ANY-STEP GENERATION VIA N-TH ORDER RECURSIVE CONSISTENT VELOCITY FIELD ESTIMATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent advances in few-step generative models (typically 1-8 steps), such as consistency models, have yielded impressive performance. However, their broader adoption is hindered by significant challenges, including substantial computational overhead, the reliance on complex multi-component loss functions, and intricate multi-stage training strategies that lack end-to-end simplicity. These limitations impede their scalability and stability, especially when applied to large-scale models. To address these issues, we introduce N-th order Recursive Consistent velocity field estimation for Generative Modeling (RCGM), a novel framework that unifies many existing approaches. Within this framework, we reveal that conventional one-step methods, such as consistency and MeanFlow models, are special cases of 1st-order RCGM. This insight enables a natural extension to higher-order scenarios ($N \geq 2$), which exhibit markedly improved training stability and achieve state-of-the-art (SOTA) performance. For instance, on ImageNet 256×256, RCGM enables a 675M parameter diffusion transformer to achieve a 1.48 FID score in just 2 sampling steps. Crucially, RCGM facilitates the stable full-parameter training of a large-scale (20B) unified multi-modal model, attaining a 0.86 GenEval score in 4 steps. In contrast, conventional 1st-order approaches, such as consistency and MeanFlow models, typically suffer from training instability, model collapse, or memory constraints under comparable settings. Code will be publicly available.

1 INTRODUCTION

Existing PF-ODE-based generative models (Song et al., 2020b), encompassing diffusion models (Ho et al., 2020; Song et al., 2020a), flow-matching models (Lipman et al., 2022; Ma et al., 2024), and consistency models (Song et al., 2023; Lu & Song, 2024), have demonstrated remarkable success in synthesizing high-fidelity data across diverse applications, including image and video generation (Google, 2025a; OpenAI, 2025; Xie et al., 2024a; Ho et al., 2022; Chen et al., 2025c; Wu et al., 2025a).

Table 1: Comparison of different methods' **reliance** on a 1st-order objective and JVP. Our method is independent of both.

Method	Independent of	
	1st-Order	JVP
CM (Song et al., 2023)	×	✓
sCM (Lu & Song, 2024)	×	×
MeanFlow (Geng et al., 2025)	×	×
RCGM (Ours)	✓	✓

Within this landscape, few-step generative models (Song et al., 2023; Frans et al., 2024; Geng et al., 2025) are particularly prized for their ability to generate high-quality samples with significantly reduced computational cost, a critical factor for practical deployment. However, the pursuit of this efficiency has introduced a distinct set of formidable challenges that plague current SOTA methods: **(a)** a prohibitive computational and memory burden during training, often necessitating expensive Jacobian-vector products (JVP) (Geng et al., 2025; Lu & Song, 2024); **(b)** the need to combine multiple losses and train auxiliary models, e.g., combining consistency loss with adversarial loss (Chen et al., 2025c) or training an additional fake image generation model (Yin et al., 2024b;a; Sauer et al., 2024a); **(c)** a fractured theoretical landscape, where highly related methods like consistency models (Song et al., 2023), shortcut models (Frans et al., 2024), and MeanFlow (Geng et al., 2025) have been developed in isolation, lacking a common theoretical foundation.

These challenges restrict their broader application, particularly in generalizing to large-scale models with guaranteed stability and efficiency. For instance, our experiments show that existing one-step models, such as consistency models, often suffer from training instability and high computational

Figure 1: **Visualization results of RCGM on Qwen-Image-20B.** The images shown were generated by the RCGM-tuned Qwen-Image-20B model using **NFE=8** (GenEval score=0.87). Please zoom in to see finer details.

demands when scaled up, frequently resulting in model collapse or GPU memory exhaustion (see Tab. 4). We argue that this fragility originates from their implicit reliance on a 1st-order recursive training objective (cf. Sec. 2 and Sec. 3). This critical insight leads us to the central question:

Problem 1 . *Can we develop a unified and simple framework that: (a) encompasses existing few-step generative models as a special case; (b) enhances training stability and generalization to large-scale models by moving beyond the 1st-order limitation, thereby obviating the need for JVP or training auxiliary models?*

To address these challenges, we propose RCGM, a novel and principled framework that unifies and generalizes existing approaches. *Within our framework, we show that conventional one-step models (e.g., consistency models and MeanFlow) correspond to the special case of* 1*st-order* RCGM.

Notably, RCGM naturally supports higher-order formulations (i.e., $N \geq 2$). These higher-order variants utilize more comprehensive trajectory information from the PF-ODE, which contributes to substantially improved training stability. This stability enables successful training in demanding large-scale settings where 1st-order models often fail, ultimately achieving SOTA performance without resorting to complex workarounds. **In summary, our contributions are:**

(a) We propose RCGM, a unified framework that contextualizes existing few-step generative models as a specific 1st-order case and generalizes them to arbitrary N-th order formulations.
(b) We identify and empirically verify that higher-order RCGM (e.g., 2nd-order) can exhibit superior training stability and robustness, enabling effective scaling to larger and more complex model architectures (cf. Sec. 4).
(c) Our method achieves SOTA performance across a range of standard benchmarks, outperforming existing methods in few-step generation tasks while maintaining computational efficiency.

As detailed in Fig. 1, Sec. 4 and App. D, our approach consistently matches or surpasses SOTA methods across various datasets, architectures, and resolutions, setting a new standard for efficient, high-fidelity generative modeling.

2 Preliminaries

Let $p(\mathbf{x})$ be the data distribution for a given training set D. This distribution can also be conditional, denoted as $p(\mathbf{x}|\mathbf{c})$ for a given condition $\mathbf{c}$. Diffusion-based generative models aim to learn a transfor-

mation from a simple prior distribution $p(\mathbf{z})$, typically the standard Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{I})$, to the complex target data distribution $p(\mathbf{x})$.

This is often achieved by learning to reverse a forward noising process. The forward process gradually perturbs a clean data sample $\mathbf{x} \sim p(\mathbf{x})$ into a noisy intermediate sample $\mathbf{x}_t$ using a predefined trajectory, such as $\mathbf{x}_t = \alpha(t)\mathbf{z} + \gamma(t)\mathbf{x}$, where $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The time variable t spans the interval $[0, 1]$, with the perturbation effect intensifying as t increases. The scheduling functions $\alpha(t)$ and $\gamma(t)$ are continuously differentiable, i.e., $\alpha(t), \gamma(t) \in C^1[0, 1]$, and satisfy the boundary conditions: $\alpha(0) = 0, \gamma(0) = 1$ (yielding the data) and $\alpha(1) = 1, \gamma(1) = 0$ (yielding pure noise).

More formally, diffusion models learn a function that guides the transformation of samples along the trajectory of the Probability Flow Ordinary Differential Equation (PF-ODE) (Song et al., 2020b), which connects the prior distribution $p(\mathbf{z})$ to the data distribution $p(\mathbf{x})$.

In this paper, we define a general prediction function $\boldsymbol{f}(\mathbf{x}_t, r) := \mathbf{x}_r - \mathbf{x}_t$ that estimates the displacement from $\mathbf{x}_t$ to a target $\mathbf{x}_r$, with further details in (6). This function aims to predict the target point $\mathbf{x}_r$ from the current point $\mathbf{x}_t$ along a specific PF-ODE trajectory. In the following sections, we will introduce several prominent learning paradigms for deep generative models.

2.1 0TH-ORDER: DIFFUSION AND FLOW-MATCHING MODELS

Diffusion and Flow-Matching Models (Ho et al., 2020; Song et al., 2020b; Lipman et al., 2022; Sun et al., 2025). Recent work by Sun et al. (Sun et al., 2025) established a unified framework for diffusion and flow-matching models. This framework reveals that both paradigms aim to learn the same PF-ODE (1), but they differ in their underlying transport processes (i.e., their specific choices of $\alpha(t)$ and $\gamma(t)$) and training objectives.

Specifically, a neural network $\boldsymbol{F}_{\boldsymbol{\theta}}$ is trained by minimizing a general objective of the form: $\mathbb{E}_{\mathbf{x}_t,t}\left[d\left(\boldsymbol{F}_{\boldsymbol{\theta}}(\mathbf{x}_t), \hat{\alpha}(t)\mathbf{z} + \hat{\gamma}(t)\mathbf{x}\right)\right]$, where $d(\cdot,\cdot)$ denotes a distance metric. As derived in (Sun et al., 2025), the output of the trained network, $\boldsymbol{F}_t := \boldsymbol{F}_{\boldsymbol{\theta}}(\mathbf{x}_t)$, can be used to construct the component functions: $\boldsymbol{f}^{\mathbf{x}}(\boldsymbol{F}_t, \mathbf{x}_t, t) := \frac{\alpha(t)\cdot\boldsymbol{F}_t - \hat{\alpha}(t)\cdot\mathbf{x}_t}{\alpha(t)\cdot\hat{\gamma}(t) - \hat{\alpha}(t)\cdot\gamma(t)}$ and $\boldsymbol{f}^{\mathbf{z}}(\boldsymbol{F}_t, \mathbf{x}_t, t) := \frac{\hat{\gamma}(t)\cdot\mathbf{x}_t - \gamma(t)\cdot\boldsymbol{F}_t}{\alpha(t)\cdot\hat{\gamma}(t) - \hat{\alpha}(t)\cdot\gamma(t)}$. These components, in turn, define the velocity field of the PF-ODE: $\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} = \frac{\mathrm{d}\alpha(t)}{\mathrm{d}t} \cdot \boldsymbol{f}^{\mathbf{z}}(\boldsymbol{F}_t, \mathbf{x}_t, t) + \frac{\mathrm{d}\gamma(t)}{\mathrm{d}t} \cdot \boldsymbol{f}^{\mathbf{x}}(\boldsymbol{F}_t, \mathbf{x}_t, t)$. The sampling process then involves numerically integrating this velocity field to solve the PF-ODE. The integration proceeds backward in time, starting from a prior sample $\mathbf{x}_1 \sim p(\mathbf{z})$ at $t = 1$ and ending at $t = 0$ to produce a data sample from $p(\mathbf{x})$.

Within our framework, we adopt a zeroth-order inductive learning perspective to interpret this process, a view supported by Fig. 2 (a). Specifically, for a sufficiently small step Δt, the prediction function's learning target becomes the product of the velocity field and the time step:

$$\boxed{\boldsymbol{f}(\mathbf{x}_t, t - \Delta t) \leftarrow \frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} \cdot \Delta t \quad \text{as} \quad \Delta t \to 0.}$$

In essence, given the current state $\mathbf{x}_t$, the prediction function $\boldsymbol{f}$ directly learns to predict the displacement required to approximate the next state, $\mathbf{x}_{t-\Delta t}$, on the PF-ODE path.

2.2 1ST-ORDER: RECURSIVE CONSISTENCY MODELS

Consistency Models (Song et al., 2023; Lu & Song, 2024; Sun et al., 2025). Consistency models are designed to bypass the iterative nature of diffusion models. Their primary goal is to learn a function that maps any noisy state $\mathbf{x}_t$ directly to the clean data endpoint $\mathbf{x}_0$ in a single step. This is achieved by estimating the endpoint of the PF-ODE trajectory originating from $\mathbf{x}_t$, using the function $\mathbf{x}_0 = \boldsymbol{f}^{\mathbf{x}}(\boldsymbol{F}_t, \mathbf{x}_t, t)$.

The training objective is specifically designed to instill a crucial "consistency" property. This property ensures coherence between the model's predictions for the clean data, even when originating from two temporally adjacent noisy states that are separated by a finite time interval $\Delta t > 0$: $\mathbb{E}_{\mathbf{x}_t,t}\left[d\left(\boldsymbol{f}^{\mathbf{x}}(\boldsymbol{F}_t, \mathbf{x}_t, t), \mathrm{stopgrad}(\boldsymbol{f}^{\mathbf{x}}(\boldsymbol{F}_{t-\Delta t}, \mathbf{x}_{t-\Delta t}, t - \Delta t))\right)\right]$. A known limitation of discrete-time consistency models is their sensitivity to the choice of Δt, which often requires manually tuned annealing schedules for efficient training (Song & Dhariwal, 2023; Geng et al., 2024). This challenge was later addressed by continuous consistency models, which derive their training objective by taking the limit as $\Delta t \to 0$ (Lu & Song, 2024).

