Unified Multi-Stage Flow Matching via Conditional Dependent Coupling

Chenrui Ma[1,2] **Xi Xiao**[3,4] **Tianyang Wang**[3] **Xiao Wang**[4] **Yanning Shen**[2*]
[1]University of Virginia [2]University of California, Irvine
[3]University of Alabama at Birmingham [4]Oak Ridge National Laboratory
zzz8fa@virginia.edu {xxiao,tw2}@uab.edu
wangx2@ornl.gov yannings@uci.edu

Abstract

Existing image generation models face critical challenges regarding the trade-off between computation and fidelity. Latent-space models use compact representations to reduce this cost, while pixel-space models learn directly from image values. Multi-stage generation allocates computation across resolutions; its stage targets, source couplings, and parameter sharing determine the refinement tasks learned by the model. To address these challenges, we introduce a unified multi-stage generative framework formulated under the *stochastic interpolant* formalism with our **Conditional Dependent Coupling** strategy as a Flow-Matching variant. It decomposes the generative process into interpolant trajectories at multiple stages, using clean image targets at each resolution while enabling end-to-end optimization. The entire process is modeled as a single unified Diffusion Transformer, enabling knowledge sharing across stages. Under aligned averaging/replication and a decaying noise schedule, the construction has a lower sum of stage-local transport-cost bounds than the independent full-resolution coupling. An evaluation-cost model connects these costs to sampling work. On ImageNet-256, CDC-FM achieves FID 1.97 at 32 total function evaluations and FID 1.81 at 128. A matched single-stage pixel model reaches FID 1.82 at 500 evaluations; measured time falls from 8.73 to 5.18 seconds per image, a 40.7% reduction. ImageNet-512 generation achieves FID 2.65 and IS 307.4.

1 Introduction

Generative models have achieved remarkable progress in recent years, driving advances in diverse domains such as natural language processing [1], computer vision [2], and scientific modeling [3]. Unlike discrete data generation (e.g., large language models for text), high-dimensional continuous data generation, such as image synthesis, suffers from high complexity and computational demands [4–6]. The primary challenge lies in balancing fidelity, efficiency, and scalability when learning to approximate intricate data distributions [7, 8].

There are two main paradigms for generative modeling: pixel space generation and latent space generation. Pixel space generation operates directly in the original data domain, preserving fine-grained details without compression [9, 10]. However, this approach requires substantial training and inference computation because of the high dimensionality of images [4, 11]. In contrast, latent space generation uses compact representations, typically through Variational Autoencoders (VAEs), to reduce dimensionality and improve efficiency [12, 13]. While effective, these methods can incur information loss during encoding and decoding, and the representation is often trained separately from the generative backbone [14, 15].

*Corresponding author: yannings@uci.edu.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

To mitigate these limitations, multi-stage generation methods have been proposed. By decomposing the synthesis process into a sequence of stages, they allow early stages to operate in lower-dimensional spaces and later stages to refine the outputs at higher resolutions [8, 16]. This hierarchical design improves efficiency and progressively captures data complexity [17, 18]. Cascaded models use stage-specific components [19, 20], while Pyramidal Flow and PixelFlow also explore shared parameterization across resolutions [16, 14]. The stage endpoint distributions and the coupling between coarse and fine images determine what each refinement task learns.

In this work, we propose a multi-stage generation framework formulated under the *stochastic interpolant* [21, 22] formalism with **conditional dependent coupling**. We instantiate the framework as a Flow-Matching model, CDC-FM. Our approach generates high-resolution images in a coarse-to-fine manner, organizing the multi-stage generation tasks through paired image views. Specifically, the proposed method enables efficient and high-performance generation directly in pixel space, thus retaining detailed information without the bottleneck of latent compression. Moreover, by proposing **conditional dependent coupling**, we unify the multi-stage generation process into one coherent framework, using the clean data distribution at each resolution as its stage target. Notably, the entire multi-stage framework can be parameterized by a single DiT [23], supporting knowledge sharing across stages and enabling full end-to-end optimization. We analyze the prescribed transport-cost bounds under aligned multiresolution operators and a decaying noise schedule, and compare sampling time against a matched single-stage pixel model. Shared-feature and recursive-source experiments examine how the common model operates across stages. See the Related Works in the Appendix. The project repository is `https://github.com/merry7cherry/Stochastic-Interpolants-via-Conditional-Dependent-Coupling`.

2 Preliminaries

The *stochastic interpolant* [21, 22] unifies the theory of Ordinary Differential Equations (ODEs) and Stochastic Differential Equations (SDEs). Our method uses this formalism to construct coupled image-generation tasks; the evaluated CDC-FM model uses deterministic Flow/ODE dynamics. We use the Flow/ODE formulation below. Appendix §D gives the full stochastic interpolant formalism, including its SDE dynamics.

Definition 2.1 (Deterministic interpolant for Flow Matching)**.** Given two probability densities $\rho_0, \rho_1 : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ and a coupling $\rho(x_0, x_1)$ with marginals ρ_0 and ρ_1, we define the deterministic interpolant [24, 25]:

$$I_t \;=\; \alpha_t x_0 + \beta_t x_1, \qquad t \in [0,1], \tag{1}$$

where α_t, β_t are differentiable in t and satisfy the boundary conditions:

$$\alpha_0 = \beta_1 = 1, \qquad \alpha_1 = \beta_0 = 0, \qquad \alpha_t^2 + \beta_t^2 > 0 \;\; \forall t \in [0,1].$$

This yields a time–dependent density ρ_t for I_t with $\rho_{t=0} = \rho_0$ and $\rho_{t=1} = \rho_1$.

Theorem 2.2 (Transport continuity equation for Flow model)**.** *Let $b_t : \mathbb{R}^d \to \mathbb{R}^d$ denote the* conditional velocity field *[24, 25]:*

$$b_t(x) \;=\; \mathbb{E}\Big[\, \dot{I}_t \,\Big|\, I_t = x \,\Big], \tag{2}$$

where $\dot{f} = \frac{\mathrm{d}f}{\mathrm{d}t}$ and the expectation is over $(x_0, x_1) \sim \rho(x_0, x_1)$. The interpolant density ρ_t solves the transport continuity equation [26, 27]:

$$\partial_t \rho_t(x) \;+\; \nabla \cdot \big(b_t(x)\, \rho_t(x)\big) \;=\; 0. \tag{3}$$

In practice, we learn a model velocity $\hat{b}_t$ to approximate b_t by minimizing:

$$L_b(\hat{b}) \;=\; \int_0^1 \mathbb{E}\Big[\, \big|\hat{b}_t(I_t)\big|^2 \;-\; 2\, \dot{I}_t \cdot \hat{b}_t(I_t) \,\Big]\, \mathrm{d}t, \tag{4}$$

which is estimable from samples $(x_0, x_1) \sim \rho(x_0, x_1)$.

Corollary 2.3 (Probability flow ODE)**.** *The transport continuity equation implies that the solution X_t to the probability flow ODE [28, 6]:*

$$\dot{X}_t \;=\; b_t(X_t) \tag{5}$$

matches the interpolant law under the usual well-posedness conditions: if $X_{t=0} \sim \rho_0$, then $X_{t=1} \sim \rho_1$. Hence generative sampling is obtained by drawing $x_0 \sim \rho_0$ and integrating (5) from t=0 to t=1.

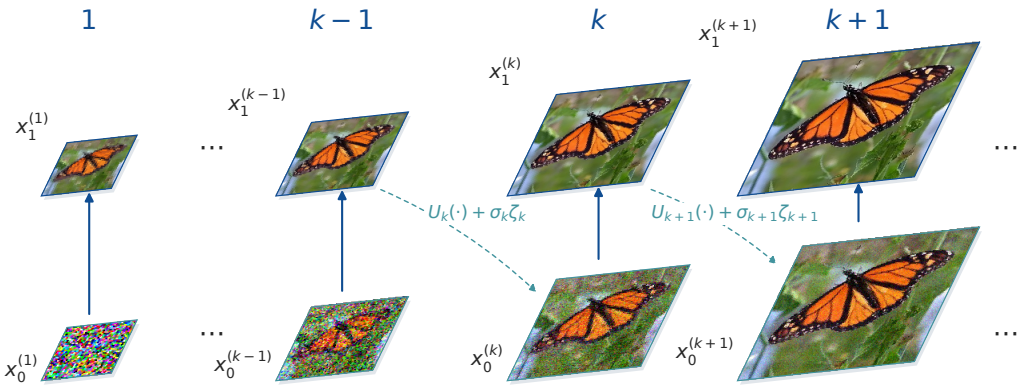

Figure 1: CDC-FM over a K-stage hierarchy. Stage 1 starts from Gaussian noise; the selected refinement stages $k-1, k, k+1$ start from noisy coarse views. Each column illustrates a source x_0 and clean target x_1 at its resolution. Dashed arrows construct the next source; solid arrows indicate the stage-wise interpolant direction. One shared DiT learns all stages. Training uses paired real views; recursive sampling uses the preceding generated endpoint. Images are illustrative.

Remark (Generalization to *stochastic interpolant*). The Appendix §D introduces the full *stochastic interpolant* $I_t = \alpha_t x_0 + \beta_t x_1 + \gamma_t z$ with $z \sim \mathcal{N}(0, I_d)$, [21, 22] which recovers diffusion-type models, the score field $s_t(x) = \nabla \log \rho_t(x)$, and the companion objective for s_t. All transport-equation statements above remain valid, with (2)–(5) appearing as the $\gamma_t \equiv 0$ case used throughout the main text.

Definition 2.4 (Transport cost)**.** Let $X_t(x_0)$ be the solution to the probability flow ODE (5) for the initial condition $X_{t=0}(x_0) = x_0 \sim \rho_0$. Then the following inequality holds:

$$\mathbb{E}_{x_0 \sim \rho_0} \left[|X_{t=1}(x_0) - x_0|^2 \right] \leq \int_0^1 \mathbb{E}[|\dot{I}_t|^2] dt < \infty. \tag{6}$$

The infimum of expected squared endpoint displacement over all couplings is the squared Wasserstein-2 distance [29, 30]. Here we compare bounds induced by specified endpoint couplings.

Independent coupling draws endpoints from $\rho_0(x_0)\rho_1(x_1)$. A **data-dependent coupling** instead defines a joint law $\rho(x_0, x_1) = \rho_0(x_0 \mid x_1)\rho_1(x_1)$ [30]. Its dependence links the source to its paired target and changes the induced velocity and cost bound. Section 3.3 studies the particular coupling supplied by coarse and fine image views.

3 Methodology

In this section, we introduce a unified multi-stage generative algorithm designed to efficiently produce high-resolution images directly in pixel space with end-to-end training. The algorithm proceeds through a sequence of stages, where the resolution is progressively refined in a coarse-to-fine manner. At each stage, the output of the previous stage is further enhanced toward a finer resolution target using *stochastic interpolant* with **conditionally dependent coupling**, thus forming a cascaded architecture with clean image targets at each stage. All stages are parameterized by a single unified DiT [5], which supports knowledge sharing and end-to-end optimization. Figure 1 illustrates the endpoint construction and recursive sampling. We construct the stage trajectories with deterministic interpolants and learn their ODE velocity fields.

3.1 Notations

Data and Resolution Hierarchy. Let the target high-resolution image distribution be defined on $\mathbb{R}^{d_K}$. We define a hierarchy of resolutions, where $k \in \{1, \ldots, K\}$ represents the generation stage: ❶ $x^{(K)} \sim p_{\text{data}}$: A sample from the true high-resolution data distribution. ❷ $x_1^{(k)} \in \mathbb{R}^{d_k}$: A ground-truth image sample at stage k, which is generated by downsampling the original high-resolution image.

Upsampling and Downsampling Operators. We define operators for changing image resolutions. With a fixed channel count and a doubling of image side length at each stage, the ambient dimension is $d_k = 4^{k-1}d_1$.

- *Downsampling Operator* D_k: This operator maps an image from resolution d_k to d_{k-1} ($D_k : \mathbb{R}^{d_k} \to \mathbb{R}^{d_{k-1}}$). It operates via neighborhood averaging, where a block of pixels in the higher-resolution image is averaged to a single pixel in the lower-resolution image.
- *Upsampling Operator* U_k: This operator maps an image from resolution d_{k-1} to d_k ($U_k : \mathbb{R}^{d_{k-1}} \to \mathbb{R}^{d_k}$). It operates via neighborhood replication, where each pixel from the lower-resolution image is copied to form a block in the higher-resolution image.

These operators are defined such that they are transitive: $D_{k+1\to k-1} = D_k \circ D_{k+1}$. We can define a composite downsampling operator $D_{K\to k}$ that maps from the highest resolution d_K directly to d_k. The ground-truth sample for any stage k is thus defined as $x_1^{(k)} = D_{K\to k}(x^{(K)}) \sim \rho_1^{(k)}(x_1^{(k)}|x^{(K)})$.

3.2 Multi-Stage Flow Matching Model

To implement the unified multi-stage generation process, we define a single unified generative model(Flow/ODEs or Diffusion/SDEs) to transform from the source distribution to the target at each stage k with **conditional dependent coupling**. The K stages are designed as below: ❶ **Stage** $k = 1$**: Noise-to-Image Generation.** The first stage generates a low-resolution image from a simple source distribution (e.g., Gaussian noise). ❷ **Stage** $k > 1$**: Image-to-Image Refinement.** In subsequent stages, the resolution is progressively refined using **conditional dependent coupling**, where the probability path is explicitly conditioned on the relationship between the low-resolution source and the high-resolution target images. This cascaded approach breaks down the complex task of high-resolution image generation into a series of more manageable sub-problems. We will elaborate on the **motivation** and **benefits** of the design at the end of this section.

Stage $k = 1$**: Noise-to-Image Generation** The first stage ($k = 1$) of our model is responsible for generating the lowest-resolution image from a simple prior distribution. The model is defined by its source and target distributions and the **conditional dependent coupling** between them. For this foundational stage, the joint distribution is the product of the marginals:

$$\begin{aligned}\rho^{(1)}(x_0^{(1)}, x_1^{(1)}|x^{(K)}) &= \rho_0^{(1)}(x_0^{(1)}|x_1^{(1)})\rho_1^{(1)}(x_1^{(1)}|x^{(K)}) \\ &= \rho_0^{(1)}(x_0^{(1)})\rho_1^{(1)}(x_1^{(1)}|x^{(K)})\end{aligned} \tag{7}$$

The **target distribution** is the empirical distribution of the lowest-resolution ground-truth images, where $x_1^{(1)} \sim \rho_1^{(1)}(x_1^{(1)}|x^{(K)})$. The **source distribution** is Gaussian: $x_0^{(1)} \sim \rho_0^{(1)}(x_0^{(1)}) = \mathcal{N}(0, \sigma^2 I_{d_1})$. The source is independent of the target, so $\rho_0^{(1)}(x_0^{(1)}|x_1^{(1)}) = \rho_0^{(1)}(x_0^{(1)})$ in (7).

Stage $k > 1$**: Image-to-Image Refinement** For all subsequent stages ($k > 1$), the model learns to perform a coarse-to-fine resolution enhancement. The core of these stages is the **conditional dependent coupling** that leverages the structural similarity between the lower-resolution source image and the higher-resolution target image. For each stage k, the **target distribution** is the distribution of ground-truth images $x_1^{(k)} \sim \rho_1^{(k)}(x_1^{(k)}|x^{(K)})$ at resolution d_k. Instead of drawing from a simple prior, the **source sample** $x_0^{(k)}$ is constructed directly from the corresponding target sample $x_1^{(k)}$: $x_0^{(k)} \sim \rho_0^{(k)}(x_0^{(k)}|x_1^{(k)})$. This is achieved by first applying a deterministic mapping $m_k(x_1^{(k)}) = U_k(D_k(x_1^{(k)}))$ and then adding Gaussian noise with stage-dependent scale:

$$x_0^{(k)} = m_k(x_1^{(k)}) + \sigma_k\zeta_k = U_k(D_k(x_1^{(k)})) + \sigma_k\zeta_k \tag{8}$$

where $\zeta_k \sim \mathcal{N}(0, I_{d_k})$ and σ_k is a small, stage-dependent noise level, which can be tuned. This construction induces the conditional probability $\rho_0^{(k)}(x_0^{(k)}|x_1^{(k)})$ as so called data-dependent coupling [30]. Thus, the joint probability density $\rho^{(k)}(x_0^{(k)}, x_1^{(k)}|x^{(K)})$ is shown as:

$$\rho_0^{(k)}(x_0^{(k)}|x_1^{(k)}) = \mathcal{N}(x_0^{(k)}; m_k(x_1^{(k)}), \sigma_k^2 I_{d_k}) \tag{9}$$

$$\rho^{(k)}(x_0^{(k)}, x_1^{(k)}|x^{(K)}) = \rho_0^{(k)}(x_0^{(k)}|x_1^{(k)})\rho_1^{(k)}(x_1^{(k)}|x^{(K)}) \tag{10}$$

The pair is constructed directly from the full-resolution training image and fresh noise. Gaussian noises are independent of the image and one another. A training update selects a stage and a local time; preceding stages need not be generated. The source mean already contains the target's coarse structure, so the endpoint displacement consists of unresolved image detail and the injected noise.

