of-the-art; it is a testament to a more efficient and powerful paradigm. By achieving monotonic quality improvement from one step to many, and scaling to ultra-high resolutions, TiM demonstrates that a unified model is not only possible but superior. We believe that this work paves the way for a new generation of foundation models that are at once efficient, scalable, and promising in their creative potential.

Limitations and Future Work. Although TiM delivers a significant contribution to the fundamental generative models, ensuring content safety and controllability remains an open challenge, and model fidelity can degrade in scenarios requiring fine-grained detail, such as rendering text and hands. We also observe occasional artifacts at high resolutions (e.g., 3072×4096), likely attributable to biases in the underlying autoencoder. In the future, we would scale up the T2I model with more parameters and larger datasets to explore scalability. Extending the TiM framework to the video domain is also a promising direction.

[6] We provide more qualitative results in the Appendix E.

[7] We also provide the ImageNet-512 results in the Appendix Tab. 14.

[8] 1-NFE: single sampling step; 8-NFE: 4 sampling steps with CFG; 50-NFE: 25 sampling steps with CFG.

[9] We provide a detailed analysis of DDE in the Appendix Tab. 10

[10] We provide a detailed analysis of transports in Appendix Tab. 11.

[11] *Shared Time Encoder* means: time t and interval Δt share the same time encoder for corresponding embeddings.

Acknowledgements

This work was supported by the JC STEM Lab of AI for Science and Engineering, funded by The Hong Kong Jockey Club Charities Trust, the Research Grants Council of Hong Kong (Project No. CUHK14213224).

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 6

[2] adebyollin. Megalith-huggingface. https://huggingface.co/datasets/madebyollin/megalith-10m. 6

[3] Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982. 3

[4] Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models. In *CVPR*, 2023. 23

[5] black-forest labs. Flux.1-dev. https://huggingface.co/black-forest-labs/FLUX.1-dev, . 1, 6

[6] black-forest labs. Flux.1-schnell. https://huggingface.co/black-forest-labs/FLUX.1-schnell, . 1, 6

[7] Nicholas Matthew Boffi, Michael Samuel Albergo, and Eric Vanden-Eijnden. Flow map matching with stochastic interpolants: A mathematical framework for consistency models. *Transactions on Machine Learning Research*, 2025. 1, 2

[8] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 7, 22, 23

[9] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. Accessed: 2024-5-1. 1

[10] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 22

[11] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *CVPR*, 2021. 6

[12] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, et al. Pixart-alpha: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*, 2023. 6

[13] Junyu Chen, Han Cai, Junsong Chen, Enze Xie, Shang Yang, Haotian Tang, Muyang Li, Yao Lu, and Song Han. Deep compression autoencoder for efficient high-resolution diffusion models. *arXiv preprint arXiv:2410.10733*, 2024. 6, 19, 21

[14] Jiuhai Chen, Zhiyang Xu, Xichen Pan, Yushi Hu, Can Qin, Tom Goldstein, Lifu Huang, Tianyi Zhou, Saining Xie, Silvio Savarese, et al. Blip3-o: A family of fully open unified multimodal models-architecture, training and dataset. *arXiv preprint arXiv:2505.09568*, 2025. 6

[15] Junsong Chen, Shuchen Xue, Yuyang Zhao, Jincheng Yu, Sayak Paul, Junyu Chen, Han Cai, Song Han, and Enze Xie. Sana-sprint: One-step diffusion with continuous-time consistency distillation. *arXiv preprint arXiv:2503.09641*, 2025. 6

[16] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023. 5

[17] Karan Desai, Gaurav Kaul, Zubin Aysola, and Justin Johnson. Redcaps: Web-curated image-text data created by the people, for the people. *arXiv preprint arXiv:2111.11431*, 2021. 6

[18] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *NeurIPS*, 2021. 21, 23

[19] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024. 1, 2, 6, 19

[20] Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. *arXiv preprint arXiv:2410.12557*, 2024. 1, 7, 17, 19, 22

[21] Peng Gao, Le Zhuo, Ziyi Lin, Chris Liu, Junsong Chen, Ruoyi Du, Enze Xie, Xu Luo, Longtian Qiu, Yuhang Zhang, et al. Lumina-t2x: Transforming text into any modality, resolution, and duration via flow-based large diffusion transformers. *arXiv preprint arXiv:2405.05945*, 2024. 7, 22, 23

[22] Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. Masked diffusion transformer is a strong image synthesizer. *arXiv preprint arXiv:2303.14389*, 2023. 22

[23] Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. Mdtv2: Masked diffusion transformer is a strong image synthesizer. *arXiv preprint arXiv:2303.14389*, 2023. 22