We interpret this process through a **1st-order inductive learning** lens, a perspective supported by our visualizations in Fig. 2 and theoretical analysis in App. E.1.1. This view frames the learning objective as a recursive formulation:

$$\boxed{\boldsymbol{f}(\mathbf{x}_t, 0) \leftarrow \frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} \cdot \Delta t + \boldsymbol{f}(\mathbf{x}_{t-\Delta t}, 0)}$$

This recursive principle—approximating a long-range prediction by combining an infinitesimal step with another long-range prediction—is also reflected in follow-up works (Frans et al., 2024; Geng et al., 2025). For instance, shortcut models (Frans et al., 2024) employ a similar self-recursive formulation and generalize it to predict between arbitrary time points t and $r \in [0, t]$: $\boldsymbol{f}(\mathbf{x}_t, r) \leftarrow \boldsymbol{f}(\mathbf{x}_t, s) + \boldsymbol{f}(\mathbf{x}_s, r)$. This is then combined with a flow-matching objective to train one-step generative models. More recently, MeanFlow (Geng et al., 2025) extended this idea by training a one-step model with the recursive objective $\boldsymbol{f}(\mathbf{x}_t, r) \leftarrow \frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} \cdot \Delta t + \boldsymbol{f}(\mathbf{x}_{t-\Delta t}, r)$ for any target time r.

In summary, while diffusion and flow-matching models are inherently multi-step frameworks, consistency models represent a paradigm shift towards few-step or one-step generation.

3 Methodology

We begin by deriving a recursive, N-th order velocity field estimator through the segmented integration of the Probability Flow ODE (PF-ODE) trajectory (Sec. 3.1). Building on this formulation, we introduce a unified training objective that enables any-step generation (Sec. 3.2). Finally, we discuss key practical considerations for implementing our method, RCGM (Sec. 3.3).

3.1 Segmented Integration along the PF-ODE Trajectory

Our methodology is grounded in the PF-ODE formulation, where a trajectory from a prior distribution to the data distribution is defined by a velocity field $\mathbf{v}(\mathbf{x}_\tau, \tau)$. For a diffusion process specified by $\mathbf{x}_t = \alpha(t)\mathbf{z} + \gamma(t)\mathbf{x}_0$, this velocity is given by (Song et al., 2020b; Sun et al., 2025):

$$\mathbf{v}(\mathbf{x}_\tau, \tau) := \frac{\gamma'(\tau)}{\gamma(\tau)}\mathbf{x}_\tau - \left[\alpha(\tau)\alpha'(\tau) - \frac{\gamma'(\tau)}{\gamma(\tau)}\alpha(\tau)^2\right]\nabla_{\mathbf{x}_\tau}\log p_\tau(\mathbf{x}_\tau)\,. \tag{1}$$

The integral form of this ODE connects any two points $\mathbf{x}_t$ and $\mathbf{x}_{t_{N+1}}$ on a trajectory. We proceed by partitioning the integration interval $[t, t_{N+1}]$ with N intermediate points, where $t = t_0 > t_1 > \cdots > t_{N+1}$. This segmentation allows us to decompose the total displacement into a sum over the sub-intervals:

$$\mathbf{x}_{t_{N+1}} - \mathbf{x}_t = \sum_{i=0}^{N}\int_{t_i}^{t_{i+1}}\mathbf{v}(\mathbf{x}_\tau, \tau)\mathrm{d}\tau = \int_{t_0}^{t_1}\mathbf{v}(\mathbf{x}_\tau, \tau)\mathrm{d}\tau + \sum_{i=1}^{N}\int_{t_i}^{t_{i+1}}\mathbf{v}(\mathbf{x}_\tau, \tau)\mathrm{d}\tau\,. \tag{2}$$

The core of our approach is to approximate the first integral segment. For a sufficiently small time step $\Delta t := t_1 - t_0$, a 1st-order Taylor approximation (i.e., a forward Euler step) is justified:

$$\int_{t_0}^{t_1}\mathbf{v}(\mathbf{x}_\tau, \tau)\mathrm{d}\tau \approx \mathbf{v}(\mathbf{x}_{t_0}, t_0)\Delta t = \frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t}\Delta t\,. \tag{3}$$

Substituting this approximation into the exact identity from (2) yields the relationship:

$$\mathbf{x}_{t_{N+1}} \approx \mathbf{x}_t + \frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t}\Delta t + \sum_{i=1}^{N}\int_{t_i}^{t_{i+1}}\mathbf{v}(\mathbf{x}_\tau, \tau)\mathrm{d}\tau\,. \tag{4}$$

By rearranging (4), we obtain our final estimator. We define the **recursive N-th order velocity field estimation** as the target derived from this multi-step formula:

$$\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} \approx \frac{1}{\Delta t}\left[(\mathbf{x}_{t_{N+1}} - \mathbf{x}_t) - \sum_{i=1}^{N}\int_{t_i}^{t_{i+1}}\mathbf{v}(\mathbf{x}_\tau, \tau)\mathrm{d}\tau\right]\,. \tag{5}$$

This formulation is termed **recursive** because the estimation of the velocity $\mathbf{v}$ at time t depends on the integral of the same velocity field over future time steps. The "N-th order" designation refers to the N integral correction terms that refine the estimate beyond a simple one-step approximation, thereby providing a more accurate target for model training.

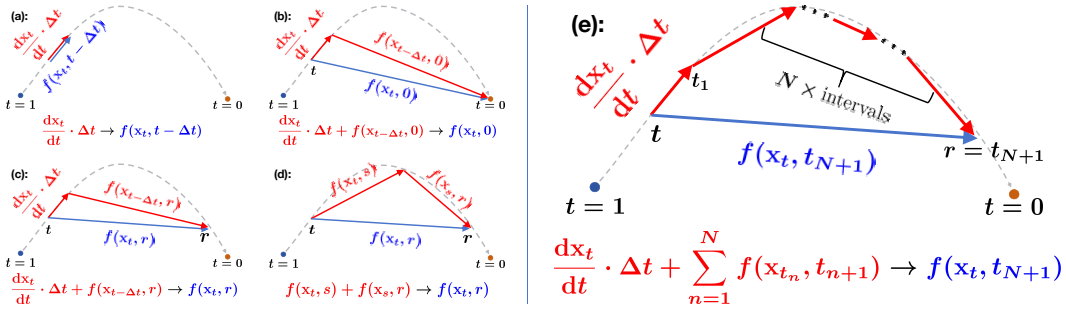


Figure 2: **A conceptual illustration of our proposed framework, RCGM, which generalizes existing generative models by formulating them within a unified higher-order structure. Trajectories map a current state to a target learning state.** **(a)** Standard diffusion (Ho et al., 2020) and flow-matching (Lipman et al., 2022) models correspond to the 0th-order case ($N = 0$) of our framework. **(b-d)** Prominent one-step models, including consistency models (Song et al., 2023), MeanFlow (Geng et al., 2025), and shortcut models (Frans et al., 2024), are special instances of the 1st-order case ($N = 1$). **(e)** RCGM extends this hierarchy to arbitrary orders ($N \geq 0$), enabling the use of higher-order information for potentially more robust training dynamics.

3.2 A Unified Training Framework for Any-Step Generation

Our goal is to train an *any-step* generative model capable of predicting the state $\mathbf{x}_r$ at any future time $r < t$ from the current state $\mathbf{x}_t$ along a given PF-ODE trajectory. To this end, we define a *displacement function* $\boldsymbol{f}(\mathbf{x}_t, r)$ that maps the current state to the total displacement required to reach the target state:

$$\boldsymbol{f}(\mathbf{x}_t, r) := \mathbf{x}_r - \mathbf{x}_t = \int_t^r \mathbf{v}(\mathbf{x}_\tau, \tau)\mathrm{d}\tau\,, \quad r \in [0, t]\,. \tag{6}$$

Using this definition, we can reformulate the recursive N-th order velocity estimator from (5) entirely in terms of displacements:

$$\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} \approx \frac{1}{\Delta t}\left[\boldsymbol{f}(\mathbf{x}_t, t_{N+1}) - \sum_{i=1}^{N} \boldsymbol{f}(\mathbf{x}_{t_i}, t_{i+1})\right]. \tag{7}$$

This identity forms the foundation of our training objective. It provides a multi-step target for the instantaneous velocity $\mathrm{d}\mathbf{x}_t/\mathrm{d}t$, which is known analytically from the PF-ODE formulation (cf. (1)).

We parameterize the displacement function with a parameterized model $\boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_t, r)$. To train $\boldsymbol{f}_{\boldsymbol{\theta}}$, we enforce the identity in (7). The terms $\boldsymbol{f}(\mathbf{x}_{t_i}, t_{i+1})$ for $i \geq 1$ represent future displacements and are treated as fixed targets during optimization. Following standard practice in consistency training (Song et al., 2023), we use a target model $\boldsymbol{f}_{\boldsymbol{\theta}^-}$ (e.g., an exponential moving average of $\boldsymbol{\theta}$ or a periodically updated copy) for these terms, applying a stop-gradient operator to prevent backpropagation through them. This yields the following learning objective:

$$\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{x}_0, \mathbf{z}, \{t_i\}_{i=0}^{N+1}}\left[d\left(\underbrace{\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t}}_{\text{True Velocity}}, \underbrace{\frac{1}{\Delta t}\left[\boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_t, t_{N+1}) - \sum_{i=1}^{N} \boldsymbol{f}_{\boldsymbol{\theta}^-}(\mathbf{x}_{t_i}, t_{i+1})\right]}_{\text{Model's Velocity Estimate}}\right)\right], \tag{8}$$

where $\mathbf{x}_t = \alpha(t)\mathbf{z} + \gamma(t)\mathbf{x}_0$ with $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, time points are sampled hierarchically (e.g., $t \sim \mathcal{U}[0, T]$, $t_1 \sim \mathcal{U}[0, t)$, etc.), and $d(\cdot, \cdot)$ is a suitable metric, such as the squared ℓ_2-norm.

This unified formulation elegantly generalizes several established generative modeling paradigms:

(a) **For** $N = 0$, the objective simplifies to matching $\mathrm{d}\mathbf{x}_t/\mathrm{d}t$ with $\boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_t, t_1)/\Delta t$. This is equivalent to the objectives used in score-based diffusion models (Song et al., 2020b) and flow matching (Lipman et al., 2022).

(b) **For** $N = 1$, the objective corresponds to those of one-step consistency models (Song et al., 2023; Lu & Song, 2024) and shortcut-based methods (Frans et al., 2024), which use a single future segment to define the training target.

By extending this framework to higher orders ($N \geq 2$), our approach leverages multiple future steps to construct a more robust and stable training signal. As we demonstrate in our experiments (Sec. 4), this generalization improves model performance and convergence across diverse generation tasks.

Notably, regardless of the setting of N, our training objective requires only a single model forward pass with gradient calculation and N forward passes without. This design avoids increased GPU memory costs during training, making it feasible for large-scale models [1]. A detailed discussion on the setting of N is provided in Sec. 4.2.

3.3 Practical Implementation of RCGM

In this section, we detail several key aspects of the practical implementation of our method, RCGM. We discuss the parameterization of our neural network under a linear transport path, the use of an enhanced target score function to improve performance, the strategy for conditioning the model on both input and target times, and the formulation of a practical loss function for stable and effective training.

Linear transport and network parameterization. We employ the linear transport path common in flow-matching literature (Lipman et al., 2022; Ma et al., 2024; Xie et al., 2024a), defined by coefficients $\alpha(t) = t$ and $\gamma(t) = 1 - t$. This transport corresponds to a constant velocity field, implying that the displacement between any two states $\mathbf{x}_r$ and $\mathbf{x}_t$ is directly proportional to the time difference $r - t$. This property motivates our parameterization of the predictive function $\boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_t, t, r)$, which estimates the displacement from $\mathbf{x}_r$ to $\mathbf{x}_t$, as: $\boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_t, t, r) = \boldsymbol{F}_{\boldsymbol{\theta}}(\mathbf{x}_t, t, r) \cdot (r - t)$, where $\boldsymbol{F}_{\boldsymbol{\theta}}$ is a neural network designed to approximate the average displacement $(\mathbf{x}_r - \mathbf{x}_t)/(r - t)$.

Enhanced target score function. The performance of continuous generative models can be significantly improved by incorporating guidance during the training or sampling process (Ho & Salimans, 2022; Dhariwal & Nichol, 2021; Karras et al., 2022). This is achieved by modifying the conditional target score function from $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$ (defined in (1)) to an enhanced version: $\nabla_{\mathbf{x}_t} \log \left(p_t(\mathbf{x}_t|\mathbf{c}) \left(p_{t,\boldsymbol{\theta}}(\mathbf{x}_t|\mathbf{c}) / p_{t,\boldsymbol{\theta}}(\mathbf{x}_t)\right)^{\zeta} \right)$, where $\zeta \in (0, 1)$ is the enhancement ratio. We follow the same implementation as previous studies (Frans et al., 2024; Sun et al., 2025).