Model Parameterization We design a unified DiT model that operates consistently across multiple resolutions by sharing a single set of parameters. To enable the model to recognize and differentiate between these resolutions, we introduce an additional resolution embedding. Specifically, we treat the absolute resolution of the feature map r_k obtained after patch embedding [31] as a conditional signal. This signal is then encoded using sinusoidal positional embeddings [32] $e_k = E(r_k)$ and fused with the timestep embedding by projected addition, $\Gamma(\tau) + W_o W_v e_k$, before being fed into the model. All stages share these projections and the DiT parameters. See Appendix §F for details about resolution embedding.

To unify the multi-stage generation, the rescaled timestep strategy [16, 14, 33] is employed to align the multi-stage generation time definition with the standard Flow model [25], which generates across the time scope $t \in [0, 1]$. For each stage k, the start time point and end time point are defined as t_0^k and t_1^k, where $0 \le t_0^k < t_1^k \le 1$, $t_0^k = t_1^{k-1}$, and meet boundary condition $t_0^1 = 0$, $t_1^K = 1$. For simplicity, we construct a linear interpolant by specifying $\alpha_t = 1 - \tau$ and $\beta_t = \tau$ in Definition 2.1:

$$I_\tau^{(k)} \;=\; (1-\tau)\cdot x_0^{(k)} + \tau \cdot x_1^{(k)}, \tag{11}$$

where $\tau = \frac{t - t_0^k}{t_1^k - t_0^k}$, so that timestep $t \in [0, 1]$ of the entire multi-stage generation process located in each stage k: $t \in [t_0^k, t_1^k]$ is rescaled to $\tau \in [0, 1]$ within each stage k.

The single unified DiT is parameterized by b^θ to model all generation stages. For the deterministic interpolant (Flow/ODEs), derived from (4), the DiT is optimized by:

$$\mathcal{L}(\theta) = \mathbb{E}_{\substack{t\sim[0,1],k,e_k,\\ (x_0^{(k)},x_1^{(k)})\sim\rho^{(k)}(\cdot|x^{(K)})}} \left[\left| b^\theta(I_\tau^{(k)}, \tau, e_k) \right|^2 - 2\dot{I}_\tau^{(k)} \cdot b^\theta(I_\tau^{(k)}, \tau, e_k) \right] \tag{12}$$

where $k = \arg_k \{ t \in [t_0^k, t_1^k] \}$, and $\dot{I}_\tau^{(k)} = x_1^{(k)} - x_0^{(k)}$. Uniform global time induces stage probability $q(k) = t_1^k - t_0^k$ and uniform local time. Both training targets and stage integration use this local-time derivative.

3.3 Conditional Dependent Coupling

For the multi-stage generation model proposed in Sec. 3.2, we employed the **conditional dependent coupling** strategy to unify them. Instead of mapping unstructured noise to an image, our approach maps a structured prior—the upsampled low-resolution image from the previous stage with added stage-dependent noise—to the target high-resolution distribution. The joint probability distribution of the entire multi-stage interpolant with **conditional dependent coupling** is expressed as:

$$\rho\big(\{x_0^{(k)}, x_1^{(k)}\}_{k=1}^K \mid x^{(K)}\big) = \prod_{k=1}^K \rho^{(k)}(x_0^{(k)}, x_1^{(k)} \mid x^{(K)}) \tag{13}$$

$$= \prod_{k=1}^K \rho_0^{(k)}(x_0^{(k)} \mid x_1^{(k)}) \rho_1^{(k)}(x_1^{(k)} \mid x^{(K)}). \tag{14}$$

This formulation reveals a key property of our model. Conditioned on the final high-resolution image $x^{(K)}$, the data coupling at any given stage k, denoted as $\rho^{(k)}(x_0^{(k)}, x_1^{(k)} | x^{(K)})$, is independent of that at any other stage $k^* \neq k$:

$$(x_0^{(k)}, x_1^{(k)}) \perp\!\!\!\perp (x_0^{(k^*)}, x_1^{(k^*)}) \mid x^{(K)}. \tag{15}$$

Sequential dependency as Markov chain At inference, the preceding generated image supplies the next source. Let $P_k^\theta(\cdot \mid x_0)$ denote the endpoint kernel induced by integrating the stage-k ODE

from x_0. The resulting **Markov chain** [34] has the factorization

$$\begin{aligned}\rho(\hat{x}_0^{(1)}, \hat{x}_1^{(1)}, \ldots, \hat{x}_0^{(K)}, \hat{x}_1^{(K)}) &= \rho^{(1)}(\hat{x}_0^{(1)}, \hat{x}_1^{(1)}) \prod_{k=2}^{K} \rho^{(k)}(\hat{x}_0^{(k)}, \hat{x}_1^{(k)} \mid \hat{x}_1^{(k-1)}) \\ &= \rho_0^{(1)}(\hat{x}_0^{(1)}) P_1^{\theta}(\hat{x}_1^{(1)} \mid \hat{x}_0^{(1)}) \times \prod_{k=2}^{K} P_k^{\theta}(\hat{x}_1^{(k)} \mid \hat{x}_0^{(k)}) \rho_0^{(k)}(\hat{x}_0^{(k)} \mid \hat{x}_1^{(k-1)}),\end{aligned} \tag{16}$$

Here, the initial prior $\rho_0^{(1)}(\hat{x}_0^{(1)})$ is $\mathcal{N}(\hat{x}_0^{(1)}; 0, \sigma^2 I_{d_1})$. The endpoint kernels P_k^{θ} are parameterized through the ODEs of the shared model b^{θ}. For the refinement stages ($k \geq 2$), the conditional prior $\rho_0^{(k)}(\hat{x}_0^{(k)} \mid \hat{x}_1^{(k-1)})$ is defined by the deterministic upsampling step with added Gaussian noise:

$$\rho_0^{(k)}(\hat{x}_0^{(k)} \mid \hat{x}_1^{(k-1)}) = \mathcal{N}(\hat{x}_0^{(k)}; U_k(\hat{x}_1^{(k-1)}), \sigma_k^2 I_{d_k}), \tag{17}$$

which is implemented by sampling $\hat{x}_0^{(k)} = U_k(\hat{x}_1^{(k-1)}) + \sigma_k \zeta_k$, akin with the distribution formed by (8). Each transition uses only the preceding endpoint and independent fresh noise. The training joint describes target-derived pairs; the sampling factorization describes the generated sequence. Appendix §E connects their source marginals through the downsampling hierarchy.

Reducing transport cost For the linear interpolant, the stage-k coupling fixes the bound $L_k = \mathbb{E}\|x_1^{(k)} - x_0^{(k)}\|^2$. At refinement stages,

$$L_k = \mathbb{E}\|x_1^{(k)} - U_k D_k x_1^{(k)}\|^2 + d_k \sigma_k^2. \tag{18}$$

The first term is residual image detail and the second is noise energy. Flow matching fits the induced velocity; the operators and noise schedule determine this bound.

Theorem 3.1. *For aligned block averaging and replication, finite second moments, $K \geq 2$, $d_k = 4^{k-1} d_1$, and $\sigma_k = 2^{-(k-1)}\sigma$ with $\sigma > 0$, the independent full-resolution Gaussian/data pair has bound*

$$L_A = \mathbb{E}\|x^{(K)}\|^2 + d_K \sigma^2 > L_B = \sum_{k=1}^{K} L_k. \tag{19}$$

Each stage uses its unit local-time interval. Appendix §G.1 gives the projection identities, native-grid norm factors, and complete proof.

Learning clean stage distributions The target at every stage is the clean lower-resolution view of a real image. This supplies the shared model with direct image supervision at each scale, with coarse structure inherited through the paired source. Figure 1 shows these endpoint pairs. For class $\mathbf{c}$, define the target marginal

$$p_k(\cdot \mid \mathbf{c}) = \int \rho_1^{(k)}(\cdot \mid x^{(K)}, \mathbf{c})\, p_{\text{data}}(dx^{(K)} \mid \mathbf{c}). \tag{20}$$

If the preceding generated endpoint follows p_{k-1}, upsampling and fresh noise produce the training source marginal. With exact conditional velocities and exact ODE integration, induction then recovers p_k at every stage. The conditional field corresponds to CFG scale one. Appendix §E gives this argument; the endpoint experiment in Table 4(b) measures the effect of recursive sources for the trained, guided sampler.

Classifier-Free Guidance for Conditional Dependent Coupling Classifier-Free Guidance (CFG) [35] uses conditional and unconditional predictions of the shared model. A single CFG strength $S_{\text{cfg}} = 3$ is applied uniformly across stages in the main experiments. Each guided Euler step requires two field evaluations, giving $2\sum_k N_k$ total NFE for per-stage step counts N_k. Appendix §K gives the field and solver-call accounting.

Algorithm Workflows Here we present the detailed unconditional training and inference procedures for the unified multi-stage generative model b^{θ} with **conditional dependent coupling** in Appendix §J. The inference steps are described using the forward Euler method [24, 25] for solving the ODE, as a simple example of a numerical solver. Moreover, the detailed training and inference procedures for modeling b^{θ} in conditional generation with CFG are provided in Appendix §K.

4 Experiment

To comprehensively evaluate the effectiveness of the proposed unified multi-stage generation model with **conditional dependent coupling** (CDC-FM), we conduct experiments on ImageNet-1k. The main 256^2 model uses $32 \to 64 \to 128 \to 256$, $\sigma = 1$, and $\gamma = 2$. FID and IS use 50,000 generated images per configuration. End-to-end time for CDC-FM and the matched pixel baseline uses one A100 at batch size 32 and includes all stages, both CFG calls, resizing, and noise injection. The H100 batch-size-one protocol in Appendix H.1 measures individual forward calls.

Class-conditional image generation. In Table 1, we compare CDC-FM with various existing methods, including GANs, Autoregressive(AR) models, Diffusion/Flow Matching models, and Multi-stage models in the ImageNet-1k/256 benchmark [36], which represent current state-of-the-art generative performance. Higher-resolution results on the ImageNet-1k/512 benchmark [36] are deferred to Appendix §I (Table 16); they show the robustness of our method at higher resolution. We visualize the class-conditional image generation result of our method in Figure 7, demonstrating high-fidelity, accuracy, and diversity across all the classes.

	Model/Method	Params	FID ↓	IS ↑	Precision ↑	Recall ↑	NFE	Speed
GAN	BigGAN [37]	112M	6.95	224.5	0.89	0.38	1	1.46
	GigaGAN [38]	569M	3.45	225.5	0.84	0.61	1	1.32
	StyleGAN [39]	166M	2.30	265.1	0.78	0.53	1	0.96
Diffusion / Flow	ADM [40]	554M	10.94	101.0	0.69	0.63	250×2	9.46
	U-ViT [9]	287M	3.40	219.9	0.83	0.52	-	-
	LDM-4-G [4]	400M	3.60	247.7	-	-	250×2	4.18
	DiT-XL/2 [5]	675M	2.27	278.2	0.83	0.57	250×2	3.66
	SiT-XL/2 [6]	675M	2.06	270.3	0.82	0.59	250×2	3.53
	VDM++ [11]	2.0B	2.12	267.7	-	-	-	-
	SiD [41]	2.0B	2.77	211.8	-	-	512×2	10.12
	SiD2 [10]	2.0B	1.72	-	-	-	-	-
	JetFormer [42]	2.8B	6.64	-	0.69	0.56	-	-
	REPA [43]	675M	1.42	305.7	0.80	0.65	250×2	3.56
	LightningDiT [44]	675M	1.35	295.3	0.79	0.65	250×2	3.54
	MeanFlow [2]	675M	3.43	234.0	0.78	0.55	1	0.18
	REPA-E [45]	675M	1.83	296.4	0.81	0.62	250×2	3.55
	REG [46]	675M	1.32	308.9	0.81	0.66	250×2	3.55
Autoregressive	VQVAE-2 [47]	13.5B	31.11	45.0	0.36	0.57	-	-
	ViT-VQGAN [48]	1.7B	4.17	175.1	-	-	-	-
	VQGAN [49]	1.4B	15.78	74.3	-	-	1024	12.76
	RQTransformer [50]	3.8B	7.55	134.0	-	-	-	-
	LlamaGen-XL [7]	775M	2.62	244.1	0.80	0.57	-	-
	MaskGIT [51]	227M	6.18	182.1	0.80	0.51	8	0.97
	RCG [52]	502M	3.49	215.5	-	-	-	-
	MAR-H [12]	943M	1.55	303.7	0.81	0.62	256×2	1.23
	FlowAR-L [18]	589M	1.90	281.4	0.83	0.57	-	-
	VAR-d30 [8]	2.0B	1.92	323.1	0.82	0.59	10×2	1.12
Multi-stage	CDM [19]	-	4.88	158.7	-	-	-	-
	PixelFlow [16]	677M	1.98	282.1	0.81	0.60	$30 \times 4 \times 2$	7.62
	RIN [53]	410M	3.42	182.0	-	-	-	-
	PaGoDA [20]	0.9B	1.56	259.6	-	0.59	1	0.72
	FractalMAR-H [17]	848M	6.15	348.9	0.81	0.46	256×2	4.46
	Single-stage Pixel (K=1)	677M	1.82	305.8	0.81	0.60	250×2	8.73
	CDC-FM (ours)	677M	1.97	298.2	0.80	0.58	$4 \times 4 \times 2$	1.31
	CDC-FM (ours)	677M	1.87	301.8	0.82	0.60	$8 \times 4 \times 2$	2.66
	CDC-FM (ours)	677M	1.81	304.3	0.83	0.61	$16 \times 4 \times 2$	5.18

Table 1: Comprehensive comparison of generative models on ImageNet-1k/256. Missing values are denoted "-". The NFE field lists solver-step products; the final $\times 2$ counts the two CFG field evaluations per step. Speed is reported in seconds per image. CDC-FM and matched $K = 1$ time uses one A100 at batch size 32. Cross-family rows provide benchmark context.

Comparison with Recent Low-NFE and Latent-Space Methods. We additionally compare against recent low-NFE and latent-space methods, including MeanFlow [2] (1-NFE), REPA-E [45], and REG [46], as well as a single-stage pixel baseline (K=1) that shares our backbone, optimizer, training schedule, CFG rule, and inference solver, with a single full-resolution generation stage. Table 1 reports their quality and sampling budgets. CDC-FM reaches FID 1.97 with 32 total NFE. At similar FID to the matched single-stage pixel model, CDC-FM uses 128 total NFE and 5.18 seconds per image, compared with 500 NFE and 8.73 seconds, a 40.7% time reduction. Appendix §H gives the matched protocol and complete budgets.

Inference Efficiency. Thanks to the disentangled design of each generation stage in the proposed multi-stage generation model with **conditional dependent coupling**, we explicitly separate coarse

image generation from fine image refinement. This disentanglement enables both efficient inference and training while maintaining strong performance. Table 1 reports a comparison of inference time costs between representative methods with CDC-FM under varying numbers of function evaluations (NFE) on the ImageNet-1k/256 benchmark. Furthermore, we measure the inference time of CDC-FM at varying image resolutions: {64, 128, 256, 512} on the ImageNet-1k benchmark. Figure 2 reports measured sampling time and FID across image sizes. The stage-local cost and evaluation-time analyses appear in Theorems 3.1 and M.1.