[24] Yu Gao, Lixue Gong, Qiushan Guo, Xiaoxia Hou, Zhichao Lai, Fanshi Li, Liang Li, Xiaochen Lian, Chao Liao, Liyang Liu, et al. Seedream 3.0 technical report. *arXiv preprint arXiv:2504.11346*, 2025. 8

[25] Zhengyang Geng, Ashwini Pokle, William Luo, Justin Lin, and J Zico Kolter. Consistency models made easy. *arXiv preprint arXiv:2406.14548*, 2024. 2

[26] Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *arXiv preprint arXiv:2505.13447*, 2025. 1, 2, 5, 7, 17, 19, 22

[27] Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36:52132–52152, 2023. 1, 2, 7

[28] Lixue Gong, Xiaoxia Hou, Fanshi Li, Liang Li, Xiaochen Lian, Fei Liu, Liyang Liu, Wei Liu, Wei Lu, Yichun

Shi, et al. Seedream 2.0: A native chinese-english bilingual image generation foundation model. *arXiv preprint arXiv:2503.07703*, 2025. 6, 8

[29] Ali Hatamizadeh, Jiaming Song, Guilin Liu, Jan Kautz, and Arash Vahdat. Diffit: Diffusion vision transformers for image generation. *arXiv preprint arXiv:2312.02139*, 2023. 23

[30] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NeurIPS*, 2017. 7

[31] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NeurIPS*, 2017. 21

[32] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 2020. 2, 4, 16, 20

[33] Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In *ICML*, 2023. 22, 23

[34] Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. Ella: Equip diffusion models with llm for enhanced semantic alignment. *arXiv preprint arXiv:2403.05135*, 2024. 7

[35] jackyhate. text-to-image-2m. https://huggingface.co/datasets/jackyhate/text-to-image-2M. 6

[36] Minguk Kang, Jun-Yan Zhu, Richard Zhang, Jaesik Park, Eli Shechtman, Sylvain Paris, and Taesung Park. Scaling up gans for text-to-image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10124–10134, 2023. 7, 22

[37] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *NeurIPS*, 2022. 1, 2, 3, 4, 15, 16, 20

[38] Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24174–24184, 2024. 16, 23

[39] 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. 17, 18

[40] T. Kynkäänniemi, T. Karras, S. Laine, and T Lehtinen, J.and Aila. Improved precision and recall metric for assessing generative models. *NeurIPS*, 2019. 21

[41] Daiqing Li, Aleks Kamko, Ehsan Akhgari, Ali Sabet, Linmiao Xu, and Suhail Doshi. Playground v2. 5: Three insights towards enhancing aesthetic quality in text-to-image generation. *arXiv preprint arXiv:2402.17245*, 2024. 6, 7

[42] Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *Advances in Neural Information Processing Systems*, 37:56424–56445, 2024. 22

[43] Zhimin Li, Jianwei Zhang, Qin Lin, Jiangfeng Xiong, Yanxin Long, Xinchi Deng, Yingfang Zhang, Xingchao Liu, Minbin Huang, Zedong Xiao, et al. Hunyuan-dit: A powerful multi-resolution diffusion transformer with fine-grained chinese understanding. *arXiv preprint arXiv:2405.08748*, 2024. 6

[44] Shanchuan Lin, Anran Wang, and Xiao Yang. Sdxl-lightning: Progressive adversarial diffusion distillation. *arXiv preprint arXiv:2402.13929*, 2024. 1

[45] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 3, 4, 13, 16, 19, 20

[46] Xingchao Liu, Chengyue Gong, and qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2023. 2, 16

[47] Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models. *arXiv preprint arXiv:2410.11081*, 2024. 1, 2, 3, 4, 5, 13, 15, 16, 17, 20, 23

[48] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in neural information processing systems*, 35:5775–5787, 2022. 1

[49] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *Machine Intelligence Research*, pages 1–22, 2025. 3

[50] Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023. 2, 6

[51] Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. *arXiv preprint arXiv:2401.08740*, 2024. 7, 8, 22, 23

[52] C. Nash, J. Menick, S. Dieleman, and P. W Battaglia. Generating images with sparse representations. *arXiv preprint arXiv:2103.03841*, 2021. 21

[53] Ziqi Pang, Tianyuan Zhang, Fujun Luan, Yunze Man, Hao Tan, Kai Zhang, William T Freeman, and Yu-Xiong Wang. Randar: Decoder-only autoregressive visual generation in random orders. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 45–55, 2025. 22

[54] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023. 1, 7, 22, 23

[55] Yansong Peng, Kai Zhu, Yu Liu, Pingyu Wu, Hebei Li, Xiaoyan Sun, and Feng Wu. Flow-anchored consistency models. *arXiv preprint arXiv:2507.03738*, 2025. 1, 5, 17