Input time conditioning. Our method learns a continuous-time model, $\boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_t, r)$, designed to predict the state $\mathbf{x}_r$ at a target time r from an initial state $\mathbf{x}_t$ along the probability flow ODE (PF-ODE) trajectory. To accurately map between arbitrary time points, the model must be effectively conditioned on both the input time t and the target time r. Following standard practice (Ho et al., 2020; Frans et al., 2024), we employ a time embedding technique where t and r are embedded into vector representations separately. These embeddings, along with the input $\mathbf{x}_t$, are then fed into the neural network $\boldsymbol{F}_{\boldsymbol{\theta}}$, redefining the model as $\boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_t, r) = \boldsymbol{F}_{\boldsymbol{\theta}}(\mathbf{x}_t, t, r) \cdot (r - t)$.

Practical loss design. The training objective in (8) necessitates a carefully designed loss function. While the ℓ_2-norm is a standard choice for the metric $d(\cdot, \cdot)$ (Ho et al., 2020; Song et al., 2020a), directly optimizing the original objective is suboptimal. We observe that the magnitude of the model output, $\boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_t, r) \approx \boldsymbol{F}_{\boldsymbol{\theta}} \cdot (r - t)$, scales linearly with the time interval $(r - t)$. This introduces an implicit, scale-dependent weighting that causes optimization instability, as larger time steps dominate the gradients. To rectify this bias and stabilize training, we employ a variance-reduction technique inspired by Lu & Song (2024). specifically, by leveraging the gradient identity derived in Lu & Song (2024) (i.e., $\nabla_\theta \mathbb{E}[\boldsymbol{F}_\theta^\top \boldsymbol{y}] = \frac{1}{2}\nabla_\theta \mathbb{E}[\|\boldsymbol{F}_\theta - \boldsymbol{F}_{\theta^-} + \boldsymbol{y}\|_2^2]$), we reformulate our objective into a regression form that decouples the gradient scale from the time interval. Substituting this into (8) yields our final training objective:

$$\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{x},\mathbf{z},\{t_i\}_{i=0}^{N+1}} \left[\left\| \left(\boldsymbol{F}_{\boldsymbol{\theta}}(\mathbf{x}_t, t, t_{N+1}) - \boldsymbol{F}_{\boldsymbol{\theta}^-}(\mathbf{x}_t, t, t_{N+1}) + \xi(\mathbf{x}_t, \{t_i\}_{i=0}^{N+1}) \right) \right\|_2^2 \right], \quad (9)$$

where the target item is $\xi(\mathbf{x}_t, \{t_i\}_{i=0}^{N+1}) := \frac{1}{\Delta t} \left[\boldsymbol{f}_{\boldsymbol{\theta}^-}(\mathbf{x}_t, t_{N+1}) - \sum_{i=1}^{N} \boldsymbol{f}_{\boldsymbol{\theta}^-}(\mathbf{x}_{t_i}, t_{i+1}) \right] - \frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t}$.

[1] This is a significant advantage over conventional few-step training methods that often rely on Jacobian-vector products (JVP), which can substantially increase GPU memory consumption (Geng et al., 2025; Lu & Song, 2024). Furthermore, the use of JVP can introduce complex technical challenges when integrating with widely-used architectural optimizations like Flash-Attention (Dao et al., 2022).

4 EXPERIMENTS

This section presents the experimental validation of our proposed methodology, denoted as RCGM. We begin by outlining the experimental setup, including datasets, network architectures, and implementation details. We then present a comprehensive evaluation of RCGM's performance. Theoretically, our approach converges to conventional flow-matching and diffusion models when $N = 0$. Consequently, to rigorously assess the unique contributions of RCGM, our empirical investigation focuses on the regime where $N \geq 1$.

4.1 EXPERIMENTAL SETUP

Datasets. Our primary evaluation is conducted on the ImageNet-1K dataset (Deng et al., 2009), utilizing resolutions of 256×256 and 512×512. This choice aligns with established benchmarks in recent high-fidelity generative modeling literature (Karras et al., 2024; Song et al., 2023). We adopt the data preprocessing pipeline from ADM (Dhariwal & Nichol, 2021) to ensure consistency and comparability with prior work. All experiments are performed in the latent space of pretrained autoencoders, a standard practice for efficient training of large-scale models. Specifically:

(a) For 256×256 images, we leverage widely adopted autoencoders, including the SD-VAE (Rombach et al., 2022) and the VA-VAE (Yao et al., 2025).
(b) For 512×512 images, in addition to the SD-VAE, we employ a DC-AE (Chen et al., 2024c) with a higher compression ratio (*f32c32*) to mitigate computational demands.

Network architectures. We build upon the success of transformer-based architectures for generative modeling. Our core model is a 675M-parameter Diffusion Transformer (DiT) (Peebles & Xie, 2023), a backbone widely employed in SOTA models such as SiT (Ma et al., 2024), Lightening-DiT (Yao et al., 2025), and DDT (Wang et al., 2025a).

Implementation details. Our models are implemented in PyTorch (Paszke, 2019) and trained using the AdamW optimizer (Loshchilov & Hutter, 2017) with $\beta_1 = 0.9$, $\beta_2 = 0.95$, a constant learning rate of 2×10^{-4}, and a batch size of 1024. For the time distribution during training, we follow the exact settings in Sun et al. (2025). To evaluate the quality of generated samples, we adhere to standard protocols established in the literature (Song et al., 2020b; Ho et al., 2020; Lipman et al., 2022; Brock et al., 2018). Our primary metric is the Fr'echet Inception Distance (FID) (Heusel et al., 2017), computed over a standard set of $50,000$ generated samples (FID-50K) against the training set.

4.2 ANALYSIS OF HIGHER-ORDER TRAINING

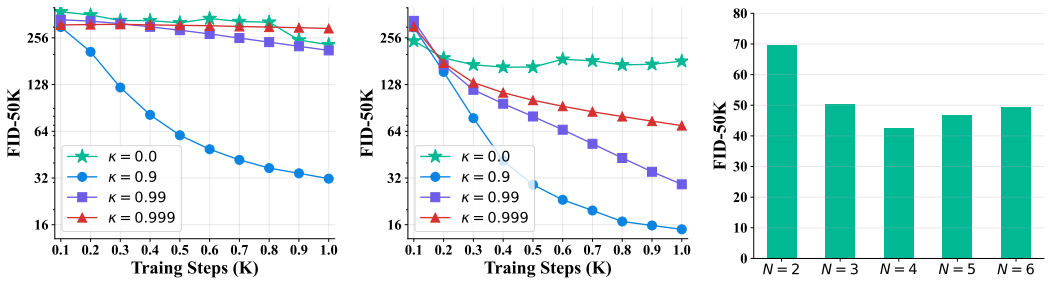
It is widely known that training few-step models is challenging due to the instability of training (Song et al., 2023; Lu & Song, 2024), especially when using a large model and a large learning rate, etc.

This issue is more severe when training few-step models in real-world applications such as high-resolution text-to-image generation.

Using Exponential Moving Average (EMA) model in (8) is a key technique help stabilize training and improve performance, which also evidenced in previous 1st-order methods (Song et al., 2023). For those without using EMA model, they typically require a careful technical design to stabilize training, e.g., using `JVP` (Lu & Song, 2024) or careful hyperparameter design (Song & Dhariwal, 2023).

In this section, to investigate how the order N in RCGM affects the training stability and performance under different EMA decay rates κ, we conduct a series of ablation studies on ImageNet-1K 256×256 using 675M diffusion transformer with SD-VAE.

A large EMA decay rate κ is critical for 1st-order training stability. We first investigate the effect of the EMA decay rate κ from (8) on the stability and performance of the conventional 1st-order ($N = 1$) model. As illustrated in Fig. 3a, training without EMA ($\kappa = 0$) is highly unstable, causing the FID score to fluctuate and fail to converge. A small decay rate ($\kappa = 0.9$) tempers this instability, leading to a smoother decrease in FID, yet the final performance remains suboptimal (FID of 31.70). Conversely, while large decay rates ($\kappa \in \{0.99, 0.999\}$) effectively stabilize training dynamics, they severely hinder convergence. This "over-stabilization" is particularly pronounced at $\kappa = 0.999$,



(a) 1**st-order over training steps.** (b) 2**nd-order over training steps.** (c) **Various orders** ($\kappa = 0.999$)**.**

Figure 3: **Ablation studies of RCGM on ImageNet-1K** 256×256**.** These studies evaluate key factors of the proposed RCGM for training few-step models, i.e., the order of RCGM (N) and the EMA decay rate κ. The sampling is performed using one step (1-NFE).

where the model converges to a poor FID of 294.18. These results reveal a fundamental tension between training stability and model performance in the 1st-order setting.

Higher-order approximations resolve the stability-performance trade-off. We next examine whether higher-order approximations can alleviate the aforementioned tension. Fig. 3b shows the results for our second-order model ($N = 2$). Strikingly, the second-order model thrives under the large EMA decay rates that were detrimental in the 1st-order case. While the no-EMA ($\kappa = 0$) setting remains unstable and the low-EMA ($\kappa = 0.9$) setting achieves a modest FID of 14.94, the high-EMA regime is transformed. With $N = 2$, a large decay rate such as $\kappa = 0.99$ no longer cripples performance but instead yields a competitive FID of 29.13. This demonstrates that higher-order models possess substantially greater robustness to the choice of κ, enabling them to benefit from strong EMA stabilization without sacrificing generative quality.

Further experiments, shown in Fig. 3c, confirm that increasing the order N (with a fixed, high $\kappa = 0.999$) can yield additional gains. Performance steadily improves as N increases from 1 to 4, which achieves the lowest FID. However, this trend reverses for $N > 4$, likely due to the accumulation of approximation errors in the higher-order velocity estimates in (8).

In summary, 1st-order models face a difficult trade-off: a large κ *is needed for stability but harms final performance. Higher-order methods effectively resolve this conflict, achieving both stable convergence and strong performance with large* κ*. Considering the balance between computational cost and performance, we adopt* $N = 2$ *and* $\kappa = 0.999$ *as our default configuration.*

4.3 Comparison with SOTA Few-step Methods

As demonstrated in Tab. 2, our proposed RCGM, when paired with various autoencoders, consistently outperforms or remains highly competitive with SOTA few-step generative models. The following analysis details its advantages across different VAE architectures.

(a) **Performance with SD-VAE** (256×256 **and** 512×512)**:** When paired with a standard SD-VAE, our method exhibits exceptional performance. At 256×256 resolution, it achieves an FID of 1.92 with 2 NFEs, outperforming IMM's best result while requiring 8 times fewer sampling steps. At 512×512 resolution, our model achieves an FID of 2.25 with 2 NFEs, which is highly competitive with specialized distillation models like sCD-L (2.04 FID) and sCD-M (2.26 FID), despite their significantly higher training costs (1434 and 1997 epochs vs. our 360).

(b) **Performance with DC-AE** (512×512)**:** When integrated with the DC-AE autoencoder, our model achieves a new SOTA FID score of **1.79** with only 2 NFEs. This result surpasses the leading consistency distillation model, sCD-XXL, which records an FID of 1.88 at 2 NFEs. Notably, our method achieves this superior image quality using a significantly more efficient model with only 675M parameters, compared to the 1.5B parameters of sCD-XXL.

(c) **Performance with VA-VAE** (256×256)**:** Using the VA-VAE architecture, our method sets another benchmark, achieving a remarkable FID of **1.48** in just 2 NFEs. This represents a substantial improvement over the best-performing distillation method, IMM, which only reaches an FID of 1.99 after a much more costly $8 \times 2 = 16$ NFEs.

Table 2: **System-level quality comparison for few-step generation task on class-conditional ImageNet-1K.** The **best** results of each resolution are highlighted.