Across Euler, Heun, RK4, and Dopri5, the multi-stage/single-stage time ratios at FID ≤ 1.83 are 0.59, 0.63, 0.67, and 0.70 (Appendix H.6).

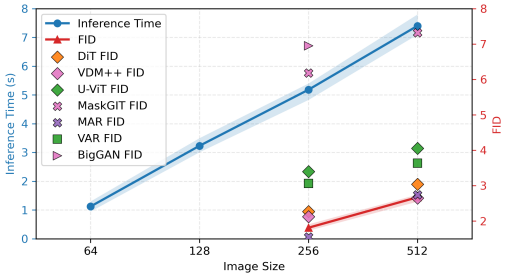

Figure 2: Inference time / FID across image sizes.

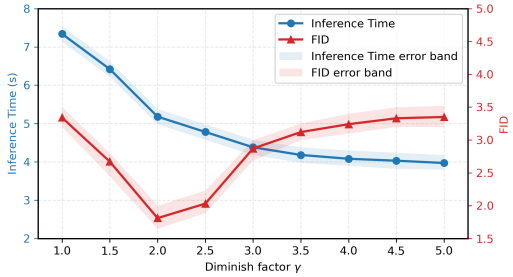

Figure 3: Effect of diminish factor.

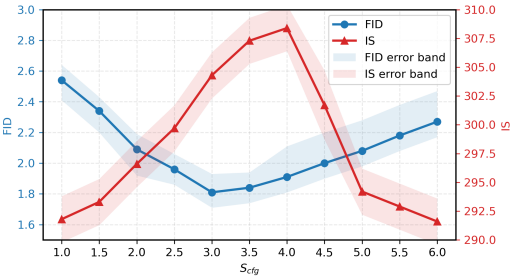

Figure 4: FID and IS under different S_{cfg}.

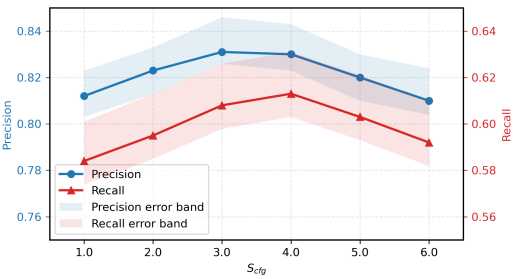

Figure 5: Precision and recall under different S_{cfg}.

Classifier-free Guidance. As discussed in Sec. 3.3, we employ a constant CFG strength S_{cfg} across all stages. The same guidance rule is used at every resolution. The performance of CDC-FM on the ImageNet-1k/256 benchmark under varying $S_{\text{cfg}} \in [1.0, 6.0]$ is reported in Figure 4 and Figure 5. To validate the effectiveness of using a constant CFG strength S_{cfg} across all stages, we report the per-stage FID under varying S_{cfg}. As shown in Figure 9, each stage attains its lowest FID at a similar S_{cfg} when using **conditional dependent coupling**. This demonstrates that employing a constant CFG strength S_{cfg} across all stages is feasible and effective.

Impact of the Diminish Factor. We further investigate the effect of the diminish factor γ in CDC-FM on the ImageNet-1k/256 benchmark, sweeping γ from 1.0 to 5.0 in increments of 0.5 and keeping $\sigma = 1$. The results, summarized in Figure 3, show that the FID initially decreases as γ increases, up to a certain tipping point, after which the FID begins to rise. The same figure records sampling time across the sweep. Table 15 gives the five-point FID comparison with a common 128-NFE budget, CFG scale three, and 50,000 samples per configuration.

To test the interaction between γ and σ, we further evaluate the FID under varying γ and σ in Figure 6. When the stage-k noise level σ_k becomes too small (which occurs when γ is too large or the base noise level σ is too small), each source concentrates around the upsampled coarse-image subspace [30]. Conversely, when σ_k becomes too large (resulting from a too-small γ or a too-large σ), $x_0^{(k)}$ retains insufficient information from the previous stage. Selecting an appropriate σ_k for each stage mitigates both

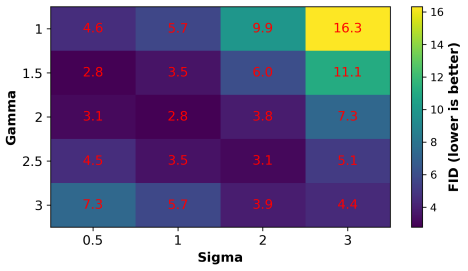

Figure 6: FID under varying γ and σ.

extremes. Introducing a small amount of Gaussian noise smooths the base density, ensuring it remains well-defined across the entire ambient space [30], while still preserving adequate information from the preceding stage. This balance enables each stage to refine its input effectively using fewer simulation steps.

To examine the effectiveness of the proposed method and the contribution of each individual component, we conduct a series of ablation studies. These include evaluating the unified model parameterization, the use of interpolants, the choice of resolution hierarchy (i.e., the stage count K), and the method by which the resolution embedding is incorporated. To evaluate whether the unified generative model parameterization enables effective knowledge sharing across stages, we compare the unified DiT (CDC-FM) with stage-specific DiTs on the ImageNet-1k/256 benchmark, reporting FID, model parameters, and training time. For this architecture comparison, we adopt the resolution hierarchy $[32, 64, 128, 256]$ with stage count $K = 4$. For the stage-specific DiTs, we use SiT-B (130M) for the 32, 64, and 128 resolutions, and SiT-L (458M) for the 256 resolution. To ensure a fair comparison, we keep the **conditional dependent coupling** identical to that used in the unified DiT. The results are presented in Table 2 (left). In the unified DiT, the resolution embedding e_k serves as a conditioning signal that identifies the current feature-map resolution r_k.

Table 2: Ablation studies on Unified Architecture and Fusion Mechanism.

Architecture Comparison				**Fusion Ablation**	
Method	FID	Params	Time	Variant	FID
Unified	**1.81**	677M	**76h**	Projected	**1.81**
Stage-spec.	2.68	$130\times3+458$M	78h	Direct	1.84

This allows all stages k to share the same model parameters, in contrast to stage-specific DiTs, which require separate parameter sets for each resolution. Such unified parameterization encourages efficient knowledge sharing across stages and leads to more compact and effective models. We examine this sharing on 10,000 matched validation images. Table 4(a) compares paired-image CKA for unified and stage-specific models after pooling features to a common grid. Adjacent-stage gradient cosines are $+0.18$, $+0.23$, and $+0.29$ over 128 matched minibatches. These measurements show related features and compatible updates; the complete shuffled control and protocol appear in Appendix F.1.

Table 3: Ablation studies on Interpolants (Left) and Hierarchy Stages (Right).

Interpolant Variant			**Hierarchy Setting**		
Type	Definition	FID	Stages	Res. Path	FID
Linear	$\alpha_t{=}1{-}\tau;\beta_t{=}\tau$	1.81	4	32…256	1.81
Trig.	$\alpha_t{=}\cos\frac{\pi\tau}{2};\beta_t{=}\sin\frac{\pi\tau}{2}$	2.11	2	64, 256	1.97
VP-shaped	$\alpha_t{=}\sqrt{1-\tau^2};\beta_t{=}\tau$	2.03	-	-	-

Table 4: Evidence for the shared model and recursive refinement. Panel (a) compares paired-image features at adjacent resolutions. Panel (b) evaluates stage endpoint FID with real or generated coarse sources; Δ is generated-source minus real-source FID.

(a) Cross-resolution representations			
Adjacent resolutions	$32 \leftrightarrow 64$	$64 \leftrightarrow 128$	$128 \leftrightarrow 256$
Unified CKA	0.76	0.82	0.87
Stage-specific CKA	0.54	0.59	0.64

(b) Recursive sources			
Endpoint resolution	64^2	128^2	256^2
Real-source FID	2.12	1.93	1.74
Generated-source FID	2.15	1.98	1.81
ΔFID	$+0.03$	$+0.05$	$+0.07$

The refinement experiment in Table 4(b) keeps one checkpoint, 16 Euler steps per stage, $S_{\text{cfg}} = 3$, and matched labels/noise. Replacing paired real coarse views with recursively generated images changes endpoint FID by 0.03, 0.05, and 0.07. This connects the stagewise training construction to the complete sampler; Appendix E.3 gives the 50,000-sample, same-resolution evaluation.

To further examine the selection of stochastic interpolants, we additionally test alternative interpolants: trigonometric [21], where $\alpha_t = \cos\frac{\pi}{2}\tau$ and $\beta_t = \sin\frac{\pi}{2}\tau$, and variance-preserving-shaped weights [54], where $\alpha_t = \sqrt{1-\tau^2}$ and $\beta_t = \tau$. The results are provided in Table 3 (left). For the ImageNet-1k/256 experiments, we use the resolution hierarchy $[32, 64, 128, 256]$ with stage count $K = 4$. We also test an alternative hierarchy $[64, 256]$ with stage count $K = 2$. A comparison between the two settings can be found in Table 3 (right). Here, we additionally evaluate a variant that replaces projected resolution addition with direct addition of e_k to the timestep embedding. The results of this comparison are shown in Table 2 (right). Additional conditioner variants and the five-stage hierarchy are reported in Appendix F.2.

5 Conclusion

We develop a unified multi-stage generation framework based on *stochastic interpolant* with conditional dependent coupling, which uses clean stage targets while operating efficiently in pixel space. The entire process can be parameterized by a single DiT, supporting end-to-end optimization and knowledge sharing across stages. Under the stated multiresolution operators and noise schedule, CDC-FM has a lower sum of prescribed stage-local transport-cost bounds than the independent full-resolution construction; the evaluation-cost model relates these bounds to sampling work. On ImageNet-1k/256, CDC-FM reaches FID 1.81 with 128 total NFE and reduces measured time by 40.7% against the matched single-stage pixel model at similar FID. The shared-feature and recursive-source studies support the use of one model across stages, and ImageNet-512 results extend the evaluation to higher resolution.

Acknowledgments and Disclosure of Funding

This manuscript was co-authored by Oak Ridge National Laboratory (ORNL), operated by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. Any subjective views or opinions expressed in this paper do not necessarily represent those of the U.S. Department of Energy or the United States Government. The authors declare no competing interests.

References

[1] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

[2] 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.

[3] Stathi Fotiadis, Noah Brenowitz, Tomas Geffner, Yair Cohen, Michael Pritchard, Arash Vahdat, and Morteza Mardani. Stochastic flow matching for resolving small-scale physics. *arXiv preprint arXiv:2410.19814*, 2024.

[4] 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 (CVPR)*, 2022.

[5] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[6] 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024.

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

[8] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[9] 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 (CVPR)*, 2023.

[10] Emiel Hoogeboom, Thomas Mensink, Jonathan Heek, Kay Lamerigts, Ruiqi Gao, and Tim Salimans. Simpler diffusion (sid2): 1.5 fid on imagenet512 with pixel-space diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[11] Diederik P. Kingma and Ruiqi Gao. Understanding diffusion objectives as the ELBO with simple data augmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

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

[13] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.

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

[15] Siyu Jiao, Gengwei Zhang, Yinlong Qian, Jiancheng Huang, Yao Zhao, Humphrey Shi, Lin Ma, Yunchao Wei, and Zequn Jie. Flexvar: Flexible visual autoregressive modeling without residual prediction. *arXiv preprint arXiv:2502.20313*, 2025.

[16] Shoufa Chen, Chongjian Ge, Shilong Zhang, Peize Sun, and Ping Luo. Pixelflow: Pixel-space generative models with flow. *arXiv preprint arXiv:2504.07963*, 2025.

[17] Tianhong Li, Qinyi Sun, Lijie Fan, and Kaiming He. Fractal generative models. *arXiv preprint arXiv:2502.17437*, 2025.

[18] Sucheng Ren, Qihang Yu, Ju He, Xiaohui Shen, Alan Yuille, and Liang-Chieh Chen. Flowar: Scale-wise autoregressive image generation meets flow matching. *arXiv preprint arXiv:2412.15205*, 2024.

[19] Jonathan Ho, Chitwan Saharia, William Chan, David J. Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research (JMLR)*, 23(47):1–33, 2022.

[20] Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Yuhta Takida, Naoki Murata, Toshimitsu Uesaka, Yuki Mitsufuji, and Stefano Ermon. Pagoda: Progressive growing of a one-step generator from a low-resolution diffusion teacher. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[21] Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

[22] Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*, 2022.

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

[24] Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky TQ Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code. *arXiv preprint arXiv:2412.06264*, 2024.

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

[26] Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2008.

[27] Cédric Villani. *Topics in optimal transportation*, volume 58. American Mathematical Soc., 2021.

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

[29] Jean-David Benamou and Yann Brenier. A computational fluid mechanics solution to the monge-kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, 2000.

[30] Michael S Albergo, Mark Goldstein, Nicholas M Boffi, Rajesh Ranganath, and Eric Vanden-Eijnden. Stochastic interpolants with data-dependent couplings. *arXiv preprint arXiv:2310.03725*, 2023.

[31] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[33] H Yan, X Liu, J Pan, JH Liew, Q Liu, and J Feng. Perflow: Piecewise rectified flow as universal plugand-play accelerator. *arXiv preprint arXiv:2405.07510*, 2024.

[34] James R Norris. *Markov chains*. Number 2. Cambridge university press, 1998.

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

[36] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[37] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations (ICLR)*, 2019.

[38] 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 (CVPR)*, 2023.

[39] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[40] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[41] Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In *International Conference on Machine Learning*, pages 13213–13232. PMLR, 2023.

[42] Michael Tschannen, André Susano Pinto, and Alexander Kolesnikov. Jetformer: An autoregressive generative model of raw images and text. *arXiv preprint arXiv:2411.19722*, 2024.

[43] 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. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=DJSZGGZYVi.

[44] Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 15703–15712, 2025.

[45] Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. REPA-e: Unlocking VAE for end-to-end tuning of latent diffusion transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

[46] Ge Wu, Shen Zhang, Ruijing Shi, Shanghua Gao, Zhenyuan Chen, Lei Wang, Zhaowei Chen, et al. Representation entanglement for generation: Training diffusion transformers is much easier than you think. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025.

[47] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[48] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved vqgan. *arXiv preprint arXiv:2110.04627*, 2021.

[49] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[50] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[51] 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*, pages 11315–11325, 2022.

[52] Tianhong Li, Dina Katabi, and Kaiming He. Self-conditioned image generation via generating representations. *CoRR*, 2023.

[53] Allan Jabri, David J. Fleet, and Ting Chen. Scalable adaptive computation for iterative generation. *arXiv preprint arXiv:2212.11972*, 2022.

[54] Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. *Advances in neural information processing systems*, 34: 1415–1428, 2021.

[55] Sucheng Ren, Qihang Yu, Ju He, Xiaohui Shen, Alan Yuille, and Liang-Chieh Chen. Flowar: Scale-wise autoregressive image generation meets flow matching. In *Forty-second International Conference on Machine Learning*, 2025.

[56] Lijun Yu, Jose Lezama, Nitesh Bharadwaj Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Agrim Gupta, Xiuye Gu, Alexander G Hauptmann, et al. Language model beats diffusion-tokenizer is key to visual generation. In *The Twelfth International Conference on Learning Representations*, 2025.

[57] 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.

[58] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022.

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

[60] Zhengyang Geng, Ashwini Pokle, Weijian Luo, Justin Lin, and J Zico Kolter. Consistency models made easy. In *The Thirteenth International Conference on Learning Representations*, 2025.

A Related Works

Latent Space Generation. Variational Autoencoders (VAEs) enable generative models to perform Flow/Diffusion generation, as well as autoregressive generation, within a lower-dimensional latent space [4–6, 47, 49, 48, 50, 7, 18, 8, 53]. This leads to more efficient training and inference. However, the compact representations produced by the encoder and the subsequent decoding process can introduce a degree of information and detail loss [12, 10, 41]. Despite this limitation, VAEs remain a common component in these generative models. Moreover, they are often trained separately from the generative model [16].