[56] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 1, 6

[57] Yuxi Ren, Xin Xia, Yanzuo Lu, Jiacheng Zhang, Jie Wu, Pan Xie, Xing Wang, and Xuefeng Xiao. Hyper-sd: Trajectory segmented consistency model for efficient image synthesis. *Advances in Neural Information Processing Systems*, 37:117340–117362, 2025. 2

[58] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 6, 21, 22

[59] Amirmojtaba Sabour, Sanja Fidler, and Karsten Kreis. Align your flow: Scaling continuous-time flow map distillation. *arXiv preprint arXiv:2506.14603*, 2025. 1, 5

[60] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X Chen. Improved techniques for training gans. *NeurIPS*, 2016. 21

[61] Axel Sauer, Katja Schwarz, and Andreas Geiger. Stylegan-xl: Scaling stylegan to large diverse datasets. In *ACM SIGGRAPH 2022 conference proceedings*, 2022. 23

[62] Axel Sauer, Katja Schwarz, and Andreas Geiger. Stylegan-xl: Scaling stylegan to large diverse datasets. In *ACM SIGGRAPH 2022 conference proceedings*, pages 1–10, 2022. 22

[63] 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*, pages 1–11, 2024. 1, 2, 6

[64] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *European Conference on Computer Vision*, pages 87–103. Springer, 2024. 1, 2, 6

[65] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021. 6

[66] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018.

[67] Vasu Singla, Kaiyu Yue, Sukriti Paul, Reza Shirkavand, Mayuka Jayawardhana, Alireza Ganjdanesh, Heng Huang, Abhinav Bhatele, Gowthami Somepalli, and Tom Goldstein. From pixels to prose: A large dataset of dense image captions. *arXiv preprint arXiv:2406.10328*, 2024. 6

[68] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 1, 3

[69] Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. In *The Twelfth International Conference on Learning Representations*. 1, 2, 7, 17, 22

[70] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *NeurIPS*, 2019. 4, 16, 20

[71] 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. 1, 2, 3, 4, 13, 14, 15, 16, 20

[72] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning*, pages 32211–32252. PMLR, 2023. 1, 2, 17, 18

[73] Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024. 22

[74] Peng Sun, Yi Jiang, and Tao Lin. Unified continuous generative models. *arXiv preprint arXiv:2505.07447*, 2025. 3, 13

[75] Zhicong Tang, Jianmin Bao, Dong Chen, and Baining Guo. Diffusion models without classifier-free guidance. *arXiv preprint arXiv:2502.12154*, 2025. 19, 20

[76] Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025. 7

[77] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in neural information processing systems*, 37:84839–84865, 2024. 22, 23

[78] Fu-Yun Wang, Zhaoyang Huang, Alexander Bergman, Dazhong Shen, Peng Gao, Michael Lingelbach, Keqiang Sun, Weikang Bian, Guanglu Song, Yu Liu, et al. Phased consistency models. *Advances in Neural Information Processing Systems*, 37:83951–84009, 2025. 2, 17, 18

[79] Shuai Wang, Zexian Li, Tianhui Song, Xubin Li, Tiezheng Ge, Bo Zheng, and Limin Wang. Exploring dcn-like architecture for fast image generation with arbitrary resolution. *Advances in Neural Information Processing Systems*, 37:87959–87977, 2024. 22, 23

[80] Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024. 6

[81] Zidong Wang, Lei Bai, Xiangyu Yue, Wanli Ouyang, and Yiyuan Zhang. Native-resolution image synthesis. *arXiv preprint arXiv:2506.03131*, 2025. 8

[82] Enze Xie, Junsong Chen, Yuyang Zhao, Jincheng Yu, Ligeng Zhu, Chengyue Wu, Yujun Lin, Zhekai Zhang, Muyang Li, Junyu Chen, et al. Sana 1.5: Efficient scaling of training-time and inference-time compute in linear diffusion transformer. *arXiv preprint arXiv:2501.18427*, 2025. 6

[83] 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. 1

[84] 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*, pages 6613–6623, 2024. 1, 2

[85] 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, 2025. 2

[86] Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Vighnesh Birodkar, Agrim Gupta, Xiuye Gu, et al. Language model beats diffusion–tokenizer is key to visual generation. *arXiv preprint arXiv:2310.05737*, 2023. 22, 23

[87] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. *arXiv preprint arXiv:2410.06940*, 2024. 22, 23

[88] Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, et al. Pytorch fsdp: experiences on scaling fully sharded data parallel. *arXiv preprint arXiv:2304.11277*, 2023. 5, 20

[89] Hongkai Zheng, Weili Nie, Arash Vahdat, and Anima Anandkumar. Fast training of diffusion models with masked transformers. *TMLR*, 2023. 23

[90] Linqi Zhou, Stefano Ermon, and Jiaming Song. Inductive moment matching. *arXiv preprint arXiv:2503.07565*, 2025. 7, 22

[91] 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. 1