512 × 512					256 × 256				
Method	**NFE ↓**	**FID ↓**	**#Params**	**#Epochs**	**Method**	**NFE ↓**	**FID ↓**	**#Params**	**#Epochs**
Diffusion & flow-matching Models									
ADM-G (Dhariwal & Nichol, 2021)	250×2	7.72	559M	388	ADM-G (Dhariwal & Nichol, 2021)	250×2	4.59	559M	396
U-ViT-H/4 (Bao et al., 2023)	50×2	4.05	501M	400	U-ViT-H/2 (Bao et al., 2023)	50×2	2.29	501M	400
DiT-XL/2 (Peebles & Xie, 2023)	250×2	3.04	675M	600	DiT-XL/2 (Peebles & Xie, 2023)	250×2	2.27	675M	1400
SiT-XL/2 (Ma et al., 2024)	250×2	2.62	675M	600	SiT-XL/2 (Ma et al., 2024)	250×2	2.06	675M	1400
MaskDiT (Zheng et al., 2023)	79×2	2.50	736M	-	MDT (Gao et al., 2023)	250×2	1.79	675M	1300
EDM2-S (Karras et al., 2024)	63	2.56	280M	1678	REPA-XL/2 (Yu et al., 2024)	250×2	1.96	675M	200
EDM2-L (Karras et al., 2024)	63	2.06	778M	1476	REPA-XL/2 (Yu et al., 2024)	250×2	1.42	675M	800
EDM2-XXL (Karras et al., 2024)	63	1.91	1.5B	734	Light.DiT (Yao et al., 2025)	250×2	2.11	675M	64
DiT-XL⊕DC-AE	250×2	2.41	675M	400	Light.DiT (Yao et al., 2025)	250×2	1.35	675M	800
GANs									
BigGAN (Brock et al., 2018)	1	8.43	160M	-	BigGAN (Brock et al., 2018)	1	6.95	112M	-
StyleGAN (Sauer et al., 2022)	1×2	**2.41**	168M	-	GigaGAN (Kang et al., 2023)	1	**3.45**	569M	-
Masked & autoregressive models									
MaskGIT (Chang et al., 2022)	12	7.32	227M	300	MaskGIT (Chang et al., 2022)	8	6.18	227M	300
VAR-*d*36-s (Tian et al., 2024)	10×2	**2.63**	2.3B	350	VAR-*d*30-re (Tian et al., 2024)	10×2	**1.73**	2.0B	350
1st-order consistency training & distillation									
sCT-M (Lu & Song, 2024)	1	5.84	498M	1837	Shortcut-XL/2 (Frans et al., 2024)	1	10.6	676M	250
	2	5.53	498M	1837		4	7.80	676M	250
sCT-L (Lu & Song, 2024)	1	5.15	778M	1274	IMM-XL/2 (Zhou et al., 2025)	1×2	7.77	675M	3840
	2	4.65	778M	1274		2×2	5.33	675M	3840
sCT-XXL (Lu & Song, 2024)	1	4.29	1.5B	762		4×2	3.66	675M	3840
	2	3.76	1.5B	762		8×2	2.77	675M	3840
sCD-M (Lu & Song, 2024)	1	2.75	498M	1997	IMM ($\omega = 1.5$)	1×2	8.05	675M	3840
	2	2.26	498M	1997		2×2	3.99	675M	3840
sCD-L (Lu & Song, 2024)	1	2.55	778M	1434		4×2	2.51	675M	3840
	2	2.04	778M	1434		8×2	**1.99**	675M	3840
sCD-XXL (Lu & Song, 2024)	1	2.28	1.5B	921	MeanFlow-XL/2 (Geng et al., 2025)	1	3.43	676M	240
	2	**1.88**	1.5B	921		2	2.93	676M	240
UCGM-XL (Sun et al., 2025)	1	2.63	675M	360	MeanFlow-XL/2 (longer training)	2	2.20	676M	1000
RCGM (Ours)									
⊕SD-VAE (Rombach et al., 2022)	1	2.61	675M	360	⊕SD-VAE (Rombach et al., 2022)	1	2.13	675M	424
⊕SD-VAE	2	2.25	675M	360	⊕SD-VAE	2	1.92	675M	424
⊕DC-AE (Chen et al., 2024c)	1	2.45	675M	800	⊕VA-VAE (Yao et al., 2025)	1	2.25	675M	424
⊕DC-AE	2	**1.79**	675M	800	⊕VA-VAE	2	**1.48**	675M	424

In summary, across multiple autoencoder architectures, our RCGM *consistently delivers a superior trade-off between sample quality, sampling speed, and model parameter efficiency. It establishes new SOTA results while substantially reducing the computational overhead required for high-fidelity image generation.*

Validating RCGM on real-world applications. To assess its practical efficacy, we evaluate RCGM on two demanding real-world tasks: text-to-image generation (App. D.1) and the training of few-step unified multimodal models (App. D.2). Our results demonstrate that RCGM exhibits remarkable performance and versatility across these diverse settings, substantially outperforming existing methods in the computationally constrained, few-step sampling regime.

For instance, in text-to-image synthesis, RCGM attains a GenEval score of 0.85 with only NFE$=2$. This marks a significant advance over the previous SOTA, SANA-Sprint (Chen et al., 2025c), which achieves a score of 0.77, thereby establishing a new benchmark for highly efficient generation.

5 Conclusion and Limitations

In this paper, we introduced RCGM, a unified framework for continuous generative modeling that bridges the gap between multi-step and few-step synthesis. Our key innovation is a novel N-th order flow matching objective that improves training stability and significantly boosts performance, especially in few-step regimes. Through extensive experiments on ImageNet-1K, we demonstrated that RCGM establishes a new state of the art across a spectrum of few-step generation settings.

Despite its strong performance, RCGM shares a limitation with contemporary generative models: achieving high-fidelity synthesis in extreme few-step regimes (e.g., 1-NFE) remains an open challenge, particularly for high-resolution imagery. We conjecture that this is partly attributable to the absence of an adversarial objective, which has proven effective for enhancing perceptual quality in other generative paradigms. Consequently, a promising direction for future research is the integration of adversarial training into the RCGM framework to further push the boundaries of sample quality in this challenging setting. We leave this promising avenue for future work.

Ethics Statement

This research adheres to the *ICLR Code of Ethics* and is committed to the principles of responsible and transparent scientific inquiry. The study involves no human participants, personal or sensitive data, or any activities requiring approval from an institutional ethics review board. All datasets used are publicly accessible under appropriate licenses, with proper attribution given to their original sources. To promote openness and reproducibility, we provide our implementation code and experimental settings for verification and further development by the research community. We also declare that no conflicts of interest or external funding have influenced the design, execution, or presentation of this work.

Reproducibility Statement

Comprehensive details regarding the datasets, model architectures, optimization settings, and training procedures are provided in Sec. 4.1 of the main paper and in App. D. These materials are designed to facilitate the reliable and transparent reproduction of our results. Additionally, our source code will be made publicly available upon acceptance of the paper.

References

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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 22669–22679, 2023.

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

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*, pp. 11315–11325, 2022.

Jiuhai Chen, Zhiyang Xu, Xichen Pan, Yushi Hu, Can Qin, Tom Goldstein, Lifu Huang, Tianyi Zhou, Saining Xie, Silvio Savarese, Le Xue, Caiming Xiong, and Ran Xu. Blip3-o: A Family of Fully Open Unified Multimodal Models-Architecture, Training and Dataset. *arXiv.org*, abs/2505.09568, 2025a.

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*, 2025b.

Junsong Chen, Chongjian Ge, Enze Xie, Yue Wu, Lewei Yao, Xiaozhe Ren, Zhongdao Wang, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-σ: Weak-to-strong training of diffusion transformer for 4k text-to-image generation. *arXiv preprint arXiv:2403.04692*, 2024a.

Junsong Chen, Yue Wu, Simian Luo, Enze Xie, Sayak Paul, Ping Luo, Hang Zhao, and Zhenguo Li. Pixart-{\delta}: Fast and controllable image generation with latent consistency models. *arXiv preprint arXiv:2401.05252*, 2024b.

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

Junying Chen, Zhenyang Cai, Pengcheng Chen, Shunian Chen, Ke Ji, Xidong Wang, Yunjin Yang, and Benyou Wang. Sharegpt-4o-Image: Aligning Multimodal Models with GPT-4o-Level Image Generation. *arXiv.org*, abs/2506.18095, 2025d.

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*, 2024c.

Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*, 2025e.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, 35: 16344–16359, 2022.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, et al. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Runpei Dong, Chunrui Han, Yuang Peng, Zekun Qi, Zheng Ge, Jinrong Yang, Liang Zhao, Jianjian Sun, Hongyu Zhou, Haoran Wei, Xiangwen Kong, Xiangyu Zhang, Kaisheng Ma, and Li Yi. Dreamllm: Synergistic Multimodal Comprehension and Creation. In *The Twelfth International Conference on Learning Representations*, 2024.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024a.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024b.

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. *arXiv preprint arXiv:2410.12557*, 2024.

Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. Masked diffusion transformer is a strong image synthesizer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 23164–23173, 2023.

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

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.

Google. Experiment with gemini 2.0 flash native image generation, 2025a. URL https://developers.googleblog.com/en/experiment-with-gemini-20-flash-native-image-generation/.

Google. Gemini 2.5 flash image: High-consistency image generation and editing, 8 2025b. URL https://aistudio.google.com/models/gemini-2-5-flash-image. Official model page on Google AI Studio. Internal development code name: nano-banana.

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. *Advances in neural information processing systems*, 30, 2017.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

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.

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*, pp. 10124–10134, 2023.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

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*, pp. 24174–24184, 2024.

Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024.

Bin Lin, Zongjian Li, Xinhua Cheng, Yuwei Niu, Yang Ye, Xianyi He, Shenghai Yuan, Wangbo Yu, Shaodong Wang, Yunyang Ge, et al. Uniworld: High-resolution semantic encoders for unified visual understanding and generation. *arXiv preprint arXiv:2506.03147*, 2025.

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

Bingchen Liu, Ehsan Akhgari, Alexander Visheratin, Aleks Kamko, Linmiao Xu, Shivam Shrirao, Joao Souza, Suhail Doshi, and Daiqing Li. Playground v3: Improving text-to-image alignment with deep-fusion large language models. *arXiv preprint arXiv:2409.10695*, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

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

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.

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. In *European Conference on Computer Vision*, pp. 23–40. Springer, 2024.

ModelTC. Qwen-image-lightning. https://github.com/ModelTC/Qwen-Image-Lightning, 2025.

OpenAI. Introducing 4o image generation, 2025. URL https://openai.com/index/introducing-4o-image-generation/.

Xichen Pan, Satya Narayan Shukla, Aashu Singh, Zhuokai Zhao, Shlok Kumar Mishra, Jialiang Wang, Zhiyang Xu, Jiuhai Chen, Kunpeng Li, Felix Juefei-Xu, Ji Hou, and Saining Xie. Transfer between Modalities with MetaQueries. *arXiv.org*, abs/2504.06256, 2025.

A Paszke. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.

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.

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.

Qi Qin, Le Zhuo, Yi Xin, Ruoyi Du, Zhen Li, Bin Fu, Yiting Lu, Xinyue Li, Dongyang Liu, Xiangyang Zhu, Will Beddow, Erwann Millon, Wenhai Wang Victor Perez, Yu Qiao, Bo Zhang, Xiaohong Liu, Hongsheng Li, Chang Xu, and Peng Gao. Lumina-image 2.0: A unified and efficient image generative framework, 2025. URL https://arxiv.org/pdf/2503.21758.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

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

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

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

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

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

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*, 2020b.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

Peng Sun, Yi Jiang, and Tao Lin. Unified continuous generative models. *arXiv preprint arXiv:2505.07447*, 2025.

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.

Fu-Yun Wang, Zhaoyang Huang, Alexander William Bergman, Dazhong Shen, Peng Gao, Michael Lingelbach, Keqiang Sun, Weikang Bian, Guanglu Song, Yu Liu, et al. Phased consistency model. *arXiv preprint arXiv:2405.18407*, 2024.

Shuai Wang, Zhi Tian, Weilin Huang, and Limin Wang. Ddt: Decoupled diffusion transformer. *arXiv preprint arXiv:2504.05741*, 2025a.

Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *Advances in neural information processing systems*, 36:8406–8441, 2023.

Zidong Wang, Yiyuan Zhang, Xiaoyu Yue, Xiangyu Yue, Yangguang Li, Wanli Ouyang, and Lei Bai. Transition models: Rethinking the generative learning objective. 2025b.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report, 2025a. URL https://arxiv.org/abs/2508.02324.

Chenyuan Wu, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, Junjie Zhou, et al. Omnigen2: Exploration to advanced multimodal generation. *arXiv preprint arXiv:2506.18871*, 2025b.

Size Wu, Zhonghua Wu, Zerui Gong, Qingyi Tao, Sheng Jin, Qinyue Li, Wei Li, and Chen Change Loy. Openuni: A simple baseline for unified multimodal understanding and generation. 2025c. URL https://arxiv.org/abs/2505.23661.