Pixel Space Generation. Although directly implementing the generation process avoids the need for VAEs and facilitates the preservation of fine-grained details, it requires substantial computation for both training and inference [4, 10, 41, 9]. Furthermore, due to the high dimensionality and complexity of the data distribution, learning to generate samples that faithfully capture detailed information from the target distribution becomes particularly challenging, especially in the absence of VAEs to extract meaningful latent features [16, 14, 18, 8, 53, 7].

Multi-stage Generation. Multi-stage generation methods advance sample synthesis by decomposing the process into a sequence of stages, where early stages operate in a lower-dimensional space to improve efficiency [16, 14, 8, 15, 55, 19, 53]. In addition, the overall generative task is divided into subtasks at each stage, enabling the model to gradually capture the complexity of the target data distribution [8, 15, 55, 19]. Pyramidal Flow and PixelFlow also use shared models across resolutions [16, 14]. CDC-FM pairs each clean stage target with its own noisy coarse view and learns the resulting refinement velocity with one resolution-conditioned transformer.

B Qualitative results

We visualize the class-conditional image generation result of our method in Figure 7, demonstrating high-fidelity, accuracy, and diversity across all the classes.

C Visualization of Unified Model with Conditional Dependent Coupling

In this section, we visualize representative examples of the output produced at each stage of the proposed method, together with the details added at each stage. Figure 8 illustrates the generated stage outputs and the corresponding detail hierarchy. For a detailed explanation of the training-pair construction and the Markovian inference recursion in (16), please refer to Appendix §E.

We also observe that the magnitude of the added details decreases as the stage index k increases. This indicates that the method decomposes the generation process into multiple stages in a coarse-to-fine manner: earlier stages focus on generating diverse coarse structures, while later stages progressively refine fine-grained details. Such decoupling enables both strong generative performance and improved sampling efficiency.

Classifier-free Guidance. As discussed in Sec. 3.3, we employ a constant CFG strength S_{cfg} across all stages. The same guidance rule is used at every resolution. The performance of CDC-FM on the ImageNet-1k/256 benchmark under varying $S_{\text{cfg}} \in [1.0, 6.0]$ is reported in Figure 4 and Figure 5. To validate the effectiveness of using a constant CFG strength S_{cfg} across all stages, we report the per-stage FID under varying S_{cfg}. As shown in Figure 9, each stage attains its lowest FID at a similar S_{cfg} when using **conditional dependent coupling**. This demonstrates that employing a constant CFG strength S_{cfg} across all stages is both feasible and effective.

D Stochastic Interpolant

The stochastic interpolant unifies the theory of Ordinary Differential Equations (ODEs) and Stochastic Differential Equations (SDEs).

Definition D.1 (Stochastic Interpolant)**.** [21, 30, 22] Given two probability density functions $\rho_0, \rho_1 : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$, a stochastic interpolant between ρ_0 *and* ρ_1 is a stochastic process I_t defined as

$$I_t = \alpha_t x_0 + \beta_t x_1 + \gamma_t z, \qquad t \in [0, 1] \tag{21}$$



Figure 7: ImageNet-1k/256 class-conditional generation results.

where α_t, β_t, and γ_t^2 are differentiable functions of time satisfying the boundary conditions

$$\alpha_0 = \beta_1 = 1, \quad \alpha_1 = \beta_0 = \gamma_0 = \gamma_1 = 0, \quad \text{and} \quad \alpha_t^2 + \beta_t^2 + \gamma_t^2 > 0 \quad \forall t \in [0, 1].$$

The pair (x_0, x_1) is drawn from a joint probability density $\rho(x_0, x_1)$ with finite second moments, whose marginals are ρ_0 and ρ_1:

$$\int_{\mathbb{R}^d} \rho(x_0, x_1) dx_1 = \rho_0(x_0), \tag{22}$$

$$\int_{\mathbb{R}^d} \rho(x_0, x_1) dx_0 = \rho_1(x_1), \tag{23}$$

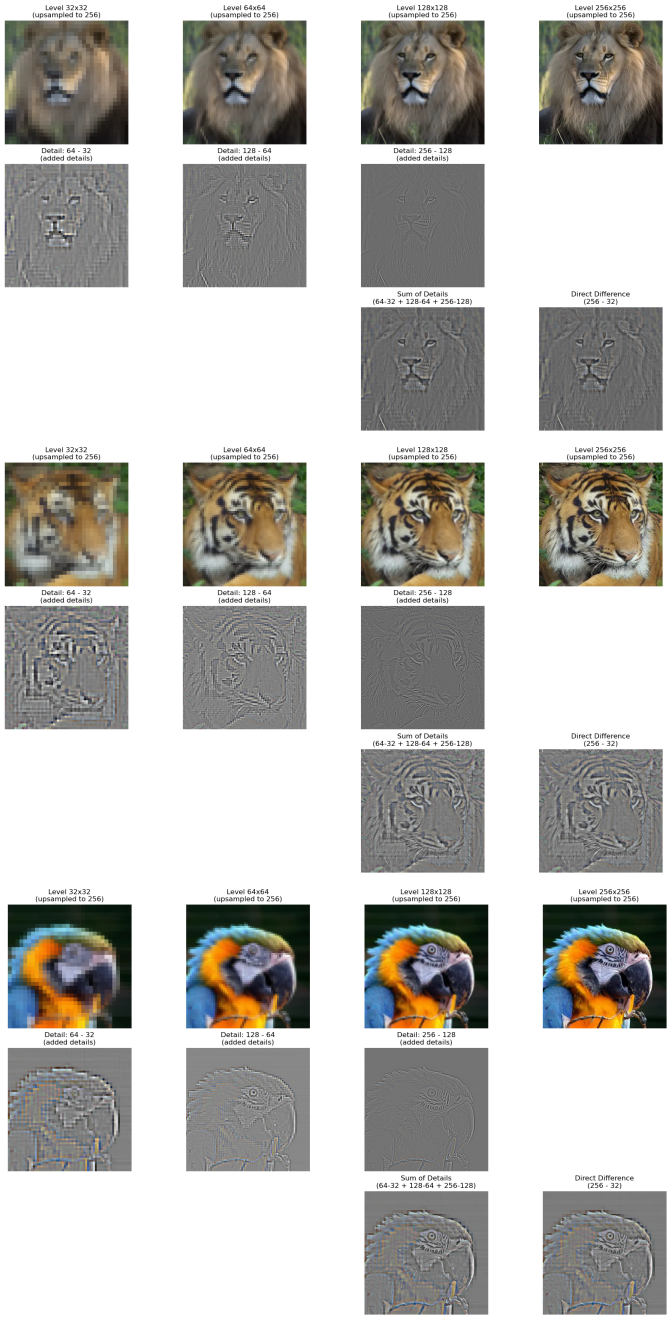

Figure 8: Visualization of each stage in CDC-FM, including the output at every stage, the details added at each stage, and the corresponding hierarchical structure of those details.

and $z \sim \mathcal{N}(0, \mathrm{Id})$ a Gaussian random variable independent of (x_0, x_1), i.e., $z \perp (x_0, x_1)$.

The stochastic interpolant framework uses information about the process I_t to derive either an ODE or an SDE. The solutions X_t to these equations are designed to push the initial law ρ_0 onto the law of the interpolant I_t for all times $t \in [0, 1]$. Consequently, the process I_t satisfies $I_{t=0} = x_0 \sim \rho_0(x_0)$ and $I_{t=1} = x_1 \sim \rho_1(x_1)$. This property allows for generative modeling: by drawing samples $x_0 \sim \rho_0(x_0)$ and using them as initial conditions $X_{t=0} = x_0$, one can generate samples $X_{t=1} \sim \rho_1(x_1)$ under the exact ODE or SDE dynamics; numerical integration approximates this evolution.

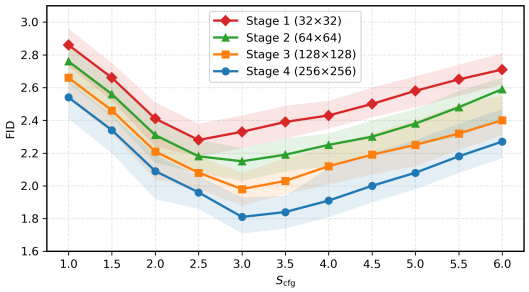

Figure 9: The per-stage FID under varying S_{cfg}.

Theorem D.2 (Transport equation)**.** *Let the time-dependent density of the stochastic interpolant I_t be $\rho_t(x)$. We define the velocity field $b_t(x)$ and the score field $s_t(x)$ as:*

$$b_t(x) = \mathbb{E}[\dot{I}_t \mid I_t = x] \tag{24}$$

$$s_t(x) = \nabla \log \rho_t(x) \tag{25}$$

where the dot denotes the time-derivative: $\dot{f} = \frac{\mathrm{d}f}{\mathrm{d}t}$ and the expectation is over $\rho(x_0, x_1)$ and z conditional on $I_t = x$. The probability density $\rho_t(x)$ satisfies the boundary conditions $\rho_{t=0}(x) = \rho_0(x)$ and $\rho_{t=1}(x) = \rho_1(x)$, and solves the transport equation:

$$\partial_t \rho_t(x) + \nabla \cdot (b_t(x)\rho_t(x)) = 0. \tag{26}$$

Moreover, for every t such that $\gamma_t \neq 0$, the score is given by:

$$s_t(x) = -\gamma_t^{-1}\mathbb{E}(z \mid I_t = x). \tag{27}$$

When the objectives are well defined, the fields b_t and s_t are the unique minimizers up to almost-everywhere equality of the respective objective functions:

$$L_b(\hat{b}) = \int_0^1 \mathbb{E}\left[\left|\hat{b}_t(I_t)\right|^2 - 2\dot{I}_t \cdot \hat{b}_t(I_t)\right] dt, \tag{28}$$

$$L_s(\hat{s}) = \int_0^1 \mathbb{E}\left[|\hat{s}_t(I_t)|^2 + 2\gamma_t^{-1} z \cdot \hat{s}_t(I_t)\right] dt, \tag{29}$$

where $\mathbb{E}$ denotes an expectation over $(x_0, x_1) \sim \rho(x_0, x_1)$ and $z \sim \mathcal{N}(0, \mathrm{Id})$.

The objective functions (28) and (29) can be readily estimated in practice from samples $(x_0, x_1) \sim \rho(x_0, x_1)$ and $z \sim \mathcal{N}(0, 1)$, which will enable us to learn approximations for use in a generative model. The transport equation 26 can be used to derive generative models, as we now show.

Corollary D.3 (Probability flow (ODE) and diffusions (SDE))**.** *The transport equation 26 implies that the density of the solutions X_t to the probability flow ODE matches the interpolant density ρ_t. The solutions to the probability flow equation:*

$$\dot{X}_t = b_t(X_t) \tag{30}$$

satisfy the properties that

$$X_{t=1} \sim \rho_1(x_1) \quad \textit{if} \quad X_{t=0} \sim \rho_0(x_0), \tag{31}$$

$$X_{t=0} \sim \rho_0(x_0) \quad \textit{if} \quad X_{t=1} \sim \rho_1(x_1). \tag{32}$$

In addition, for any choice of a time-dependent diffusion coefficient $\epsilon_t \geq 0$, solutions to the forward SDE

$$dX_t^F = \left(b_t(X_t^F) + \epsilon_t s_t(X_t^F)\right) dt + \sqrt{2\epsilon_t} dW_t, \tag{33}$$

satisfy the property that

$$X_{t=1}^F \sim \rho_1(x_1) \quad \textit{if} \quad X_{t=0}^F \sim \rho_0(x_0). \tag{34}$$

And solutions to the backward SDE

$$dX_t^R = \left(b_t(X_t^R) - \epsilon_t s_t(X_t^R)\right) dt + \sqrt{2\epsilon_t} dW_t, \tag{35}$$

satisfy the property that

$$X_{t=0}^R \sim \rho_0(x_0) \quad \textit{if} \quad X_{t=1}^R \sim \rho_1(x_1). \tag{36}$$

Both deterministic and stochastic generative models were derived within the stochastic interpolant framework.

E Justification of Conditional Dependent Coupling and Markovian Inference

In this appendix, we describe the training joint in (13) and the sampling recursion in (16), then connect their source marginals through the image hierarchy.

E.1 From Training Pairs to Markovian Inference

During training, $x_1^{(k)} = D_{K\to k}x^{(K)}$ and the source is constructed by Eq. (8). Given $x^{(K)}$, the stage pairs are independent because their noises are independent. Averaging this construction over the real image gives each stage's joint training law.

Sampling instead composes learned ODE maps. For a fixed class and guidance rule, write $\Phi_k^\theta(x_0)$ for the endpoint obtained by integrating stage k. Its kernel is $P_k^\theta(\cdot \mid x_0) = \delta_{\Phi_k^\theta(x_0)}$. The source kernel is

$$Q_k(y,\cdot) = \mathcal{N}(\cdot; U_k y, \sigma_k^2 I_{d_k}).$$

The generated endpoint transition is $R_k^\theta(y,\cdot) = (\Phi_k^\theta)_\# Q_k(y,\cdot)$, where $f_\# p$ denotes the law of $f(X)$ for $X \sim p$. It uses the preceding endpoint and fresh noise. Hence the generated path has law

$$\hat{p}_1(dx_1)\prod_{k=2}^{K} R_k^\theta(x_{k-1}, dx_k). \tag{37}$$

This factorization defines the Markov sampler directly. It is also the endpoint marginal of Eq. (16). The same shared DiT supplies the stage velocities used in all Φ_k^θ.

E.2 Equivalence of Training and Inference Priors at Refinement Stages

We now show that the refinement-stage prior (17) used at inference, which is implemented by:

$$\hat{x}_0^{(k)} = U_k(\hat{x}_1^{(k-1)}) + \sigma_k\zeta_k$$

uses the same upsampling/noise kernel as the training construction. The source marginal follows by integrating this kernel over the preceding endpoint law.

First, recall the *transitivity* of the downsampling operators defined in Sec. 3.1:

$$D_{k+1\to k-1} = D_k \circ D_{k+1}, \qquad D_{K\to k-1} = D_k \circ D_{K\to k}.$$

By definition of the stage-k ground-truths,

$$x_1^{(k)} = D_{K\to k}(x^{(K)}) \quad \Rightarrow \quad D_k(x_1^{(k)}) = D_k(D_{K\to k}(x^{(K)})) = D_{K\to k-1}(x^{(K)}) = x_1^{(k-1)}.$$

Hence, for the *data-construction process* used in training,

$$m_k(x_1^{(k)}) = U_k(D_k(x_1^{(k)})) = U_k(x_1^{(k-1)}).$$

Plugging this into (8) gives

$$x_0^{(k)} = U_k(x_1^{(k-1)}) + \sigma_k\zeta_k, \quad \zeta_k \sim \mathcal{N}(0, I_{d_k}),$$

so that

$$x_0^{(k)} \sim \mathcal{N}\big(x_0^{(k)}; U_k(x_1^{(k-1)}), \sigma_k^2 I_{d_k}\big).$$

At inference time, we define the refinement prior by:

$$\hat{x}_0^{(k)} = U_k(\hat{x}_1^{(k-1)}) + \sigma_k\zeta_k, \quad \zeta_k \sim \mathcal{N}(0, I_{d_k}),$$

which induces to (17):

$$\rho_0^{(k)}(\hat{x}_0^{(k)} \mid \hat{x}_1^{(k-1)}) = \mathcal{N}\big(\hat{x}_0^{(k)}; U_k(\hat{x}_1^{(k-1)}), \sigma_k^2 I_{d_k}\big).$$

Source marginals and ideal generation. Fix class $\mathbf{c}$ and let p_k be the target marginal in Eq. (20). The hierarchy gives $(D_k)_{\#}p_k = p_{k-1}$. Integrating the training conditional over real images therefore gives

$$\rho_{0,k}(\cdot \mid \mathbf{c}) = \int Q_k(y, \cdot)\, p_{k-1}(dy \mid \mathbf{c}). \tag{38}$$

If the preceding generated endpoint has law p_{k-1}, its next source has this same marginal. With exact conditional velocity fields, well-posed flows and exact ODE integration, stage 1 transports its Gaussian source to p_1. The source identity and exact stage flow then give p_2, and induction recovers all target marginals. This conditional field is the CFG-scale-one field. Guided finite-step generation is evaluated with the trained checkpoint below.

Same-stage endpoint comparison. For a learned sampler, compare the generated endpoint law $\hat{p}_k$ and target law p_k on the same $\mathbb{R}^{d_k}$. Wasserstein distance requires finite relevant moments; KL additionally requires absolute continuity. The model b^θ parameterizes a velocity field, while these endpoint laws describe image distributions.

E.3 Refinement from real and generated sources

Table 4(b) evaluates the stage map with the paired real coarse view $D_k x_1^{(k)}$ or the recursively generated $\hat{x}_1^{(k-1)}$. One checkpoint is used for both conditions. Each evaluation generates 50,000 samples with 16 Euler steps per stage, $S_{\text{cfg}} = 3$, and matched labels and noise. FID uses held-out real images downsampled to the evaluated endpoint resolution. The real/generated-source FIDs are 2.12/2.15 at 64^2, 1.93/1.98 at 128^2, and 1.74/1.81 at 256^2, giving differences 0.03, 0.05, and 0.07. This measures the endpoint effect of changing the coarse input while keeping the stage map, solver budget, class, and noise fixed.

F Detailed Implementation of Resolution Embedding

e_k is fused with the timestep embedding by projected addition before being fed into the model. The detailed fusion procedure is given below. For each stage k and rescaled timestep $\tau \in [0, 1]$, We follow the standard DiT sinusoidal embedding followed by an MLP:

$$\gamma_t = \Gamma(\tau) \in \mathbb{R}^{d_{\text{model}}}. \tag{39}$$

Given the absolute resolution r_k of stage k, we construct the resolution embedding using an analogous sinusoidal embedding followed by an MLP:

$$e_k = E(r_k) \in \mathbb{R}^{d_{\text{model}}}. \tag{40}$$

Both γ_t and e_k are 1D vectors in the same model space $\mathbb{R}^{d_{\text{model}}}$.

The projected resolution vector is

$$v_k = W_v e_k, \qquad z_{k,t} = W_o v_k, \qquad W_v \in \mathbb{R}^{d_h \times d_{\text{model}}}, \quad W_o \in \mathbb{R}^{d_{\text{model}} \times d_h}. \tag{41}$$

The fused conditioning vector is

$$c_{k,t} = \gamma_t + W_o W_v e_k. \tag{42}$$

For a single key, the softmax weight equals one, giving the projected addition above. The projections and embeddings are shared across stages.

Let $X_0^{(k)} \in \mathbb{R}^{N_k \times d_{\text{model}}}$ denote the sequence of patch tokens at stage k after patch embedding of $I_\tau^{(k)}$. The unified DiT is conditioned on the fused vector $c_{k,t}$. We use $c_{k,t}$ as a global conditioning signal for all DiT blocks via AdaLN modulation [23], following DiT [23], SiT [6] and PixelFlow [16]:

$$X^{\ell+1} = X^\ell + F^{(\ell)}\big(\mathrm{LN}(X^\ell);\, c_{k,t}\big), \quad \ell = 1, \ldots, L, \tag{43}$$

where each block $F^{(\ell)}$ is a DiT block whose AdaLN modulation parameters are produced from $c_{k,t}$.

Conceptually, the same DiT parameters are reused for all stages, while the pair (τ, r_k) is compressed into the single fused conditioner $c_{k,t}$, which tells the model both *where in time* and *at which resolution* it is operating.

Table 5: Cross-resolution feature and gradient measurements.

Metric	$32 \leftrightarrow 64$	$64 \leftrightarrow 128$	$128 \leftrightarrow 256$
CKA, unified	0.76	0.82	0.87
CKA, stage-specific	0.54	0.59	0.64
CKA, shuffled identities	0.22	0.24	0.27
Gradient cosine, unified	$+0.18$	$+0.23$	$+0.29$

Table 6: Resolution conditioning and an additional hierarchy on ImageNet-256.

(a) Matched-training conditioning variants		FID
Projected resolution addition		1.81
Direct resolution addition		1.84
Learned stage embedding with projected fusion		1.90
Time conditioning alone		2.08
(b) Same-checkpoint hierarchy	Total NFE	FID
$32 \to 64 \to 128 \to 256$	128	1.81
$16 \to 32 \to 64 \to 128 \to 256$	160	1.96

F.1 Shared cross-resolution representations

The architecture comparison keeps CDC in both models. The unified DiT has 677M parameters, takes 76 hours of training, and gives FID 1.81. Stage-specific models use 130M parameters at each of 32^2, 64^2, and 128^2, and 458M at 256^2, totaling 848M; training takes 78 hours and gives FID 2.68.

On 10,000 ImageNet validation images, adjacent-stage inputs share the full-resolution image, class label, and local time. Higher-resolution tokens are pooled to the lower grid. Linear CKA measures normalized similarity of centered feature Gram matrices, averaged at 25%, 50%, 75%, and 100% relative depth. The shuffled control permutes image identities at one resolution. Gradient cosine compares adjacent-stage loss gradients with respect to shared DiT parameters, averaged over 128 matched minibatches. Table 5 reports all measurements.

F.2 Conditioning variants and an additional hierarchy

The conditioning variants are retrained with matched data, backbone, training budget, and evaluator on $32 \to 64 \to 128 \to 256$. They use projected resolution addition, direct addition, a learned stage embedding with projected fusion, or time conditioning alone. Table 6(a) gives their FIDs.

The additional-hierarchy test uses the same trained checkpoint, averaging/replication, 16 Euler steps per stage, $S_{\text{cfg}} = 3$, and $\sigma_k = 2^{-(k-1)}\sigma$. The five-stage path introduces 16^2 before the original four resolutions. Table 6(b) reports both path budgets and FIDs.

G Proof

G.1 Proof of Theorem 3.1

We provide a formal proof that a cascaded image generation model using Flow Matching with **conditional dependent coupling** has a lower sum of stage-local transportation-cost bounds than a direct, single-stage model that generates a high-resolution image from Gaussian noise, as shown in Theorem 3.1. The proof is based on an orthogonal decomposition of the image signal energy, which separates the effects of image-detail energy and the stage-dependent Gaussian noise.

G.1.1 Problem Formulation and Definitions

Our goal is to prove that the transportation cost of a direct Flow Matching model (L_A) is greater than the total transport cost of a multi-stage cascaded model with conditional dependent couplings (L_B).

Proposition G.1 (Transportation Cost Bound)**.** *From Definition 2.4, the transportation cost of a probability flow is bounded by the integral of the expected squared norm of the interpolant's time*

derivative. For an interpolant path from a source distribution $\rho_0(x_0)$ to a target distribution $\rho_1(x_1)$, we denote this bound as L:

$$\textit{Cost} = \mathbb{E}_{x_0\sim\rho_0}\left[|X_{t=1}(x_0) - x_0|^2\right] \le \int_0^1 \mathbb{E}\left[|\dot{I}_t|^2\right] dt := L. \tag{44}$$

For simplicity in the proof, we analyze the case where the interpolant is $I_t = (1-t)x_0 + tx_1$, such that its derivative is $\dot{I}_t = x_1 - x_0$. The comparison below concerns these linear paths. In this case, the cost bound reduces to [24, 30]:

$$L = \mathbb{E}[|x_1 - x_0|^2]. \tag{45}$$

Model A: Direct Generation This model generates a high-resolution image $x^{(K)} \in \mathbb{R}^{d_K}$ in a single stage.

- **Source Distribution** $\rho_0(x_0)$: $x_0 \sim \mathcal{N}(0, \sigma^2 I_{d_K})$.
- **Target Distribution** $\rho_1(x_1)$: The high-resolution data distribution, $x_1 \sim x^{(K)}$.
- **Cost** (L_A): The transport cost from x_0 to x_1.

Model B: Cascaded Generation This model generates the image in K stages, from resolution d_1 to d_K.

- **Stage 1:** Generates a low-resolution image $x_1^{(1)} \in \mathbb{R}^{d_1}$.
  - Source: $x_0^{(1)} \sim \mathcal{N}(0, \sigma^2 I_{d_1})$.
  - Target: $x_1^{(1)} \sim \rho_1^{(1)}(x_1^{(1)}|x^{(K)})$.
  - Cost: L_1.
- **Stages** $k \in [2, K]$**:** Refines a coarse image to a higher resolution image $x_1^{(k)} \in \mathbb{R}^{d_k}$.
  - Target: $x_1^{(k)} \sim \rho_1^{(k)}(x_1^{(k)}|x^{(K)})$.
  - Source: $x_0^{(k)} \sim \rho_0^{(k)}(x_0^{(k)}|x_1^{(k)})$ is defined by a deterministic **conditional dependent coupling** with a small stage-dependent Gaussian noise: $x_0^{(k)} = U_k(D_k(x_1^{(k)})) + \sigma_k\zeta_k$, as shown in (8).
  - Cost: L_k.
- **Total Cost** (L_B): The sum of the costs of all stages, $L_B = \sum_{k=1}^K L_k$.

We seek to prove that $L_A > L_B$.

Native and full-resolution norms. Let $A_k = U_{k\to K}$. Replication repeats each coordinate 4^{K-k} times, so $A_k^\top A_k = 4^{K-k} I_{d_k}$. Consequently, a native-grid residual and its full-resolution embedding have different squared norms. We keep this factor explicit throughout the proof.

G.1.2 Orthogonal Decomposition of Image Energy

We formalize the concept that an image can be represented as a base layer plus a sum of details.

Definition G.2 (Multiresolution Projections)**.** Let the vector space be that of the highest resolution, $\mathbb{R}^{d_K}$. We define an embedding operator $U_{k\to K} : \mathbb{R}^{d_k} \to \mathbb{R}^{d_K}$ that upscales a level-k image to the full resolution space. We then define a projection operator $D_{K\to k} : \mathbb{R}^{d_K} \to \mathbb{R}^{d_k}$ that projects a high-resolution image onto the subspace of images with resolution k. These two definitions are inherited from the previous section. For an image $x \in \mathbb{R}^{d_K}$, let $\tilde{x}^{(k)} = U_{k\to K}(D_{K\to k}(x))$. We also define the null image $\tilde{x}^{(0)} = \mathbf{0}$. Let the **detail vector** at level k be $\tilde{\epsilon}^{(k)} = \tilde{x}^{(k)} - \tilde{x}^{(k-1)}$.

Assumption G.3 (Orthogonality of Details)**.** The detail vector $\tilde{\epsilon}^{(k)}$, which contains the frequency content for level k, is orthogonal to the detail vectors of all other levels.

$$\mathbb{E}[(\tilde{\epsilon}^{(k)})^T \tilde{\epsilon}^{(j)}] = 0 \quad \text{for} \quad k \neq j \tag{46}$$

For aligned block averaging and replication, $D_{K\to k} = (A_k^\top A_k)^{-1} A_k^\top$, so $P_k = A_k D_{K\to k}$ is an orthogonal projection. The projection ranges are nested: $P_i P_j = P_i$ for $i < j$. Thus $(P_i - P_{i-1})(P_j - P_{j-1}) = 0$, which proves the stated orthogonality for these operators. Under Assumption G.3, the full image can be written as a telescoping sum:

$$x = \tilde{x}^{(K)} = \sum_{k=1}^{K} (\tilde{x}^{(k)} - \tilde{x}^{(k-1)}) = \sum_{k=1}^{K} \tilde{\epsilon}^{(k)} \tag{47}$$

Lemma G.4 (Energy Decomposition)**.** *Under Assumption G.3, the expected energy of an image is the sum of the expected energies of its orthogonal detail components.*

$$\mathbb{E}[|x|^2] = \mathbb{E}\left[\left|\sum_{k=1}^{K} \tilde{\epsilon}^{(k)}\right|^2\right] = \sum_{k=1}^{K} \mathbb{E}[|\tilde{\epsilon}^{(k)}|^2] \tag{48}$$

Proof. This follows directly from the linearity of expectation and the orthogonality assumption, as all cross-terms $\mathbb{E}[(\tilde{\epsilon}^{(k)})^T \tilde{\epsilon}^{(j)}]$ for $k \neq j$ vanish. □

G.1.3 Analysis of Transportation Costs

Cost of the Direct Model (L_A). The cost is $L_A = \mathbb{E}[|x_1 - x_0|^2]$, where $x_1 \sim x^{(K)}$ and $x_0 \sim \mathcal{N}(0, \sigma^2 I_{d_K})$. Due to independence and $\mathbb{E}[x_0] = 0$:

$$\begin{aligned} L_A &= \mathbb{E}[|x_1|^2] - 2\mathbb{E}[x_1^T x_0] + \mathbb{E}[|x_0|^2] \\ &= \mathbb{E}[|x_1|^2] + \mathbb{E}[|x_0|^2] \end{aligned} \tag{49}$$

Using Lemma G.4 and the variance of the noise, $\mathbb{E}[|x_0|^2] = d_K \sigma^2$:

$$L_A = \left(\sum_{k=1}^{K} \mathbb{E}[|\tilde{\epsilon}^{(k)}|^2]\right) + d_K \sigma^2 \tag{50}$$

Cost of the Cascaded Model with conditional dependent couplings (L_B). The total cost is $L_B = \sum_{k=1}^{K} L_k$.

- **Stage 1:** $L_1 = \mathbb{E}[|x_1^{(1)} - x_0^{(1)}|^2] = \mathbb{E}[|x_1^{(1)}|^2] + \mathbb{E}[|x_0^{(1)}|^2]$. The data $x_1^{(1)}$ represents the coarsest level of detail, corresponding to $\tilde{\epsilon}^{(1)}$. The noise variance is $\mathbb{E}[|x_0^{(1)}|^2] = d_1 \sigma^2$. Thus,
$$L_1 = 4^{-(K-1)} \mathbb{E}[|\tilde{\epsilon}^{(1)}|^2] + d_1 \sigma^2 \tag{51}$$
- **Stages $k \in [2, K]$:** The cost is $L_k = \mathbb{E}[|x_1^{(k)} - x_0^{(k)}|^2]$. Substituting the data-dependent coupling $x_0^{(k)} = U_k(D_k(x_1^{(k)})) + \sigma_k \zeta_k$, we get:
$$\begin{aligned} L_k &= \mathbb{E}[|x_1^{(k)} - U_k(D_k(x_1^{(k)})) - \sigma_k \zeta_k|^2] \\ &= \mathbb{E}[|x_1^{(k)} - U_k(D_k(x_1^{(k)}))|^2] - 2\,\mathbb{E}[\sigma_k \zeta_k^T \cdot (x_1^{(k)} - U_k(D_k(x_1^{(k)})))] + \mathbb{E}[|\sigma_k \zeta_k|^2] \\ &= \mathbb{E}[|x_1^{(k)} - U_k(D_k(x_1^{(k)}))|^2] - 2\,\mathbb{E}[\sigma_k \zeta_k]\,\mathbb{E}[(x_1^{(k)} - U_k(D_k(x_1^{(k)})))] + \mathbb{E}[|\sigma_k \zeta_k|^2] \\ &= \mathbb{E}[|x_1^{(k)} - U_k(D_k(x_1^{(k)}))|^2] + \mathbb{E}[|\sigma_k \zeta_k|^2], \end{aligned} \tag{52}$$
where independence: $\sigma_k \zeta_k \perp\!\!\!\perp (x_1^{(k)} - U_k(D_k(x_1^{(k)})))$ and $\mathbb{E}[\sigma_k \zeta_k] = 0$. The native residual $\delta_k = x_1^{(k)} - U_k D_k x_1^{(k)}$ satisfies $A_k \delta_k = \tilde{\epsilon}^{(k)}$. The replication identity therefore gives $\mathbb{E}\|\delta_k\|^2 = 4^{-(K-k)} \mathbb{E}\|\tilde{\epsilon}^{(k)}\|^2$. Hence
$$L_k = 4^{-(K-k)} \mathbb{E}\|\tilde{\epsilon}^{(k)}\|^2 + d_k \sigma_k^2. \tag{53}$$

Summing the costs for all stages of Model B gives

$$L_B = \sum_{k=1}^{K} 4^{-(K-k)} \mathbb{E}\|\tilde{\epsilon}^{(k)}\|^2 + \sum_{k=1}^{K} d_k \sigma_k^2. \tag{54}$$

Table 7: CDC stage costs on held-out images. Per-coordinate and native costs are normalized by their stage-1 values.