Shitao Xiao, Yueze Wang, Junjie Zhou, Huaying Yuan, Xingrun Xing, Ruiran Yan, Shuting Wang, Tiejun Huang, and Zheng Liu. Omnigen: Unified image generation. *arXiv preprint arXiv:2409.11340*, 2024.

Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, et al. Sana: Efficient high-resolution image synthesis with linear diffusion transformers. *arXiv preprint arXiv:2410.10629*, 2024a.

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

Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv preprint arXiv:2408.12528*, 2024b.

Jinheng Xie, Zhenheng Yang, and Mike Zheng Shou. Show-o2: Improved native unified multimodal models. *arXiv preprint arXiv:2506.15564*, 2025b.

Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. *arXiv preprint arXiv:2501.01423*, 2025.

Junyan Ye, Dongzhi Jiang, Zihao Wang, Leqi Zhu, Zhenghao Hu, Zilong Huang, Jun He, Zhiyuan Yan, Jinghua Yu, Hongsheng Li, Conghui He, and Weijia Li. Echo-4o: Harnessing the Power of GPT-4o Synthetic Images for Improved Image Generation. *arXiv*, 2025.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T Freeman. Improved distribution matching distillation for fast image synthesis. *arXiv preprint arXiv:2405.14867*, 2024a.

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

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.

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.

Hongkai Zheng, Weili Nie, Arash Vahdat, and Anima Anandkumar. Fast training of diffusion models with masked transformers. *arXiv preprint arXiv:2306.09305*, 2023.

Kaiwen Zheng, Yongxin Chen, Huayu Chen, Guande He, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. Direct discriminative optimization: Your likelihood-based visual generative model is secretly a gan discriminator. *arXiv preprint arXiv:2503.01103*, 2025.

Linqi Zhou, Stefano Ermon, and Jiaming Song. Inductive moment matching. *arXiv preprint arXiv:2503.07565*, 2025.

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.

CONTENTS

A Utilization of Large Language Models (LLMs)

In this study, Large Language Models (LLMs) are employed at the sentence level to assist in linguistic refinement. Their use was strictly confined to improving grammatical accuracy and overall readability of the manuscript. All research concepts, methodological designs, experimental processes, and analytical findings remain entirely original and have been solely contributed by the authors.

B Implementation Details

Algorithm 1 N-th Order RCGM Training Step

Require: Data distribution $p(\mathbf{x})$, order $N \geq 0$, initial time step Δt (we set $\Delta t = -0.01$ in this paper), EMA decay rate κ.
Require: Initial parameters $\boldsymbol{\theta}$, stop gradient operator sg, EMA parameters $\boldsymbol{\theta}^-$.

Preparation Phase:
Sample data $\mathbf{x} \sim p(\mathbf{x})$, noise $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
Sample current time $t \sim \mathcal{U}(0, 1)$ and target time $t_{N+1} \sim \mathcal{U}(0, t)$.
Construct perturbed state $\mathbf{x}_t = t\mathbf{z} + (1 - t)\mathbf{x}$.
Compute target velocity $\mathbf{v} = \mathbf{z} - \mathbf{x}$.
Define time trajectory: set $t_0 = t$, $t_1 = t_0 + \Delta t$, and let $\{t_i\}_{i=1}^{N+1}$ be linearly spaced between t_1 and t_{N+1}.
Recursive Velocity Estimation:
Initialize accumulated displacement $\boldsymbol{\Delta} = \mathbf{0}$.
Estimate state at t_1: $\mathbf{x}_{t_1} = \mathbf{x}_t + \mathbf{v}\Delta t$.
for $i = 1$ **to** N **do**
    Estimate segment displacement: $\mathbf{d}_i = \boldsymbol{F}_{\boldsymbol{\theta}^-}(\mathbf{x}_{t_i}, t_i, t_{i+1}) \cdot (t_{i+1} - t_i)$.
    Update state: $\mathbf{x}_{t_{i+1}} \leftarrow \mathbf{x}_{t_i} + \mathbf{d}_i$.
    Accumulate: $\boldsymbol{\Delta} \leftarrow \boldsymbol{\Delta} + \mathbf{d}_i$ {*Note: $\boldsymbol{\Delta}$ approximates* $\sum_{i=1}^{N} \int_{t_i}^{t_{i+1}} \mathbf{v}(\mathbf{x}_\tau, \tau) d\tau$}
end for
Loss Calculation:
Compute model output: $\mathbf{u} = \boldsymbol{F}_{\boldsymbol{\theta}}(\mathbf{x}_t, t, t_{N+1})$.
Update $\boldsymbol{\theta}$ by minimizing objective:

$$\mathcal{L}(\boldsymbol{\theta}) = \left\| \mathbf{u} - \text{sg}(\mathbf{u}) - \frac{1}{\Delta t} \left[\text{sg}(\mathbf{u}) \cdot (t - t_{N+1}) + \boldsymbol{\Delta} \right] - \mathbf{v} \right\|_2^2$$

Update EMA parameters: $\boldsymbol{\theta}^- \leftarrow \kappa\boldsymbol{\theta}^- + (1 - \kappa)\boldsymbol{\theta}$.

Our sampling strategy is designed for simplicity and tuning independence. For one-step generation, we rely on the direct prediction procedure in Sec. 2, which requires zero external hyperparameter configuration. For few-step and multi-step generation, we adopt the UCGM sampler (Sun et al., 2025), only using its standard settings without any custom adjustmentTT

B.1 Qualitative Analysis of RCGM Training Dynamics

We investigate the convergence properties of the proposed N-th Order RCGM by modeling the training process as a dynamic interplay between a *Ground-Truth Anchor* and a *Recursive Bootstrap Tail*. The student model aims to match a hybrid target displacement $\mathbf{D}_{\text{target}}$ constructed over the interval $[t, t_{N+1}]$. Based on Alg. 1, this target can be conceptually decomposed as:

$$\mathbf{D}_{\text{target}} \approx \underbrace{\mathbf{v} \cdot \Delta t}_{\substack{\text{Anchor} \\ \text{(Low Bias, High Variance)}}} + \underbrace{\sum_{i=1}^{N} \boldsymbol{F}_{\boldsymbol{\theta}^-}(\mathbf{x}_{t_i}, t_i, t_{i+1}) \cdot (t_{i+1} - t_i)}_{\substack{\text{Recursive Tail} \\ \text{(High Bias, Low Variance)}}} \tag{10}$$

Here, the **Anchor** is derived from the real data distribution (providing unbiased instantaneous direction), while the **Tail** is generated by the EMA teacher $\boldsymbol{\theta}^-$ (providing smooth but potentially

biased manifold approximations). The stability of this system depends critically on the interaction between the recursive order N and the EMA decay rate κ. We identify two distinct operating regimes:

Regime I: The Error Accumulation Regime ($N \geq 2$). When the recursive order N is high, the target trajectory is dominated by the model's own predictions (the Tail). In this regime, the primary challenge is **recursive variance amplification**. Since the estimation at step i depends on the state $\mathbf{x}_{t_i}$ predicted at step $i-1$, small perturbations in the teacher's parameters $\boldsymbol{\theta}^-$ are compounded exponentially through the recursive chain. This phenomenon is analogous to error propagation in numerical integration. If the teacher evolves too quickly (i.e., low κ), the target values become non-stationary and noisy, preventing the student from converging. Consequently, high-order training requires strong stabilization: we must employ a high EMA decay rate (e.g., $\kappa \approx 0.999$) to effectively "freeze" the teacher, minimizing the temporal variance of the target and ensuring that the recursive tail provides a consistent guidance signal.

Regime II: The Bias Correction Regime ($N = 1$). When $N = 1$, the recursive tail is short, and the dominant error source shifts from variance to **geometric bias**. A single-step linear approximation inevitably undershoots the curvature of the true data manifold. If the teacher is too stable (i.e., $\kappa \approx 1$), the model risks converging to a spurious fixed point where the student simply mimics the teacher's biased linear prediction, ignoring the curvature information. Crucially, the **Anchor** term $\mathbf{v} \cdot \Delta t$ contains the necessary first-order derivative information to correct this bias. To effectively incorporate this correction, the system requires "plasticity": the teacher must update rapidly to reflect the rectified trajectory suggested by the Anchor. Thus, low-order training necessitates a lower EMA decay rate (e.g., $\kappa \approx 0.90$). This increases the system's responsiveness, allowing the Anchor to actively correct the Tail's geometric errors and preventing the solidification of incorrect linear assumptions.

Synthesis. Our analysis reveals a fundamental trade-off between **stability** and **plasticity**. High N enables better long-range approximation but introduces instability, necessitating a rigid teacher (High κ). Low N is stable but geometrically biased, necessitating an adaptive teacher (Low κ) to drive correction. This informs our hyperparameter selection strategy: κ should be positively correlated with the recursive order N.

C Related Work

The landscape of continuous-time generative models has evolved from multi-step integration towards high-fidelity, few-step synthesis. Our work builds upon this trajectory by addressing the limitations of existing paradigms. We contextualize our contributions by surveying the two dominant research thrusts that enable rapid generation: interval-based consistency and adversarial refinement.

C.1 Foundations: Multi-Step Integration of Instantaneous Fields

The dominant paradigm in continuous generative modeling, including Denoising Diffusion Models (Ho et al., 2020; Song et al., 2020b) and Flow-Matching (Lipman et al., 2022), is the learning of an *instantaneous* velocity field. These models train a neural network to approximate the local dynamics $\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t}$ of a Probability Flow Ordinary Differential Equation (PF-ODE). To generate a sample, one must numerically integrate this field, typically requiring hundreds or thousands of steps to ensure fidelity. The core limitation of this approach is its sensitivity to coarse discretization; when using few steps, large truncation errors accumulate, particularly for trajectories with high curvature, leading to a significant degradation in sample quality (Karras et al., 2022). This challenge has catalyzed the development of methods designed for the few-step regime.

C.2 Interval-Based Consistency for Few-Step Generation

A major research thrust aims to overcome this limitation by enforcing consistency over finite time intervals, effectively teaching the model about the integrated structure of the ODE path. Consistency Models (CMs) (Song et al., 2023) pioneered this approach by enforcing a *relative* constraint: the model's prediction of the trajectory's endpoint ($\mathbf{x}_0$) should be consistent across different starting points ($\mathbf{x}_t, \mathbf{x}_{t-\Delta t}$) on the same path. This concept was extended by methods like MeanFlow (Geng

et al., 2025), which directly model their proposed average velocity to predict other points beyond the endpoint along the PF-ODE.

However, a critical implementation challenge emerged: the need to compute time derivatives to enforce these interval-based objectives. Early methods relied on Jacobian-Vector Products (JVP) (Geng et al., 2025; Lu & Song, 2024), which introduced a severe scalability bottleneck. JVP is computationally intensive and, more importantly, incompatible with essential modern training optimizations like FlashAttention (Dao et al., 2022) and FSDP-based distributed training (Zhao et al., 2023), hindering its application to billion-parameter models. Consequently, the field has pivoted to using finite-difference estimators as a scalable and hardware-friendly alternative (Sun et al., 2025). These estimators, which rely only on forward passes, ensure compatibility with contemporary large-scale training infrastructures.

C.3 Adversarial Refinement for One-Step Synthesis

A parallel and complementary approach achieves high-fidelity, one-step generation by incorporating external, adversarial signals. This is motivated by the fact that relative consistency constraints do not explicitly guarantee that the final output lies on the true data manifold. Adversarial objectives provide an *absolute* anchor to the data distribution.

Methods in this family, such as distillation techniques like DMD/DMD2 (Yin et al., 2024b;a) and other GAN-based refiners (Sauer et al., 2024b;a; Zheng et al., 2025), employ an auxiliary discriminator to sharpen model outputs. This adversarial pressure can be powerful enough to enable a distilled "student" model to surpass the performance of its "teacher." However, this reliance is a double-edged sword. It often introduces training instability and increases computational overhead due to the auxiliary network. Critically, these frameworks typically depend on a frozen, pre-trained teacher to generate a large dataset of target samples. For ultra-large models, the cost of generating this dataset can be prohibitive, in some cases exceeding the cost of training the student model itself (Yin et al., 2024a). This trade-off between sample fidelity and training complexity remains a key challenge.

D Detailed Experiment

D.1 Comparison with Text-to-image Models

To validate the real-world applicability of our approach, we benchmarked RCGM on the text-to-image synthesis task, presenting detailed results in Tab. 3. For this evaluation, we fine-tuned the SANA-0.6B and SANA-1.6B backbones for $30,000$ steps, using batch sizes of 128 and 64, respectively. The experimental results clearly demonstrate that RCGM achieves SOTA performance while operating with an extremely low NFE. We conduct all experiments on publicly available datasets (Chen et al., 2025d; Ye et al., 2025) and models (Chen et al., 2025c; Xie et al., 2024a) to ensure reproducibility and transparency.