Stage	Resolution	d_k/d_1	$(L_k/d_k)/(L_1/d_1)$	L_k/L_1
1	32^2	1	1.00	1.00
2	64^2	4	0.43	1.72
3	128^2	16	0.20	3.20
4	256^2	64	0.09	5.76

Assumption G.5 (Noise-decay schedule)**.** For side lengths doubling at each stage and fixed channels, $d_k = 4^{k-1}d_1$. Let $\sigma_1 = \sigma > 0$ and $\sigma_k = 2^{-(k-1)}\sigma$. Then $d_k\sigma_k^2 = d_1\sigma^2$ at every stage.

For general $\sigma_k = \gamma^{-(k-1)}\sigma$, the aggregate noise term is $d_1\sigma^2 \sum_{k=1}^{K}(4/\gamma^2)^{k-1}$. The theorem uses $\gamma = 2$.

G.1.4 Main Proof and Conclusion

Proof. The direct and cascaded bounds are

$$L_A = \sum_{k=1}^{K} \mathbb{E}\|\tilde{\epsilon}^{(k)}\|^2 + 4^{K-1}d_1\sigma^2, \tag{55}$$

$$L_B = \sum_{k=1}^{K} 4^{-(K-k)}\mathbb{E}\|\tilde{\epsilon}^{(k)}\|^2 + Kd_1\sigma^2. \tag{56}$$

Subtracting yields

$$L_A - L_B = \sum_{k=1}^{K}(1 - 4^{-(K-k)})\mathbb{E}\|\tilde{\epsilon}^{(k)}\|^2 + (4^{K-1} - K)d_1\sigma^2 > 0. \tag{57}$$

The signal terms are nonnegative. The noise term is strictly positive for $K \geq 2$ and $\sigma > 0$. This proves Theorem 3.1. □

Local-time convention and regression. Each L_k integrates over its own $\tau \in [0, 1]$. The theorem compares these stage-local bounds. For $V_k = x_1^{(k)} - x_0^{(k)}$, the stage contribution to the original training objective satisfies

$$\mathbb{E}[\|b^\theta\|^2 - 2V_k^\top b^\theta] = \mathbb{E}\|b^\theta - V_k\|^2 - L_k.$$

The endpoint law fixes L_k, while regression learns the conditional velocity induced by that law.

G.1.5 Stage-cost measurements and operators

The stage-cost study uses 50,000 held-out ImageNet images and averaging/replication. Table 7 distinguishes costs measured on the native image grid from costs per coordinate. The aggregate costs are $L_B/L_1 = 11.68$, $L_A/L_1 = 97.3$, and $L_B/L_A = 0.12$. For the operator comparison, define the normalized full-resolution cross-detail energy

$$R_{\text{cross}} = \frac{\sum_{i\neq j} |\mathbb{E}[\tilde{\epsilon}^{(i)\top}\tilde{\epsilon}^{(j)}]|}{\sum_i \mathbb{E}\|\tilde{\epsilon}^{(i)}\|^2}.$$

Each pair is trained separately with the same architecture, optimization budget, and evaluator. Table 8 reports the diagnostic, cost ratio, and FID. Averaging/replication has the projection structure established above.

G.2 Proof of Theorem M.1

We provide a formal proof that the cascaded Flow Matching model with the **conditional dependent coupling** strategy has strictly smaller network-evaluation time under the stated cost and budget models

Table 8: Down/upsampling operators under matched training settings.

Operators	R_{cross}	L_B/L_A	FID
Average/replication	$< 0.001\%$	0.12	1.81
Area/bilinear	1.5%	0.15	1.84
Bicubic/bicubic	3.5%	0.18	1.86

than the direct single-stage model for high-resolution image generation, as stated in Theorem M.1. The argument follows three steps: (i) relate the expected network-evaluation-time functional $\iota(\cdot)$ to the Number of Function Evaluations (NFE) and the per-evaluation cost, (ii) connect NFE to a transportation cost, and (iii) compare the resulting computational "loads" of the direct and cascaded schemes by separating signal and noise contributions.

G.2.1 Problem Formulation and Assumptions

Using the network-evaluation times in Theorem M.1, write

$$T_{A,\text{eval}} = \mathbb{E}\big[\iota\big(x_0 \to \hat{x}_1\big)\big]\,,$$
$$T_{B,\text{eval}} = \sum_{k=1}^{K} \mathbb{E}\Big[\iota\big(\hat{x}_0^{(k)} \to \hat{x}_1^{(k)}\big)\Big]\,.$$

The direct model draws $x_0 \sim \mathcal{N}(0, \sigma^2 I_{d_K})$ and produces $\hat{x}_1 \in \mathbb{R}^{d_K}$ in one stage. The cascaded sampler starts from $\hat{x}_0^{(1)} \sim \mathcal{N}(0, \sigma^2 I_{d_1})$. For $k > 1$, it constructs $\hat{x}_0^{(k)} = U_k(\hat{x}_1^{(k-1)}) + \sigma_k \zeta_k$ and integrates the stage ODE to obtain $\hat{x}_1^{(k)}$, as in Eq. (17). The bounds L_k below are defined by the paired training endpoints $x_0^{(k)}, x_1^{(k)}$ and enter the stated evaluation-budget model.

Assumption G.6 (Per-Evaluation Cost)**.** For ODE-based generators (e.g., probability flow ODEs) under a fixed backbone and solver tolerance, the per-function-evaluation cost scales linearly with the ambient dimensionality d. Per-call measurements at three resolutions are reported in Appendix §H.

Per-evaluation cost model. Under Assumption G.6, the expected inference-time cost is the product of the NFE and the cost per function evaluation, and the per-evaluation cost scales linearly with the ambient dimensionality:

$$\mathbb{E}\big[\iota\big(x_0 \to \hat{x}_1\big)\big] \;=\; \alpha\, N_A\, d_K, \qquad \mathbb{E}\Big[\iota\big(\hat{x}_0^{(k)} \to \hat{x}_1^{(k)}\big)\Big] \;=\; \alpha\, N_k\, d_k,$$

for some constant $\alpha > 0$. Hence

$$T_{A,\text{eval}} \;=\; \alpha\, N_A\, d_K, \qquad T_{B,\text{eval}} \;=\; \alpha \sum_{k=1}^{K} N_k\, d_k.$$

To prove $T_{A,\text{eval}} > T_{B,\text{eval}}$ it suffices to show

$$\sum_{k=1}^{K} N_k\, d_k \;<\; N_A\, d_K. \tag{58}$$

Assumption G.7 (NFE vs. Transportation Cost)**.** For this computational model, assume that the evaluation budget at a chosen solver accuracy is proportional to the prescribed coupling-cost bound $L = \mathbb{E}[|x_1 - x_0|^2]$. This cost serves as a proxy for the complexity of the vector field.

$$N_i = C \cdot L_i \tag{59}$$

where the same $C > 0$ is assumed for the direct model and each stage at the chosen solver accuracy. Expected evaluation budgets may be used in this model. Appendix §H reports interval-cost measurements and complete-sampler comparisons.

Under this assumption, (58) is equivalent to

$$\sum_{k=1}^{K} L_k\, d_k \;<\; L_A\, d_K. \tag{60}$$

Define the *computational load* of a transport as Load $:= L \times d$. Then

$$\text{Load}_A \;=\; L_A d_K, \qquad \text{Load}_B \;=\; \sum_{k=1}^{K} L_k d_k,$$

and proving (60) is equivalent to showing $\text{Load}_B < \text{Load}_A$.

G.2.2 Analysis of Computational Load

We use the transportation-cost decompositions established in Section G.1. Writing $\tilde{\epsilon}^{(k)}$ for the signal component associated with scale/stage k:

Direct model (single stage). The load is the product of its transportation cost L_A((55)) and the dimensionality of the final image d_K.

$$\begin{aligned} \text{Load}_A &= L_A \cdot d_K \\ &= \left(\left(\sum_{j=1}^{K} \mathbb{E}[|\tilde{\epsilon}^{(j)}|^2] \right) + d_K \sigma^2 \right) d_K \\ &= d_K \sum_{j=1}^{K} \mathbb{E}[|\tilde{\epsilon}^{(j)}|^2] + d_K^2 \sigma^2 \end{aligned} \tag{61}$$

Cascaded model (multi-stage). Using the native-grid factors in Eq. (53),

$$\text{Load}_B = \sum_{k=1}^{K} 4^{-(K-k)} \mathbb{E}\|\tilde{\epsilon}^{(k)}\|^2 d_k + d_1 \sigma^2 \sum_{k=1}^{K} d_k. \tag{62}$$

G.2.3 Main Proof and Conclusion

We compare (61) and (62) by splitting into signal and noise parts.

(1) Signal component. Since $4^{-(K-k)} \le 1$ and $d_k \le d_K$,

$$\sum_k 4^{-(K-k)} \mathbb{E}\|\tilde{\epsilon}^{(k)}\|^2 d_k \le d_K \sum_k \mathbb{E}\|\tilde{\epsilon}^{(k)}\|^2. \tag{63}$$

(2) Noise component. For $d_k = 4^{k-1} d_1$ and $\sigma_k = 2^{-(k-1)}\sigma$,

$$\text{Load}_{B,\text{noise}} = d_1^2 \sigma^2 \frac{4^K - 1}{3} < 16^{K-1} d_1^2 \sigma^2 = \text{Load}_{A,\text{noise}}, \qquad K \ge 2. \tag{64}$$

Combining (1) and (2). These inequalities give $\sum_k L_k d_k < L_A d_K$. Equivalently, Theorem 3.1 gives $\sum_k d_k L_k \le d_K L_B < d_K L_A$. Multiplying by the common positive constants αC proves the evaluation-time comparison.

Complete-sampler timings in Appendix H.3 include resizing and noise injection and use the measured budgets for each tested solver.

H Sampling-Time Measurements and Stage-Wise Efficiency

Theorem M.1 uses the per-evaluation cost and budget models in Appendix §G.2. We report individual-call latency and interval-cost measurements, then evaluate complete-sampler time against a matched single-stage pixel baseline.

H.1 Per-Evaluation Cost across Resolutions

Under our experimental setting (single H100 GPU, batch size 1, FP16, 200-call rolling average), Table 9 reports the per-function-evaluation cost of the unified DiT across the resolutions used by CDC-FM. The measured time grows approximately $4\times$ when the spatial resolution doubles, matching the fourfold growth in ambient image dimension at fixed channel count.

Table 9: Per-function-evaluation cost across resolutions: a single forward pass, averaged over 200 calls on one H100 at batch size one and FP16.

Image resolution	32^2	64^2	128^2
Time / function-eval (ms)	5.6	22.0	87.4
Normalized to d_1	$1.0\times$	$\approx 4.0\times$	$\approx 16.0\times$

Table 10: SiT interval costs and reported evaluation budgets at target FID 1.85. Each interval uses 1,000 sampled pairs of intermediate states from a linear interpolant.

(t_1, t_2)	$(0.0, 1.0)$	$(0.0, 0.75)$	$(0.0, 0.5)$	$(0.0, 0.25)$	$(0.25, 0.5)$
Transport cost C	1.00	0.563	0.250	0.063	0.063
Required NFE	64	36	16	4	4
NFE / C	64	63.9	64.0	63.5	63.5

H.2 Interval Costs and Evaluation Budgets

For the interval-cost study, we construct intermediate states by linearly interpolating between source and target along $(t_1, t_2) \in [0, 1]^2$ and estimate $C(t_1, t_2) = \mathbb{E}[|x_{t_2} - x_{t_1}|^2]$ from 1,000 sampled pairs. Using SiT [6] as the generative model, we gradually increase the NFE until reaching a fixed target FID of 1.85, and report the resulting NFE in Table 10.

The required NFE scales tightly with $C(t_1, t_2)$: the ratio NFE/C stays in a narrow band $(63.5-64.0)$ across very different choices of (t_1, t_2) that span an order-of-magnitude range in C.

H.3 Single-Stage vs. Multi-Stage Wall-Clock Comparison

To evaluate the complete multi-stage construction, we compare CDC-FM against a single-stage pixel baseline that is matched in backbone, optimizer, training schedule, classifier-free guidance, and ODE solver, with $K = 1$. As shown in Table 11, at similar FID/IS the single-stage baseline incurs $\sim 1.7\times$ the wall-clock of CDC-FM, with the hierarchical model allocating its evaluations across resolutions: low-level information is generated cheaply at low-resolution stages, and high-resolution stages benefit from the noised low-resolution outputs rather than re-generating all information at full resolution.

H.4 Additional Ablations: Stage Path and Per-Stage NFE Allocation

Stage count and resolution path. Table 12 ablates the stage count K and the resolution path. Excessive per-stage ratios (rows 1–2) hurt FID; with moderate ratios, K=3 remains close to K=4, while K=4 attains the best FID. The additional 16^2-stage path is evaluated separately in Appendix F.2.

Per-stage NFE allocation. Table 13 examines the sensitivity of FID to per-stage NFE. Reducing the NFE of any single stage hurts FID, but the degradation is substantially larger at later, higher-resolution stages (e.g. FID jumps from 1.81 to 1.94 when N_4 drops from 16 to 4, vs. 1.81 to 1.85 when N_1 drops from 16 to 4). This indicates that solver-step budget should be biased toward high-resolution stages.

H.5 Additional Multi-Stage Speed Comparison

Table 1 compares benchmark quality and sampling time across model families. The controlled CDC-FM/$K = 1$ comparison uses one A100, batch size 32, and includes all stages, both CFG evaluations, resizing, and noise injection. The H100, batch-size-one, FP16 protocol measures individual forward evaluations in Appendix H.1.

H.6 Solver sensitivity at a common fidelity threshold

For each checkpoint and solver, the study reports the smallest tested budget attaining FID ≤ 1.83. Each row compares CDC-FM and $K = 1$ under the same solver. NFE includes both CFG branches and every solver sub-evaluation. Heun uses six steps per stage with two sub-evaluations, RK4

Table 11: Matched pixel-space comparison. Both models share the backbone, optimizer, training schedule, CFG rule, and Euler solver. Time is seconds per image on one A100 at batch size 32. Total NFE includes both CFG calls.

Method	FID ↓	IS ↑	NFE	Speed (s/img) ↓
Single-stage Pixel (K=1)	1.82	305.8	500	8.73
CDC-FM, $4 \times 4 \times 2$	1.97	298.2	32	1.31
CDC-FM, $8 \times 4 \times 2$	1.87	301.8	64	2.66
CDC-FM (Ours)	1.81	304.3	128	**5.18**

Table 12: Impact of the stage count K and the resolution path on ImageNet-1k/256. Excessively large per-stage resolution ratios degrade FID (rows 1–2). With moderate ratios, fewer stages remain viable (rows 3–4): the gap between K=3 and K=4 is small, while K=4 obtains the best FID. All rows use 16 Euler steps per stage and CFG scale three, so total NFE is $32K$.

Stages	Resolution Path	Total NFE	FID ↓
2	$32 \to 256$	64	2.06
2	$64 \to 256$	64	1.97
3	$32 \to 128 \to 256$	96	1.83
4	$32 \to 64 \to 128 \to 256$	128	**1.81**

four steps with four sub-evaluations. Dopri5 uses relative and absolute tolerances 10^{-5} and reports rounded mean NFE, counting accepted and rejected evaluations. Time follows the A100 batch-size-32 end-to-end protocol. Table 14 gives all values.

H.7 Noise-decay and interpolation settings

The five-point noise study fixes $\sigma = 1$, the four-stage hierarchy, 128 total NFE, CFG scale three, and a 50,000-sample evaluator. Each γ configuration is trained separately with the same backbone, data, and training budget. Table 15 reports the FIDs. The interpolant comparison in Table 3 uses this same hierarchy and evaluation budget, with linear, trigonometric, or variance-preserving-shaped deterministic weights. Its VP-shaped row evaluates an interpolant path and ODE sampler.

I Higher-Resolution Results on ImageNet-1k/512

To assess CDC-FM at higher resolution, we report ImageNet-1k/512 class-conditional generation results in Table 16. The same backbone, training pipeline, and CFG schedule are used as for the 256^2 experiments, with the resolution hierarchy extended accordingly. The table compares quality with the full set of reference methods at this resolution.

J Algorithm

Here we present the detailed unconditional training and inference procedures for the unified multi-stage generative model b^θ with **conditional dependent coupling** in Algorithm 1 and Algorithm 2. The inference steps are described using the forward Euler method [24, 25] for solving the ODE, as a simple example of a numerical solver.

K Classifier-Free Guidance for Conditional Dependent Coupling

Rationale. Each refinement stage receives a noisy coarse image and targets clean images at its resolution. We apply the same CFG rule [35] at all stages. Figures 4, 5, and 9 evaluate the shared guidance scale.

Training with conditional dropout (classifier-free). We train the unified DiT b^θ exactly as in the flow-matching objective shown in (12), with the only change that the conditioning signal is randomly

Table 13: Per-stage Euler-step allocation ablation on ImageNet-1k/256 with the $32 \to 64 \to 128 \to 256$ hierarchy. We hold three of the per-stage step counts at 16 and reduce the fourth to $\{8, 4\}$. Reducing the NFE of any single stage degrades FID, but the degradation is substantially larger at later, higher-resolution stages, showing greater sensitivity to the tested step reductions at later stages. Total NFE includes both CFG evaluations per step.

N_1	N_2	N_3	N_4	Total NFE	FID ↓
16	16	16	16	128	**1.81**
8	16	16	16	112	1.83
4	16	16	16	104	1.85
16	8	16	16	112	1.85
16	4	16	16	104	1.87
16	16	8	16	112	1.88
16	16	4	16	104	1.91
16	16	16	8	112	1.90
16	16	16	4	104	1.94

Table 14: Solver sensitivity at FID ≤ 1.83. Time is CDC-FM/$K = 1$ under the same solver.

Solver	$K = 1$ FID	NFE	CDC-FM FID	NFE	Time ratio
Euler	1.82	500	1.81	128	0.59
Heun	1.83	320	1.82	96	0.63
RK4	1.82	256	1.82	128	0.67
Dopri5	1.83	280	1.82	120	0.70

dropped. Let $\mathbf{c}$ denote the conditioning signal (e.g., class label or text prompt), and let $\varnothing$ denote the null condition. Define the mixed conditioning

$$\bar{\mathbf{c}} \sim q(\bar{\mathbf{c}} \mid \mathbf{c}) \;=\; (1 - p_{\varnothing})\, \delta_{\mathbf{c}} \;+\; p_{\varnothing}\, \delta_{\varnothing}, \tag{65}$$

where $p_{\varnothing} \in (0, 1)$ is the dropout probability and $\delta_{\mathbf{c}}$ and $\delta_{\varnothing}$ are point masses at the paired image's class and the null condition. The training loss becomes

$$\mathcal{L}_{\text{cfg}}(\theta) = \mathbb{E}_{\substack{t\sim[0,1],\, k,\, e_k,\\ (x_0^{(k)}, x_1^{(k)})\sim\rho^{(k)}(x_0^{(k)}, x_1^{(k)} \mid x^{(K)}),\\ \bar{\mathbf{c}}\sim q(\cdot\mid\mathbf{c})}} \Big[\big\| b^{\theta}\big(I_\tau^{(k)}, \tau, e_k, \bar{\mathbf{c}}\big) \big\|^2 \;-\; 2\, \dot{I}_\tau^{(k)} \cdot b^{\theta}\big(I_\tau^{(k)}, \tau, e_k, \bar{\mathbf{c}}\big) \Big], \tag{66}$$

with $I_\tau^{(k)} = (1 - \tau) x_0^{(k)} + \tau x_1^{(k)}$, $\dot{I}_\tau^{(k)} = x_1^{(k)} - x_0^{(k)}$, and $\tau = \frac{t - t_0^k}{t_1^k - t_0^k}$ as in Sec. 3.3.

CFG at inference: guided vector field. Let the *conditional* and *unconditional* vector-field predictions be

$$b_{\mathbf{c}}^{\theta}(z, \tau, e_k) \;=\; b^{\theta}(z, \tau, e_k, \mathbf{c}), \qquad b_{\varnothing}^{\theta}(z, \tau, e_k) \;=\; b^{\theta}(z, \tau, e_k, \varnothing). \tag{67}$$

Given a global guidance scale $S_{\text{cfg}} \geq 0$ that is *shared across all stages* k, we define the CFG-guided field by the standard linear rule [35]:

$$\begin{aligned} b_{\text{cfg}}^{\theta}(z, \tau, e_k; \mathbf{c}, S_{\text{cfg}}) \;&=\; b_{\varnothing}^{\theta}(z, \tau, e_k) \;+\; S_{\text{cfg}} \Big(b_{\mathbf{c}}^{\theta}(z, \tau, e_k) - b_{\varnothing}^{\theta}(z, \tau, e_k) \Big) \\ &=\; (1 - S_{\text{cfg}})\, b_{\varnothing}^{\theta} + S_{\text{cfg}}\, b_{\mathbf{c}}^{\theta}. \end{aligned} \tag{68}$$

This recovers unconditional generation when $S_{\text{cfg}} = 0$, the nominal conditional model when $S_{\text{cfg}} = 1$, and stronger condition-following for $S_{\text{cfg}} > 1$.

Stage-wise conditional generation under coupling. At inference, the state $X_\tau^{(k)}$ in stage k evolves under the guided ODE driven by (68):

$$\frac{\mathrm{d}}{\mathrm{d}\tau} X_\tau^{(k)} \;=\; b_{\text{cfg}}^{\theta}\big(X_\tau^{(k)}, \tau, e_k; \mathbf{c}, S_{\text{cfg}}\big), \qquad \tau \in [0, 1], \tag{69}$$

with initial condition

$$X_0^{(1)} = x_0^{(1)} \sim \mathcal{N}(0, \sigma^2 I_{d_1}), \qquad X_0^{(k)} \;=\; U_k\big(\hat{x}_1^{(k-1)}\big) + \sigma_k \zeta_k, \;\; \zeta_k \sim \mathcal{N}(0, I_{d_k}), \;\; k \geq 2, \tag{70}$$

Table 15: Five-point noise-decay study at $\sigma = 1$ and 128 total NFE.

Decay γ	1.0	1.5	2.0	2.5	3.0
FID	3.34	2.65	1.81	2.03	2.86

Table 16: Comparison on ImageNet-1k/512 conditional generation.

Model/Method	Params	FID ↓	IS ↑
BigGAN [37]	-	8.43	177.9
ADM [40]	554M	7.72	172.7
VDM++ [11]	2B	2.65	278.1
DiT-XL/2 [5]	675M	3.04	240.8
U-ViT [9]	501M	4.05	263.8
SiD2 [10]	2.0B	2.19	-
MaskGIT [51]	227M	7.32	156.0
MAGVIT-v2 [56]	307M	3.07	324.3
MAR-L [12]	481M	2.74	205.2
VQGAN [49]	1.4B	26.52	66.8
VAR-d36-s [8]	2.0B	2.63	303.2
PaGoDA [20]	0.9B	1.80	251.3
CDC-FM	684M	2.65	307.4

where the next coarse image is the preceding generated endpoint, as in Eq. (17). The stage output is the terminal state

$$\hat{x}_1^{(k)} \;=\; X_1^{(k)} \;=\; X_0^{(k)} \;+\; \int_0^1 b^\theta_{\text{cfg}}\big(X_t^{(k)}, t, e_k; \mathbf{c}, S_{\text{cfg}}\big)\,\mathrm{d}t, \tag{71}$$

and the full cascade is obtained by the recursion

$$\hat{x}_1^{(1)} \;=\; X_1^{(1)}, \qquad \hat{x}_1^{(k)} \;=\; X_1^{(k)}\big(\,U_k(\hat{x}_1^{(k-1)}) + \sigma_k\zeta_k\,\big), \quad k = 2, \ldots, K, \tag{72}$$

which is the same Markov structure as (16) but with the guided field b^θ_{cfg}.

Guidance through the refinement stages. At each refinement, the source combines the preceding generated image with fresh noise and the shared field also receives the class label. Conditional information therefore enters the velocity at every resolution. Figure 9 reports similar locations of the lowest FID across stages under the evaluated CFG sweep. The main model uses $S_{\text{cfg}} = 3$ throughout.

Evaluation counts. A guided Euler step uses two network calls, so the total NFE is $2\sum_k N_k$. Heun and RK4 multiply this by their two and four field sub-evaluations. Dopri5 counts every accepted and rejected field evaluation, including both CFG branches. At guidance scale one, the field equals the conditional velocity used in the ideal-marginal argument.

Stochastic extension and algorithms. For an SDE parameterization, the same rule (68) is applied to the predicted score/drift in place of b^θ:

$$s^\theta_{\text{cfg}}(z, t, e_k; \mathbf{c}, S_{\text{cfg}}) = s^\theta_\varnothing(z, t, e_k) + S_{\text{cfg}}\big(s^\theta_{\mathbf{c}}(z, t, e_k) - s^\theta_\varnothing(z, t, e_k)\big)\,,$$

and the sampler integrates the corresponding guided stochastic dynamics with $X_0^{(k)}$ as in (70). Following the CFG strategy introduced in this section for conditional generation, we extend the unconditional generation procedure described in Appendix §J, and provide the training and inference procedures of the unified multi-stage generative model b^θ with CFG as shown below.

L Implement Details

We report the implementation details of our model in Table 17.

Training is conducted on a platform equipped with 32 × A100 GPUs (80GB). Under the configuration shown below, the global training batch size is $32 \times 8 = 256$. We train for 200 epochs; for ImageNet-1k, which contains 1.28M samples, each epoch consists of approximately 5000 steps. Thus, the full training schedule corresponds to roughly 1M iterations (a metric used in several prior works). With this setup, the total training time is approximately 76 hours, averaging 0.38 hours per epoch. During training, each A100 GPU consumes about 43GB of memory.

For inference, we use a single A100 GPU and set the batch size to 32 for CDC-FM and the matched single-stage pixel baseline. Timing includes the complete sampler, both CFG calls, resizing, and

Algorithm 1 Training a unified multi-stage Flow-Matching model b^θ (ODE view)

Require: Dataset $\mathcal{D} = \{x^{(K)}\} \sim p_{\text{data}}$, stages $k \in \{1, \dots, K\}$ with dimensions d_k, feature-map resolutions r_k and resolution embedding $e_k = E(r_k)$; down/upsamplers $D_k : \mathbb{R}^{d_k} \to \mathbb{R}^{d_{k-1}}$ and $U_k : \mathbb{R}^{d_{k-1}} \to \mathbb{R}^{d_k}$, composite downsampler $D_{K\to k}$; time partition $0 = t_0^1 < t_1^1 = t_0^2 < \cdots < t_1^K = 1$; noise scales σ (stage 1) and $\{\sigma_k\}_{k=2}^K$; batch size B.

Comment: Ground-truth at stage k is $x_1^{(k)} = D_{K\to k}(x^{(K)}) \sim \rho_1^{(k)}(\cdot \mid x^{(K)})$.

for each training step **do**
  for $i = 1$ **to** B **do**
    Sample $x^{(K)} \sim p_{\text{data}}$; sample $t \sim \mathcal{U}[0, 1]$.
    Find k such that $t \in [t_0^k, t_1^k]$; set the rescaled time $\tau \leftarrow \dfrac{t - t_0^k}{t_1^k - t_0^k} \in [0, 1]$.
    Target at stage k: $x_1^{(k)} \leftarrow D_{K\to k}(x^{(K)})$ ▷ $x_1^{(k)} \sim \rho_1^{(k)}(\cdot \mid x^{(K)})$
    if $k = 1$ **then** ▷ Stage 1: noise-to-image coupling is independent
      Sample $x_0^{(1)} \sim \mathcal{N}(0, \sigma^2 I_{d_1})$ ▷ $\rho_0^{(1)}(x_0^{(1)})$
    else ▷ Stage $k > 1$: conditional dependent coupling
      Sample $\zeta_k \sim \mathcal{N}(0, I_{d_k})$
      $m_k(x_1^{(k)}) \leftarrow U_k\big(D_k(x_1^{(k)})\big)$
      $x_0^{(k)} \leftarrow m_k(x_1^{(k)}) + \sigma_k \zeta_k$ ▷ $\rho_0^{(k)}(\cdot \mid x_1^{(k)}) = \mathcal{N}(m_k, \sigma_k^2 I)$
    end if
    Linear interpolant: $I_\tau^{(k)} \leftarrow (1 - \tau)\, x_0^{(k)} + \tau\, x_1^{(k)}$
    Target velocity: $\dot{I}_\tau^{(k)} \leftarrow x_1^{(k)} - x_0^{(k)}$ ▷ constant in τ for linear path
    $e_k \leftarrow E(r_k)$ ▷ resolution embedding
    $u_\theta \leftarrow b^\theta\big(I_\tau^{(k)}, \tau, e_k\big)$ ▷ DiT vector field shared across stages
    Per-sample loss: $\ell_i \leftarrow \|u_\theta\|_2^2 - 2\, \dot{I}_\tau^{(k)} \cdot u_\theta$ ▷ equiv. to $\|u_\theta - \dot{I}_\tau^{(k)}\|_2^2$ up to a const.
  end for
  $\mathcal{L}(\theta) \leftarrow \frac{1}{B} \sum_{i=1}^B \ell_i$ ▷ matches $\mathbb{E}\big[\|b^\theta\|^2 - 2\, \dot{I} \cdot b^\theta\big]$
  Update parameters: $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}(\theta)$
end for

noise injection. We generate 50k images for evaluation and report inference speed in seconds per image.

The implementation defaults to 24 steps per stage. The main comparison uses 16 Euler steps per stage ($16 \times 4 \times 2 = 128$ total NFE), with four and eight steps in the lower-budget settings. Class-dropout probability is denoted $p_\varnothing$ in Algorithm 3; stage sampling follows the time partition with probability $q(k) = t_1^k - t_0^k$.

M Theoretical Justification and Key Property

Mathematical premise and intuition. Our method adopts a unified multi-scale, coarse-to-fine image generation procedure that aligns with the natural way humans recognize images [57]. Compared to previous coarse-to-fine AR approaches [57, 55], our method is based on a unified and continuous transformation between distributions in a highly controllable manner, which better reflects the smooth nature of image transitions. Compared to existing multi-stage generation methods [58], our method models a unified multi-stage generation process with **conditional dependent coupling** and can be parameterized by a single DiT that can be trained from end-to-end. Each stage receives clean image targets from the real data hierarchy. The ideal conditional-field result and the measured recursive-source comparison are given in Appendix E. This resolves the generation problem step by step within a unified process.

Diversity and stochasticity. In image generation, variation in coarse image structure is established in the early stages (initial transition steps for ODE/SDE) [16, 8]. For the proposed method, this corresponds to the transition modeled in the initial and intermediate stages. During these stages, key attributes like object presence, layout, and structure are determined. In contrast, the subsequent

Algorithm 2 Inference via sequential multi-stage ODE integration (forward Euler)

Require: Trained vector field b^θ; stages $k \in \{1, \dots, K\}$ with dimensions d_k, feature-map resolutions r_k and resolution embedding $e_k = E(r_k)$; upsamplers $U_k : \mathbb{R}^{d_{k-1}} \to \mathbb{R}^{d_k}$; noise scales σ (stage 1) and $\{\sigma_k\}_{k=2}^K$; per-stage step counts $\{N_k\}$.

Comment: The Markovian cascade uses $\hat{x}_0^{(1)} \sim \mathcal{N}(0, \sigma^2 I_{d_1})$ and, for $k \geq 2$, $\hat{x}_0^{(k)} = U_k(\hat{x}_1^{(k-1)}) + \sigma_k \zeta_k$ with $\zeta_k \sim \mathcal{N}(0, I_{d_k})$.