(a) **SOTA quality at 2-NFE:** With the addition of a second inference step, RCGM's generative quality is further enhanced, reaching a GenEval score of 0.85 for the 0.6B model and 0.84 for the 1.6B version. This level of performance surpasses not only the leading few-step models but also powerful multi-step architectures such as SANA-1.5 (0.81) and Playground v3 (0.76). This top-tier output is delivered with a highly competitive throughput of 6.50 samples/s and a latency of just 0.26s.

(b) **Superiority in the 1-NFE setting:** When constrained to a single inference step—a challenging setting for generative models—RCGM markedly outperforms its peers. Our 0.6B variant achieves a GenEval score of 0.80, placing it ahead of strong contenders like SANA-Sprint-1.6B (0.76) and FLUX-Schnell (0.69). Crucially, this high-quality output is paired with unmatched efficiency; at 7.30 samples/s, RCGM-0.6B stands as the fastest model in this category.

The success of RCGM *is especially compelling given its fundamental simplicity. Powerful baselines like SANA-Sprint employ a sophisticated hybrid loss, integrating sCM (Lu & Song, 2024) with LADD (Sauer et al., 2024a)—an adversarial technique requiring a dedicated discriminator. Our approach, however, relies solely on the straightforward objective in* (8)*. The fact that this minimalist framework yields SOTA results demonstrates that* RCGM *offers a more elegant and direct solution to the enduring conflict between sampling speed and visual fidelity.*

Table 3: **System-level benchmark of our RCGM against text-to-image models.** Throughput (batch=10) and latency (batch=1) were measured on a single A100 (BF16). The **best** and second-best results among few-step models are highlighted. †Our evaluation.

Method	NFE ↓	Throughput ↑ (samples/s)	Latency (s) ↓	#Params	GenEval ↑	DPG-Bench ↑
Multi-step models						
SDXL (Podell et al., 2023)	50×2	0.15	6.5	2.6B	0.55	74.7
PixArt-Σ (Chen et al., 2024a)	20×2	0.40	2.7	0.6B	0.54	80.5
SD3-Medium (Esser et al., 2024b)	28×2	0.28	4.4	2.0B	0.62	84.1
FLUX-Dev (Labs, 2024)	50×2	0.04	23.0	12.0B	0.67	84.0
Playground v3 (Liu et al., 2024)	-	0.06	15.0	24B	0.76	87.0
SANA-0.6B (Xie et al., 2024a)	20×2	1.7	0.9	0.6B	0.64	83.6
SANA-1.6B (Xie et al., 2024a)	20×2	1.0	1.2	1.6B	0.66	84.8
SANA-1.5 (Xie et al., 2025a)	20×2	0.26	4.2	4.8B	0.81	84.7
Lumina-Image-2.0 (Qin et al., 2025)	18×2	-	-	2.6B	0.73	87.2
Few-step models						
SDXL-DMD2 (Yin et al., 2024a)	2	2.89	0.40	0.9B	0.58	-
FLUX-Schnell (Labs, 2024)	2	0.92	1.15	12.0B	0.71	-
SANA-Sprint-0.6B (Chen et al., 2025c)	2	6.46	0.25	0.6B	0.76	81.5†
SANA-Sprint-1.6B (Chen et al., 2025c)	2	5.68	0.24	1.6B	0.77	82.1†
SDXL-LCM (Luo et al., 2023)	2	2.89	0.40	0.9B	0.44	-
PixArt-LCM (Chen et al., 2024b)	2	3.52	0.31	0.6B	0.42	-
PCM (Wang et al., 2024)	2	2.62	0.56	0.9B	0.55	-
SD3.5-Turbo (Esser et al., 2024a)	2	1.61	0.68	8.0B	0.53	-
PixArt-DMD (Chen et al., 2024a)	1	4.26	0.25	0.6B	0.45	-
SDXL-DMD2 (Yin et al., 2024a)	1	3.36	0.32	0.9B	0.59	-
FLUX-Schnell (Labs, 2024)	1	1.58	0.68	12.0B	0.69	-
SANA-Sprint-0.6B (Chen et al., 2025c)	1	7.22	0.21	0.6B	0.72	78.6†
SANA-Sprint-1.6B (Chen et al., 2025c)	1	6.71	0.21	1.6B	0.76	80.1†
SDXL-LCM (Luo et al., 2023)	1	3.36	0.32	0.9B	0.28	-
PixArt-LCM (Chen et al., 2024b)	1	4.26	0.25	0.6B	0.41	-
PCM (Wang et al., 2024)	1	3.16	0.40	0.9B	0.42	-
SD3.5-Turbo (Esser et al., 2024a)	1	2.48	0.45	8.0B	0.51	-
TiM (Wang et al., 2025b)	1	-	-	0.8B	0.67	75.0
RCGM-0.6B (Ours)	2	6.50	0.26	0.6B	**0.85**	80.3
RCGM-1.6B (Ours)	2	5.71	0.25	1.6B	**0.84**	79.1
RCGM-0.6B (Ours)	1	7.30	0.23	0.6B	**0.80**	77.2
RCGM-1.6B (Ours)	1	6.75	0.22	1.6B	**0.78**	76.5

D.2 Comparison with Unified Multimodal Models

The development of Unified Multimodal Models (UMM), which are capable of both profound comprehension (typically yielding textual outputs) and sophisticated generation (resulting in visual outputs), represents a significant frontier in artificial intelligence. Such integrated systems hold the potential to unlock synergistic capabilities, where understanding informs generation and vice versa, leading to more intelligent and versatile applications (Google, 2025a;b; OpenAI, 2025).

Recent advancements in UMMs have showcased their considerable potential across a diverse range of applications, including high-fidelity text-to-image generation and intricate image editing (Wu et al., 2025a; Pan et al., 2025). These models have been lauded within the research community for their powerful generative abilities (Chen et al., 2025a; Dong et al., 2024).

However, a primary obstacle to the widespread adoption of these models is their prohibitive computational cost. This inefficiency stems from their reliance on iterative, diffusion-based generation processes, which incur significant overhead and lead to slow inference times. To address this critical efficiency bottleneck, we integrate our proposed method, RCGM, into SOTA UMMs.

We demonstrate this by fine-tuning two prominent open-source models: first, conducting full-parameter tuning on OpenUni-L-512 (Wu et al., 2025c) for $60,000$ steps with a batch size of 128; and second, applying parameter-efficient LoRA (Hu et al., 2022) tuning (with $r = 64$ and $\alpha = 64$) to Qwen-Image-20B Wu et al. (2025a) for $7,000$ steps with a batch size of 64. The experimental results presented in Tab. 4 clearly demonstrate our method's efficacy. Specifically, we observe the following key outcomes:

(a) **Significant reduction in computational cost:** Our method dramatically reduces the NFE to just 1 or 2, a stark contrast to the 40 to 100 NFEs required by the original models. This represents a

Table 4: **System-level comparison of RCGM with unified multimodal models on generation tasks.** Results compare inference efficiency (NFE) and performance across three benchmarks. Best and second-best scores are highlighted as **bold** and underline, respectively. † indicates results using LLM-rewritten prompts on GenEval. All our experiments were conducted on 8× NVIDIA H800 GPUs.

Method	NFE ↓	Image Generation		
		GenEval ↑	DPG-Bench ↑	WISE ↑
Show-o2-7B (Xie et al., 2025b)	50×2	0.76	86.14	0.39
OmniGen (Xiao et al., 2024)	50×2	0.70	81.16	-
OmniGen2 (Wu et al., 2025b)	50×2	0.80 / 0.86†	83.57	-
Show-o (Xie et al., 2024b)	50×2	0.68	67.27	0.35
Janus-Pro (Chen et al., 2025e)	-	0.80	84.19	0.35
MetaQuery-XL (Pan et al., 2025)	30×2	0.78 / 0.80†	81.10	0.55
BLIP3-o-8B (Chen et al., 2025b)	30×2 + 50×2	0.84	81.60	**0.62**
UniWorld-V1 (Lin et al., 2025)	28×2	0.80 / 0.84†	-	0.55
OpenUni-L-512 (Wu et al., 2025c)	20×2	0.85	81.54	0.52
Bagel (Deng et al., 2025)	50×2	0.82 / 0.88†		0.52
Qwen-Image-20B (Wu et al., 2025a)	50×2	**0.87**	**88.32**	**0.62**
OpenUni-L-512⊕CM (Song et al., 2023) (model collapse)	2	0.0	-	-
OpenUni-L-512⊕CM (Song et al., 2023) (model collapse)	1	0.0	-	-
Qwen-Image-20B⊕CM (Song et al., 2023) (model collapse)	2	0.0	-	-
Qwen-Image-20B⊕CM (Song et al., 2023) (model collapse)	1	0.0	-	-
Qwen-Image-20B⊕MeanFlow (Geng et al., 2025) (out of memory)	2	-	-	-
Qwen-Image-20B⊕MeanFlow (Geng et al., 2025) (out of memory)	1	-	-	-
OpenUni-L-512⊕RCGM (ours)	2	0.85	80.15	0.50
OpenUni-L-512⊕RCGM (ours)	1	0.80	76.40	0.45
Qwen-Image-20B⊕RCGM (ours)	8	**0.87**	87.39	0.58
Qwen-Image-20B⊕RCGM (ours)	2	0.82	84.09	0.50
Qwen-Image-20B⊕RCGM (ours)	1	0.52	59.50	0.30

reduction of over 95% in computational workload, thereby enabling faster and more efficient image generation.

(b) **Competitive performance with fewer steps:** When applied to *OpenUni-L-512*, our method with an NFE of 2 achieves a GenEval score of 0.85, matching the performance of the original model which requires 40 steps. While there is a slight decrease in the DPG-Bench and WISE scores, the performance remains highly competitive. Even with a single step (NFE=1), our model maintains a strong GenEval score of 0.80.

(c) **Effective application to larger models:** With the more powerful *Qwen-Image-20B*, our method at 2-NFE achieves a GenEval score of 0.82 and a DPG-Bench score of 84.09. Although these scores are slightly lower than the original model's 100-NFE process, they are still comparable to other leading UMMs that require significantly more computational resources. This demonstrates the scalability and effectiveness of our approach on larger, more capable models.

In summary, our proposed method provides a compelling solution to the efficiency challenges inherent in diffusion-based UMMs. By substantially decreasing the required number of generation steps while preserving a high level of performance, RCGM paves the way for more practical and accessible applications of these powerful multimodal systems.

Discussion on open-source community efforts. To the best of our knowledge, Qwen-Image-Lightning (ModelTC, 2025) represents the sole open-source initiative focused on training a few-step variant of a large-scale UMM. This method is based on the Distribution Matching Distillation (DMD2) framework (Yin et al., 2024a); however, it notably omits its generative adversarial network (GAN) loss component. This crucial omission, however, directly leads to a significant and widely acknowledged problem: **generation pattern collapse**. Specifically, Qwen-Image-Lightning is known to suffer from generating highly similar, or even nearly identical, images across diverse input prompts, severely limiting its generative diversity and overall practical utility.

D.3 Quantitative Analysis on Qwen-Image-20B

Tab. 5 presents a comprehensive quantitative comparison of RCGM against SOTA acceleration methods on the Qwen-Image-20B model. The experiments were conducted on 8 GPUs with a local batch size of 4. Unless otherwise specified, models were trained for 3,000 steps with an AdamW

Table 5: **Quantitative comparison of RCGM against baselines on Qwen-Image-20B for text-to-image generation.** *raw* denotes full-parameter implementations for the generator and scoring networks, which incur excessive GPU memory costs. Consequently, to enable baseline methods (VSD, SiD, and DMD) to run within memory constraints, we implement their fake score networks using LoRA (r=64).

Method	NFE ↓	Image Generation		
		GenEval ↑	DPG-Bench ↑	WISE ↑
VSD (Wang et al., 2023) (*raw*, out of memory)	N/A	-	-	-
DMD (Yin et al., 2024b) (*raw*, out of memory)	N/A	-	-	-
DMD2 (Yin et al., 2024a) (*raw*, out of memory)	N/A	-	-	-
SiD (Zhou et al., 2024) (*raw*, out of memory)	N/A	-	-	-
VSD (Wang et al., 2023) ⊕LoRA	1	0.67	84.44	0.22
VSD (Wang et al., 2023) ⊕LoRA	2	0.73	86.16	0.34
DMD (Yin et al., 2024b) ⊕LoRA	1	0.81	84.31	0.47
DMD (Yin et al., 2024b) ⊕LoRA	2	0.80	84.08	0.46
SiD (Zhou et al., 2024) ⊕LoRA	1	0.77	87.05	0.42
SiD (Zhou et al., 2024) ⊕LoRA	2	0.78	86.94	0.41
RCGM (N=1)	1	0.47	74.31	0.19
RCGM (N=1)	2	0.71	81.48	0.42
RCGM (N=2)	1	0.55	75.56	0.33
RCGM (N=2)	2	0.78	85.27	0.51
RCGM (N=2)	4	0.82	85.13	0.53
RCGM (N=2)	8	0.86	86.49	0.57
RCGM (N=2)	16	0.87	87.11	0.59
RCGM (N=2)	32	0.89	87.68	0.61
RCGM (N=2) ⊕Longer Training	1	0.58	78.57	0.35
RCGM (N=2) ⊕Longer Training	4	0.86	85.71	0.55
RCGM (N=2) ⊕Longer Training	16	0.88	87.10	0.60
RCGM (N=3)	1	0.57	76.85	0.35
RCGM (N=3)	2	0.80	86.48	0.53
RCGM (N=1) ⊕Adversarial Loss	1	0.76	83.41	0.42
RCGM (N=1) ⊕Adversarial Loss	2	0.81	84.75	0.47
RCGM (N=2) ⊕Adversarial Loss	1	0.83	83.82	0.49
RCGM (N=2) ⊕Adversarial Loss	2	0.85	85.46	0.53
RCGM (N=3) ⊕Adversarial Loss	1	0.85	85.12	0.52
RCGM (N=3) ⊕Adversarial Loss	2	0.87	86.37	0.53

learning rate of 1×10^{-5}. Our analysis highlights three key observations regarding scalability, order capability, and compatibility with adversarial objectives.

Scalability to Full-Parameter Training on 20B Models. A significant challenge in accelerating large-scale diffusion models is the excessive memory consumption and training instability associated with Consistency Model (CM) based approaches. As shown in the top section of Tab. 5, baseline methods such as VSD, DMD, and SiD fail to run in the "raw" full-parameter setting due to Out-Of-Memory (OOM) errors, forcing them to rely on low-rank approximations (LoRA, $r = 64$) to fit within memory constraints. In contrast, RCGM demonstrates superior memory efficiency and stability, successfully enabling **full-parameter training** on the 20B parameter model. This result underscores the robustness of our method, overcoming the well-known instability issues often plaguing CM-series methods in high-parameter regimes.

Impact of Higher-Order Trajectory Approximation. We observe a clear performance correlation with the approximation order N. While the first-order variant ($N = 1$) provides a baseline acceleration, increasing the order to $N = 2$ yields a substantial performance leap. Specifically, RCGM ($N = 2$) significantly outperforms the first-order counterpart across all metrics. For instance, at NFE=2, the second-order model achieves a GenEval score of 0.78 compared to 0.71 for $N = 1$. Furthermore, the performance of the second-order model scales effectively with inference steps, reaching a GenEval score of 0.89 at NFE=32. Extending training to 5,000 steps (*Longer Training*)

further improves the one-step generation quality, raising the GenEval score from 0.55 to 0.58 for the $N = 2$ configuration.

Orthogonality with Adversarial Objectives. Finally, our method is highly compatible with adversarial training techniques. We integrate a DMD-like adversarial loss (specifically excluding the regression loss component) to further refine the generation quality. As evidenced in the bottom section of Tab. 5, the addition of the adversarial loss significantly boosts performance across all orders, particularly in the one-step regime. For the $N = 2$ setting, adding the adversarial loss increases the 1-step GenEval score from 0.55 to 0.83, surpassing the strongest LoRA-based baselines (e.g., DMD-LoRA at 0.81). This demonstrates that RCGM provides a robust consistency backbone that can be effectively combined with distribution-matching objectives for SOTA performance.

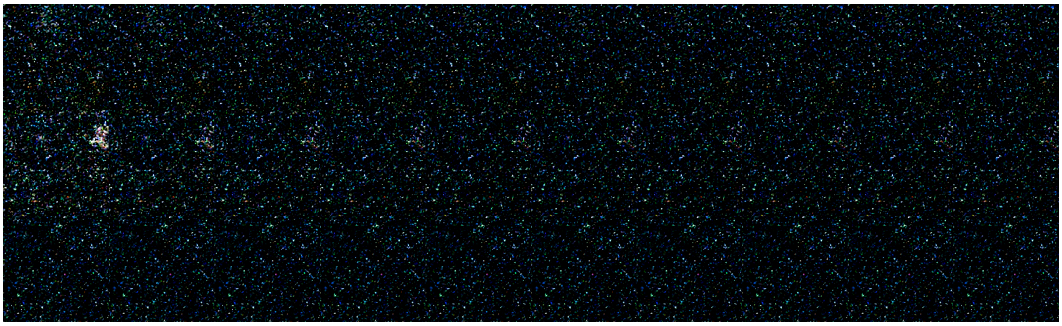


Figure 4: **Visualization of image generation using a collapsed CM (Song et al., 2023) on Qwen-Image-20B (2,000 training steps).** Columns represent varying Numbers of Function Evaluations (NFEs) ranging from 0 to 20 (left to right), while rows correspond to three distinct prompts.

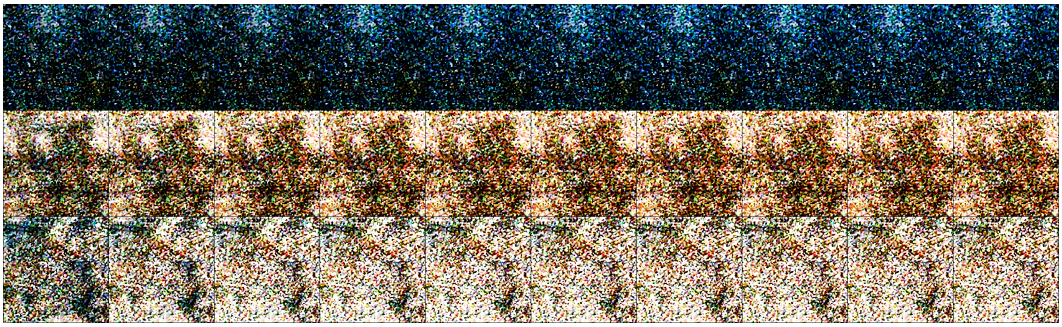


Figure 5: **Visualization of image generation using a collapsed CM (Song et al., 2023) on Qwen-Image-20B (4,000 training steps).** The layout configuration (NFEs and prompts) is identical to Fig. 4.

E Theoretical Analysis

E.1 Main Results

E.1.1 A Recursive Learning Perspective of Consistency Models

The consistency model training objective enforces self-consistency along the sampling trajectory. Given the parameterization $\boldsymbol{f}^{\mathbf{x}}(\boldsymbol{F}_t, \mathbf{x}_t, t) := \frac{\alpha(t)\cdot\boldsymbol{F}_t - \hat{\alpha}(t)\cdot\mathbf{x}_t}{\alpha(t)\cdot\hat{\gamma}(t) - \hat{\alpha}(t)\cdot\gamma(t)}$, the objective is formulated as:

$$\mathbb{E}_{\mathbf{x}_t,t}\left[d\left(\boldsymbol{f}^{\mathbf{x}}(\boldsymbol{F}_t, \mathbf{x}_t, t), \text{stopgrad}(\boldsymbol{f}^{\mathbf{x}}(\boldsymbol{F}_{t-\Delta t}, \mathbf{x}_{t-\Delta t}, t-\Delta t))\right)\right].$$

We focus on the specific case of flow matching (Lipman et al., 2022), where $\alpha(t) = t$, $\gamma(t) = 1 - t$, $\hat{\alpha}(t) = 1$, and $\hat{\gamma}(t) = -1$.

Under these conditions, the training loss $\mathcal{L}_{\text{CM}}(\boldsymbol{\theta})$ simplifies to:

$$\mathcal{L}_{\text{CM}}(\boldsymbol{\theta}) = d\left(t \cdot \boldsymbol{F}_{\boldsymbol{\theta}^-}(\mathbf{x}_t) - \mathbf{x}_t, (t-\Delta t)\cdot \boldsymbol{F}_{\boldsymbol{\theta}^-}(\mathbf{x}_{t-\Delta t}) - \mathbf{x}_{t-\Delta t}\right). \tag{11}$$

This objective minimizes the distance between the current model prediction and the target prediction derived from the preceding time step.

We can express the ℓ-2 loss explicitly:

$$\mathcal{L}_{\mathrm{CM}}(\boldsymbol{\theta}) = \|t \cdot \boldsymbol{F}_{\boldsymbol{\theta}^-}(\mathbf{x}_t) - \mathbf{x}_t - (t-\Delta t) \cdot \boldsymbol{F}_{\boldsymbol{\theta}^-}(\mathbf{x}_{t-\Delta t}) + \mathbf{x}_{t-\Delta t}\|_2^2 \,. \tag{12}$$

To analyze the training dynamics, we consider the limit $\Delta t \to 0$ and apply a Taylor expansion, which yields:

$$\mathcal{L}_{\mathrm{CM}}(\boldsymbol{\theta}) = \left\| t \cdot \boldsymbol{F}_{\boldsymbol{\theta}^-}(\mathbf{x}_t) - (t-\Delta t) \cdot \boldsymbol{F}_{\boldsymbol{\theta}^-}(\mathbf{x}_{t-\Delta t}) - \frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} \cdot \Delta t \right\|_2^2 . \tag{13}$$

Minimizing this loss corresponds to the following update rule:

$$t \cdot \boldsymbol{F}_{\boldsymbol{\theta}^-}(\mathbf{x}_t) \leftarrow (t-\Delta t) \cdot \boldsymbol{F}_{\boldsymbol{\theta}^-}(\mathbf{x}_{t-\Delta t}) + \frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} \cdot \Delta t \,. \tag{14}$$

By induction, the model learns the integrated velocity field:

$$t \cdot \boldsymbol{F}_{\boldsymbol{\theta}^-}(\mathbf{x}_t) \leftarrow \int_0^t \frac{\mathrm{d}\mathbf{x}_\tau}{\mathrm{d}\tau} \mathrm{d}\tau = \mathbf{x}_t - \mathbf{x}_0 \,. \tag{15}$$

Let us define the prediction function as $\boldsymbol{f}(\mathbf{x}_t, 0) := \mathbf{x}_0 - \mathbf{x}_t$. Substituting this into (14), we obtain the following recursive relationship:

$$\boxed{\boldsymbol{f}(\mathbf{x}_t, 0) \leftarrow \boldsymbol{f}(\mathbf{x}_{t-\Delta t}, 0) - \frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} \cdot \Delta t} \,. \tag{16}$$

This result confirms that the consistency model training objective is equivalent to recursively learning the velocity field of the underlying ODE.

E.2 Error Bound Analysis of Recursive Learning Objective

Definition 1 (Discrete operator) . *Let $g_t := \frac{\mathrm{d}x_t}{\mathrm{d}t}$ denote the true temporal derivative, $t = t_0 > t_1 > \cdots > t_N > t_{N+1} = 0$ be the time points, and $\Delta t = t_1 - t_0$ be the time step. We define the discrete operator*

$$A_\theta := \frac{1}{\Delta t}\Big(f_\theta(x_t, t_{N+1}) - \sum_{i=1}^{N} f_{\theta^-}(x_{t_i}, t_{i+1})\Big), \tag{17}$$

and the analogous operator with the ground-truth function f^*,

$$A^* := \frac{1}{\Delta t}\Big(f^*(x_t, t_{N+1}) - \sum_{i=1}^{N} f^*(x_{t_i}, t_{i+1})\Big). \tag{18}$$

Lemma 1 (Numerical integration error) . *Let $f \in C^1([a,b])$, then we have:*

$$\int_a^b f(x)\mathrm{d}x = f(a)(b-a) + \frac{f'(\xi)}{2}(b-a)^2 \,, \quad \textit{for some } \xi \in (a,b) \tag{19}$$

Proof. By the Taylor's theorem, we have:

$$f(x) = f(a) + f'(\xi)(x-a) \,, \quad \text{for some } \xi \in (a,x) \tag{20}$$

Therefore, we have:

$$\int_a^b f(x)\mathrm{d}x = \int_a^b f(a)\mathrm{d}x + \int_a^b f'(\xi)(x-a)\mathrm{d}x \tag{21}$$

$$= f(a)(b-a) + \frac{f'(a)}{2}(b-a)^2 \tag{22}$$

$$= f(a)(b-a) + \frac{f'(\xi)}{2}(b-a)^2 \tag{23}$$

□

Lemma 2 (Trunction error) . *Let's assume that the trajectories* $\mathbf{x}_t \in C^2[0,1]$ *and*

$$f^\star(x_r, t) = \int_r^t \frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t}\mathrm{d}t = \mathbf{x}_t - \mathbf{x}_r$$

For $t = t_0 > t_1 > \cdots > t_N > t_{N+1} = 0$*, the following equality holds:*

$$\left\| \frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} - A^\star \right\|_2^2 \leq C_1 \cdot (t_0 - t_1)^2 \tag{24}$$

where $C_1 = \sup_t \left\| \frac{1}{2}\frac{\mathrm{d}^2\mathbf{x}_t}{\mathrm{d}t^2} \right\|_2^2$.