Initialize stage 1 (noise-to-image): Sample $\hat{x}_0^{(1)} \sim \mathcal{N}(0, \sigma^2 I_{d_1})$; set $\hat{X}_0^{(1)} \leftarrow \hat{x}_0^{(1)}$.
for $k = 1$ **to** K **do**
  $\Delta\tau \leftarrow 1/N_k$; $e_k \leftarrow E(r_k)$
  if $k > 1$ **then** ▷ Conditional dependent coupling from previous stage
    Sample $\zeta_k \sim \mathcal{N}(0, I_{d_k})$
    $\hat{x}_0^{(k)} \leftarrow U_k\big(\hat{x}_1^{(k-1)}\big) + \sigma_k \zeta_k$
    $\hat{X}_0^{(k)} \leftarrow \hat{x}_0^{(k)}$
  end if
  for $n = 0$ **to** $N_k - 1$ **do**
    $\tau \leftarrow n\,\Delta\tau$
    $\hat{v} \leftarrow b^\theta\big(\hat{X}_\tau^{(k)}, \tau, e_k\big)$ ▷ ODE velocity at rescaled time $\tau \in [0, 1]$
    Euler step: $\hat{X}_{\tau+\Delta\tau}^{(k)} \leftarrow \hat{X}_\tau^{(k)} + \Delta\tau \cdot \hat{v}$
  end for
  Stage output: $\hat{x}_1^{(k)} \leftarrow \hat{X}_1^{(k)}$ ▷ resolution d_k
end for
return $\hat{x}_1^{(K)} \in \mathbb{R}^{d_K}$ ▷ final high-resolution sample

high-resolution refinement stages require less diversity. By employing the proposed **conditional dependent coupling** strategy, we explicitly decouple the diverse coarse image generation from the fine image refinement, thereby achieving both high performance and efficiency. See Appendix §C for the visualization of the multi-stage generation and the details hierarchy it forms. Specifically, as the stage index k increases, we gradually reduce the stage-dependent noise level σ_k in (8). The schedule controls source variation around the inherited structure and is defined as

$$\sigma_k = \gamma^{-(k-1)}\sigma, \quad \text{where } \gamma \geq 1, \quad 2 \leq k \leq K. \tag{73}$$

We refer to γ as the "diminish factor" and further analyze its impact in our experiments.

Few-step generation and Efficient inference. The unified multi-stage method with **conditional dependent coupling** decomposes generation into a coarse-generation task and successive refinement tasks. The experiments evaluate the resulting sampling budgets [59, 60]. Moreover, the simulations in the early stages are substantially faster than those in the final stage, which targets the full-resolution image. Quantitatively, this construction gives a lower stage-local cost bound, as established in Theorem 3.1. The following conditional computational model relates these stage costs to network-evaluation work. Here we summarize as Theorem M.1 below.

Theorem M.1. *Under Theorem 3.1, assume the same evaluation-cost model $a(d) = \alpha d$ and budget model $N_i = CL_i$ for the direct construction and all stages, with common $\alpha, C > 0$. Then their network-evaluation time components satisfy*

$$T_{B,\text{eval}} = \alpha C \sum_k d_k L_k < \alpha C d_K L_A = T_{A,\text{eval}}. \tag{74}$$

Appendix §G.2 proves the comparison. Appendix §H reports complete-sampler measurements for the tested solvers and budgets.

N Use of Large Language Models (LLMs)

We acknowledge generative AI assistance with language editing, manuscript revision, figure preparation, and mathematical exposition and checking.

Algorithm 3 Training a unified multi-stage Flow-Matching model b^θ with Classifier-Free Guidance

Require: Dataset $\mathcal{D} = \{x^{(K)}, \mathbf{c}\}$ with conditions $\mathbf{c}$ (e.g., class/text); stages $k \in \{1, \dots, K\}$ with dimensions d_k, feature-map resolutions r_k and resolution embedding $e_k = E(r_k)$; down/upsamplers $D_k : \mathbb{R}^{d_k} \to \mathbb{R}^{d_{k-1}}$ and $U_k : \mathbb{R}^{d_{k-1}} \to \mathbb{R}^{d_k}$; composite downsampler $D_{K\to k}$; time partition $0 = t_0^1 < t_1^1 = t_0^2 < \cdots < t_1^K = 1$; noise scales σ (stage 1) and $\{\sigma_k\}_{k=2}^K$; classifier-free dropout probability $p_\varnothing \in (0, 1)$; batch size B.

Comment: Ground-truth at stage k is $x_1^{(k)} = D_{K\to k}(x^{(K)}) \sim \rho_1^{(k)}(\cdot \mid x^{(K)})$. Conditional dropout samples $\bar{\mathbf{c}}$ from $q(\bar{\mathbf{c}}) = (1 - p_\varnothing)\, p(\mathbf{c}) + p_\varnothing\, \delta_\varnothing$.

for each training step **do**
  for $i = 1$ **to** B **do**
    Sample $(x^{(K)}, \mathbf{c}) \sim \mathcal{D}$; sample $t \sim \mathcal{U}[0, 1]$.
    Find k such that $t \in [t_0^k, t_1^k]$; set $\tau \leftarrow \dfrac{t - t_0^k}{t_1^k - t_0^k} \in [0, 1]$.
    Target at stage k: $x_1^{(k)} \leftarrow D_{K\to k}(x^{(K)})$.
    if $k = 1$ **then** ▷ Stage 1: noise-to-image, independent prior
      Sample $x_0^{(1)} \sim \mathcal{N}(0, \sigma^2 I_{d_1})$ ▷ $\rho_0^{(1)}(x_0^{(1)})$
    else ▷ Stage $k > 1$: conditional dependent coupling
      Sample $\zeta_k \sim \mathcal{N}(0, I_{d_k})$
      $m_k(x_1^{(k)}) \leftarrow U_k\big(D_k(x_1^{(k)})\big)$
      $x_0^{(k)} \leftarrow m_k(x_1^{(k)}) + \sigma_k \zeta_k$ ▷ $\rho_0^{(k)}(\cdot \mid x_1^{(k)}) = \mathcal{N}(m_k, \sigma_k^2 I)$
    end if
    Linear interpolant: $I_\tau^{(k)} \leftarrow (1 - \tau)\, x_0^{(k)} + \tau\, x_1^{(k)}$
    Target velocity: $\dot{I}_\tau^{(k)} \leftarrow x_1^{(k)} - x_0^{(k)}$ ▷ constant in τ for linear path
    Conditional dropout: Draw $u \sim \mathcal{U}[0, 1]$; set $\bar{\mathbf{c}} \leftarrow \varnothing$ if $u < p_\varnothing$, else $\bar{\mathbf{c}} \leftarrow \mathbf{c}$.
    $e_k \leftarrow E(r_k)$ ▷ resolution embedding fused with time embedding
    $u_\theta \leftarrow b^\theta\big(I_\tau^{(k)}, \tau, e_k, \bar{\mathbf{c}}\big)$ ▷ DiT vector field shared across stages
    Per-sample loss: $\ell_i \leftarrow \|u_\theta\|_2^2 - 2\, \dot{I}_\tau^{(k)} \cdot u_\theta$ ▷ matches (66)
  end for
  $\mathcal{L}_{\text{cfg}}(\theta) \leftarrow \frac{1}{B} \sum_{i=1}^B \ell_i$
  Update parameters: $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{\text{cfg}}(\theta)$
end for

O Discussion

First-stage prior. Stage 1 uses an independent Gaussian source, $\mathcal{N}(0, \sigma^2 I_{d_1})$, to generate the coarsest image. Conditional dependent coupling supplies the structured sources for the subsequent refinement stages.

Operators and computation. Theorem 3.1 analyzes aligned averaging/replication and the noise schedule in Assumptions G.3 and G.5. Theorem M.1 relates these bounds to network-evaluation time through the common cost and budget constants in Assumptions G.6 and G.7. Appendix H reports per-call latency and complete-sampler time, including resizing and noise injection, for the evaluated hardware and solvers.

Deterministic sampling. CDC-FM learns the stage velocity fields and samples by ODE integration. Appendix D gives the general stochastic interpolant formalism. The interpolant ablation compares the deterministic paths used with the ODE sampler.

Resolution hierarchy. The main 256^2 model uses $32 \to 64 \to 128 \to 256$ with $K = 4$. Appendix H.4 compares the FID and evaluation budgets of the tested $K = 2$, $K = 3$, and $K = 4$ hierarchies. Moderate per-stage resolution changes retain similar FID, while larger jumps reduce quality.

Algorithm 4 Inference via sequential multi-stage ODE integration with Classifier-Free Guidance

Require: Trained vector field b^θ; user condition $\mathbf{c}$; global guidance scale $S_{\text{cfg}} \geq 0$ (shared across stages); stages $k \in \{1, \dots, K\}$ with dimensions d_k, feature-map resolutions r_k and resolution embedding $e_k = E(r_k)$; upsamplers $U_k : \mathbb{R}^{d_{k-1}} \to \mathbb{R}^{d_k}$; noise scales σ (stage 1) and $\{\sigma_k\}_{k=2}^K$; per-stage step counts $\{N_k\}$.
Comment: Guided field $b^\theta_{\text{cfg}}(z, \tau, e_k; \mathbf{c}, S_{\text{cfg}}) = (1 - S_{\text{cfg}}) b^\theta_\varnothing(z, \tau, e_k) + S_{\text{cfg}} b^\theta_{\mathbf{c}}(z, \tau, e_k)$, cf. (68).

Initialize stage 1 (noise-to-image): Sample $\hat{x}_0^{(1)} \sim \mathcal{N}(0, \sigma^2 I_{d_1})$; set $\hat{X}_0^{(1)} \leftarrow \hat{x}_0^{(1)}$.
for $k = 1$ **to** K **do**
  $\Delta\tau \leftarrow 1/N_k$; $e_k \leftarrow E(r_k)$
  if $k > 1$ **then** ▷ Conditional dependent coupling from previous stage
    Sample $\zeta_k \sim \mathcal{N}(0, I_{d_k})$
    $\hat{x}_0^{(k)} \leftarrow U_k\big(\hat{x}_1^{(k-1)}\big) + \sigma_k \zeta_k$
    $\hat{X}_0^{(k)} \leftarrow \hat{x}_0^{(k)}$
  end if
  for $n = 0$ **to** $N_k - 1$ **do**
    $\tau \leftarrow n\,\Delta\tau$
    Unconditional: $\hat{v}_\varnothing \leftarrow b^\theta\big(\hat{X}_\tau^{(k)}, \tau, e_k, \varnothing\big)$
    Conditional: $\hat{v}_{\mathbf{c}} \leftarrow b^\theta\big(\hat{X}_\tau^{(k)}, \tau, e_k, \mathbf{c}\big)$
    Guided field: $\hat{v}_{\text{cfg}} \leftarrow (1 - S_{\text{cfg}})\hat{v}_\varnothing + S_{\text{cfg}}\hat{v}_{\mathbf{c}}$ ▷ S_{cfg}=0 uncond., S_{cfg}=1 nominal cond.
    Euler step: $\hat{X}_{\tau+\Delta\tau}^{(k)} \leftarrow \hat{X}_\tau^{(k)} + \Delta\tau \cdot \hat{v}_{\text{cfg}}$
  end for
  Stage output: $\hat{x}_1^{(k)} \leftarrow \hat{X}_1^{(k)}$ ▷ resolution d_k
end for
return $\hat{x}_1^{(K)} \in \mathbb{R}^{d_K}$ ▷ final high-resolution sample

Impact Statement

This paper presents a framework for efficient, high-fidelity image generation operating directly in pixel space. Our goal is to advance the field of generative models by mitigating the computational trade-offs inherent in existing diffusion and flow-based methods.

There are two primary societal consequences of our work. First, by improving the efficiency of high-resolution generation, our method has the potential to democratize access to powerful generative tools, reducing the energy consumption and hardware barriers associated with training and inference. Second, like other high-fidelity generative models, our method carries the risk of misuse for creating deceptive content (e.g., Deepfakes) or propagating biases present in the training data (e.g., ImageNet). We encourage the research community to continue developing robust detection methods and fairness-aware training protocols to mitigate these risks.

Table 17: Complete Parameter Configuration

Category	Parameter	Value	Description
Model Architecture	num_attention_heads	16	Number of attention heads
	attention_head_dim	72	Dimension of each attention head
	in_channels	3	Input channels (RGB)
	out_channels	3	Output channels (RGB)
	depth	28	Number of transformer layers
	num_classes	1000	Number of ImageNet classes
	patch_size	4	Patch size for patch embedding
	attention_bias	true	Whether to use bias in attention
	embed_dim	1152	Embedding dimension (16×72)
	dropout	0.0	Dropout rate
Scheduler	num_train_timesteps	1000	Number of training timesteps
	num_stages	4	Number of cascade stages
	diminish factor γ	2.0	γ in CDC-FM
Training	lr	1e-4	Learning rate
	weight_decay	0.0	Weight decay
	epochs	200	Number of training epochs
	grad_clip_norm	1.0	Gradient clipping norm
	ema_decay	0.9999	EMA decay rate
	logging_steps	100	Logging frequency
Data	root	/public/datasets/ILSVRC2012/train	Dataset root path
	center_crop	false	Whether to center crop
	resolution	256	Image resolution
	expand_ratio	1.125	Image expansion ratio
	num_workers	8	Number of data loader workers
	batch_size	8	Batch size per GPU
Data Augmentation	RandomHorizontalFlip	true	Random horizontal flip
	RandomCrop	true	Random crop (if not center crop)
	Resize	LANCZOS	Resize interpolation method
	Normalize	[0.5, 0.5, 0.5]	Normalization mean and std
Optimization	optimizer	AdamW	Optimizer type
	Optimizer Hyperparameters	$\beta_1 = 0.9,\ \beta_2 = 0.95,\ \epsilon = 1e^{-6}$	Optimizer Hyperparameters
	Learning rate	$1e^{-4}$	Learning rate
	precision	bfloat16	Training precision
	deterministic_ops	false	Deterministic operations
Inference	num_inference_steps	24	Steps per stage (default)
	CFG strength S_{cfg}	3.0	CFG scale (evaluation)
	num_fid_samples	50000	Number of FID samples

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 present CDC-FM and the matched pixel-space comparison. Theorems 3.1 and M.1 state the cost results and their assumptions.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section O discusses the first-stage prior, the operator/noise and computational assumptions, the deterministic experimental scope, and the empirical hierarchy choice. The solver and recursive-source studies describe the evaluated sampling settings.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Theorems 3.1 and M.1 state their conditions. Appendices G.1 and G.2 give complete proofs with the replication norm factors, stage-local time convention, and common cost/budget constants.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [No]

   Justification: The paper provides the method, training/sampling algorithms, configuration table, and evaluation protocols. Some numerical training and ImageNet-512 sampling settings are not reported.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The paper provides algorithms and experimental settings. Experiment code and checkpoints are not distributed with the paper; ImageNet access follows the dataset provider's terms.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [No]

   Justification: Appendix L and Table 17 report the backbone, optimizer, schedule, preprocessing, and main evaluation budget. Numerical class-dropout and stage probabilities and the complete ImageNet-512 sampling settings are not reported.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: FID and IS use 50,000 generated images per configuration. Statistical uncertainty across independent training runs is not quantified.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix L reports 32 A100 GPUs (80GB), global batch size 256, approximately 200 epochs/one million iterations, 76 hours and 43GB per GPU for the main training run. End-to-end A100 batch-size-32 timing and H100 batch-size-one per-call timing are specified separately.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The work studies class-conditional generation using ImageNet. The Impact Statement discusses generated-image misuse and dataset bias.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The Impact Statement discusses access to image-generation research, possible misleading generated content, and dataset biases, together with detection and fairness considerations.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: No pretrained checkpoint or new dataset is released with the paper. Generated-image misuse is discussed in the Impact Statement.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No]

    Justification: ImageNet and the DiT/SiT architectures are cited. Their license terms and exact asset versions are not listed in the paper.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper introduces a generation method and reports ImageNet experiments. No new dataset or pretrained checkpoint accompanies the paper.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human subjects, so no IRB approval is required.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: Appendix N describes generative AI assistance with manuscript preparation. LLMs are not components of CDC-FM or its evaluator.