Proof.

$$\mathbf{x}_{t_{N+1}} - \mathbf{x}_{t_0} = \sum_{i=0}^{N}(\mathbf{x}_{t_{i+1}} - \mathbf{x}_{t_i}) = \sum_{i=0}^{N}\int_{t_i}^{t_{i+1}} \frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t}\mathrm{d}t = \sum_{i=0}^{N} f^\star(\mathbf{x}_{t_i}, t_{i+1}) \tag{25}$$

$$= f^\star(x_{t_0}, t_1) + \sum_{i=1}^{N} f^\star(\mathbf{x}_{t_i}, t_{i+1}) = \int_{t_0}^{t_1} \frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t}\mathrm{d}t + \sum_{i=1}^{N}\int_{t_i}^{t_{i+1}} \frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t}\mathrm{d}t \tag{26}$$

$$= \frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t}|_{t=t_0}(t_1 - t_0) + \frac{1}{2}\frac{\mathrm{d}^2\mathbf{x}_t}{\mathrm{d}t^2}|_{t=\xi}(t_1 - t_0)^2 \tag{27}$$

$$+ \sum_{i=1}^{N}\int_{t_i}^{t_{i+1}} \frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t}\mathrm{d}t\,, \quad \text{for some } \xi \in (t_1, t_0) \tag{28}$$

$$= \frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t}|_{t=t_0}(t_1 - t_0) + \frac{1}{2}\frac{\mathrm{d}^2\mathbf{x}_t}{\mathrm{d}t^2}|_{t=\xi}(t_1 - t_0)^2 + \sum_{i=1}^{N}\int_{t_i}^{t_{i+1}} \frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t}\mathrm{d}t \tag{29}$$

$$\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t}|_{t=t_0} = \frac{1}{t_1 - t_0}\left[(\mathbf{x}_{t_{N+1}} - \mathbf{x}_{t_0}) - \sum_{i=1}^{N} f^\star(\mathbf{x}_{t_i}, t_{i+1})\right] - \frac{1}{2}\frac{\mathrm{d}^2\mathbf{x}_t}{\mathrm{d}t^2}|_{t=\xi}(t_1 - t_0) \tag{30}$$

$$= \frac{1}{t_1 - t_0}\left[f^\star(\mathbf{x}_{t_0}, t_{N+1}) - \sum_{i=1}^{N} f^\star(\mathbf{x}_{t_i}, t_{i+1})\right] - \frac{1}{2}\frac{\mathrm{d}^2\mathbf{x}_t}{\mathrm{d}t^2}|_{t=\xi}(t_1 - t_0) \tag{31}$$

Therefore,

$$\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t}|_{t=t_0} - A^\star = -\frac{1}{2}\frac{\mathrm{d}^2\mathbf{x}_t}{\mathrm{d}t^2}|_{t=\xi}(t_1 - t_0)$$

□

Lemma 3 (Approximation error) . *Define the approximation error as:*

$$\varepsilon_{t_i,t_j} = f^*(x_{t_i}, t_j) - f_\theta(x_{t_i}, t_j)$$

and assume that the approximation error is uniformly bounded by ε*, i.e.,*

$$\sup_{x,t} \left\| f^*(x,t) - f_\theta(x,t) \right\|_2^2 \leq \varepsilon^2$$

Then we have:

$$\left\| A^\star - A_\theta \right\|_2^2 \leq \frac{(N+1)^2}{(\Delta t)^2} \cdot \varepsilon^2$$

Proof.

$$\|A^\star - A_\theta\|_2^2 \tag{32}$$

$$= \|\frac{1}{\Delta t}\Big([f^*(x_t, t_{N+1}) - f_\theta(x_t, t_{N+1})] - \sum_{i=1}^{N}[f^*(x_{t_i}, t_{i+1}) - f_{\theta^-}(x_{t_i}, t_{i+1})]\Big)\|_2^2 \tag{33}$$

$$= \frac{1}{(\Delta t)^2}\|[f^*(x_t, t_{N+1}) - f_\theta(x_t, t_{N+1})] - \sum_{i=1}^{N}[f^*(x_{t_i}, t_{i+1}) - f_{\theta^-}(x_{t_i}, t_{i+1})]\|_2^2 \tag{34}$$

$$= \frac{1}{(\Delta t)^2}\|\varepsilon_{t_0,t_{N+1}} - \sum_{i=1}^{N}\varepsilon_{t_i,t_{i+1}}\|_2^2 \tag{35}$$

$$\leq \frac{N+1}{(\Delta t)^2}\left(\|\varepsilon_{t_0,t_{N+1}}\|_2^2 + \sum_{i=1}^{N}\|\varepsilon_{t_i,t_{i+1}}\|_2^2\right) \quad \text{(By Cauchy-Schwarz inequality)}$$

$$\leq \frac{(N+1)^2}{(\Delta t)^2}\cdot\varepsilon^2 \tag{36}$$

□

Theorem 1 (Error bound) . *Let's assume that the trajectory* $\mathbf{x}_t \in C^2[0,1]$,

$$f^\star(x_r, t) = \int_r^t \frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t}\mathrm{d}t = \mathbf{x}_t - \mathbf{x}_r$$

and the approximation error is uniformly bounded by ε*, i.e.,*

$$\sup_{x,t}\|f^*(x,t) - f_\theta(x,t)\|_2^2 \leq \varepsilon^2$$

For $t = t_0 > t_1 > \cdots > t_N > t_{N+1} = 0$*, the following inequality holds:*

$$\mathbb{E}_{\mathbf{x}_0,\mathbf{z},\{t_i\}_{i=0}^{N+1}}\left[\|\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} - A_\theta\|_2^2\right] \leq \mathbb{E}_{\mathbf{x}_0,\mathbf{z},\{t_i\}_{i=0}^{N+1}}\left[2C_1\cdot(t_0-t_1)^2 + \frac{2(N+1)^2}{(t_0-t_1)^2}\cdot\varepsilon^2\right] \tag{37}$$

where $C_1 = \sup_t\|\frac{1}{2}\frac{\mathrm{d}^2\mathbf{x}_t}{\mathrm{d}t^2}\|_2^2$*, and if* $t \in \mathcal{U}[\delta, 1]$*, then the upper bound can attain the minimum value when setting* $N+1 = \lfloor\sqrt[6]{\frac{C_1\cdot\delta\cdot(1+\delta+\delta^2)}{6\cdot\varepsilon^2}}\rfloor$.

Proof. By Lem. 2 and Lem. 3, we have:

$$\begin{aligned}
&\mathbb{E}_{\mathbf{x}_0,\mathbf{z},\{t_i\}_{i=0}^{N+1}}\left[\|\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} - A_\theta\|_2^2\right] \\
&\leq \mathbb{E}_{\mathbf{x}_0,\mathbf{z},\{t_i\}_{i=0}^{N+1}}\left[2\|\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} - A^\star\|_2^2\right] + \mathbb{E}_{\mathbf{x}_0,\mathbf{z},\{t_i\}_{i=0}^{N+1}}\left[2\|A^\star - A_\theta\|_2^2\right] \\
&\leq \mathbb{E}_{\mathbf{x}_0,\mathbf{z},\{t_i\}_{i=0}^{N+1}}\left[2C_1\cdot(t_0-t_1)^2 + \frac{2(N+1)^2}{(t_0-t_1)^2}\cdot\varepsilon^2\right]
\end{aligned} \tag{38}$$

If we set $t_k = t - \frac{k}{N+1}\cdot t$, then $t_0 - t_1 = \frac{t}{N+1}$. Therefore, the equation (38) becomes:

$$\mathbb{E}_{\mathbf{x}_0,\mathbf{z},\{t_i\}_{i=0}^{N+1}}\left[2C_1\cdot(t_0-t_1)^2 + \frac{2(N+1)^2}{(t_0-t_1)^2}\cdot\varepsilon^2\right] \tag{39}$$

$$= \mathbb{E}_t\left[2C_1\cdot\frac{t^2}{(N+1)^2} + \frac{2(N+1)^2}{\frac{t^2}{(N+1)^2}}\cdot\varepsilon^2\right] \tag{40}$$

$$= \mathbb{E}_t\left[2C_1\cdot\frac{t^2}{(N+1)^2} + \frac{2(N+1)^4}{t^2}\cdot\varepsilon^2\right] \tag{41}$$

Let's consider $t \sim U(\delta, 1)$, then we have:

$$\mathbb{E}\left[t^2\right] = \int_{\delta}^{1} t^2 \cdot \frac{1}{1-\delta} \mathrm{d}t = \frac{1}{3(1-\delta)} \cdot (1-\delta^3) = \frac{1}{3}(1+\delta+\delta^2) \tag{42}$$

$$\mathbb{E}\left[\frac{1}{t^2}\right] = \int_{\delta}^{1} \frac{1}{t^2} \cdot \frac{1}{1-\delta} \mathrm{d}t = \frac{1}{1-\delta} \cdot (-1+\frac{1}{\delta}) = \frac{1}{1-\delta} \cdot (\frac{1}{\delta}-1) = \frac{1}{\delta} \tag{43}$$

Therefore, the equation (38) becomes:

$$\mathbb{E}_t\left[2C_1 \cdot \frac{t^2}{(N+1)^2} + \frac{2(N+1)^4}{t^2} \cdot \varepsilon^2\right] \tag{44}$$

$$= 2C_1 \cdot \frac{1+\delta+\delta^2}{3(N+1)^2} + \frac{2(N+1)^4}{\delta} \cdot \varepsilon^2 \tag{45}$$

$$= C_1 \cdot \frac{1+\delta+\delta^2}{3(N+1)^2} + C_1 \cdot \frac{1+\delta+\delta^2}{3(N+1)^2} + \frac{2(N+1)^4}{\delta} \cdot \varepsilon^2 \tag{46}$$

$$\geq 3 \cdot \left(C_1 \cdot \frac{1+\delta+\delta^2}{3(N+1)^2} \cdot C_1 \cdot \frac{1+\delta+\delta^2}{3(N+1)^2} \cdot \frac{2(N+1)^4}{\delta} \cdot \varepsilon^2\right)^{\frac{1}{3}} \tag{47}$$

$$= 3 \cdot \left(C_1^2 \cdot \frac{2(1+\delta+\delta^2)^2}{9\delta} \cdot \varepsilon^2\right)^{\frac{1}{3}} \tag{48}$$

The equality holds when $C_1 \cdot \frac{1+\delta+\delta^2}{3(N+1)^2} = C_1 \cdot \frac{1+\delta+\delta^2}{3(N+1)^2} = \frac{2(N+1)^4}{\delta} \cdot \varepsilon^2$, i.e.

$$(N+1)^6 = \frac{C_1 \cdot \delta \cdot (1+\delta+\delta^2)}{6 \cdot \varepsilon^2} \tag{49}$$

□

Remark 1 . *Thm. 1 shows that the error bound of the recursive learning objective has a minimum value, rather than negatively related to N.*

Corollary 1 (Relationship between ε and N) . *Under the same assumptions as Thm. 1, if the upper bound of the loss*

$$\mathcal{L}(\theta) \leq \mathbb{E}_{\mathbf{x}_0, \mathbf{z}, \{t_i\}_{i=0}^{N+1}}\left[2C_1(t_0-t_1)^2 + \frac{2(N+1)^2}{(t_0-t_1)^2}\varepsilon^2\right]$$

is minimized with respect to N, then the uniform approximation error ε and the number of steps N satisfy the relation

$$\varepsilon \approx \sqrt{\frac{C_1\,\delta\,(1+\delta+\delta^2)}{6\,(N+1)^6}}\,,$$

This shows that, for a given number of steps N, the uniform approximation error ε scales roughly as $\varepsilon \sim (N+1)^{-3}$.