Post-Hoc Guidance for Consistency Models by Joint Flow Distribution Learning

Chia-Hong Hsu
Brown University
chia_hong_hsu@brown.edu

Randall Balestriero
Brown University
randall_balestriero@brown.edu

Abstract

Classifier-free Guidance (CFG) lets practitioners trade-off fidelity against diversity in Diffusion Models (DMs). The practicality of CFG is however hindered by DMs sampling cost. On the other hand, Consistency Models (CMs) generate images in one or a few steps, but existing guidance methods require knowledge distillation from a separate DM teacher, limiting CFG to Consistency Distillation (CD) methods. We propose Joint Flow Distribution Learning (JFDL), a lightweight alignment method enabling guidance in a pre-trained CM. With a pre-trained CM as an ordinary differential equation (ODE) solver, we verify with normality tests that the variance-exploding noise implied by the velocity fields from unconditional and conditional distributions is Gaussian. In practice, JFDL equips CMs with the familiar adjustable guidance knob, yielding guided images with similar characteristics to CFG. Applied to an original Consistency Trained (CT) CM that could only do conditional sampling, JFDL unlocks guided generation and reduces FID on both CIFAR-10 and ImageNet-64×64 datasets. This is the first time that CMs are able to receive effective guidance post-hoc without a DM teacher, thus, bridging a key gap in current methods for CMs.

1 Introduction

Diffusion models (DMs) have emerged as a powerful class of generative models, achieving remarkable success in various domains of artificial intelligence [15, 54, 55, 43, 51, 46, 45, 32, 49, 23]. Their ability to generate high-fidelity samples has been demonstrated in tasks such as text-to-image synthesis, speech synthesis, and video generation [62, 17, 58, 19]. These models operate through an iterative denoising process, where they gradually transform a noisy input into a structured data sample [15, 39]. The field of DMs has seen extensive research in areas like denoising schedulers, network architectures, controllability, and distillation techniques aimed at improving their performance and efficiency [21, 23, 37, 38, 48, 42, 44, 20, 57].

Classifier-Free Guidance (CFG) [16] is a widely adopted technique that allows for controlling the generation process in DMs. CFG involves jointly training a conditional score and an unconditional score, by often using a "null" label for the unconditional case [16, 2]. A key advantage of CFG is its simplicity and post-hoc nature, as the guidance effect is realized during inference without particularly learning a guided path for generation [16, 47]. By interpolating between the conditional and unconditional predictions using a guidance scale, CFG enables a trade-off between the fidelity of the generated samples and their diversity [22]. Due to its effectiveness in controllability, CFG has become a standard technique in DM applications, with guidance often preferred over unguided generation [47, 61, 18, 22, 42, 5, 40].

Consistency Models (CMs) are a new family of generative models designed for fast sampling, often generating high-quality images in just one or a few steps [53, 52, 26, 25, 63, 56, 29, 13, 30, 6, 36, 59, 14, 56]. They achieve this speed by learning to directly map noisy inputs to clean data samples

Preprint. Under review.

through two main approaches [53]: Consistency Distillation (CD) [30, 40, 63, 31, 10], which involves distilling knowledge from a pre-trained DM, and Consistency Training (CT), where the training procedure is totally data-driven [52, 30, 13, 7, 18]. Existing guidance methods for CMs typically rely on knowledge distillation of CFG from a teacher DM, which inherently binds guidance to the CD approach for CMs [40, 3]. In DMs, the benefits of CFG can be directly evaluated at inference time using the same model [22]. For distilled CMs, however, guidance effects are inherently tied to training, making the learning task more complex than simply fitting an unguided ordinary differential equation (ODE) trajectory [40, 63]. This makes it challenging to directly evaluate the benefits of guidance since it would require training a separate unguided version for a fair comparison. To conclude, current guidance approaches for CMs are largely dependent on the existence of a DM teacher, and the isolated impact of guidance on CMs remains unclear without a direct comparison to another CM that only does unguided generation.

In this paper, we introduce a novel post-hoc guidance method for CMs that operates independently of DMs. Starting with a pre-trained unguided CM as an ODE solver of the diffusion path, our method enables guidance learning by interpolating the directions of synthesized unconditional and conditional distributions. We call this **Joint Flow Distribution Learning (JFDL)** and provide insights of why it works based on the connection to Flow-based generative models (FMs) [27, 33, 59, 12, 35, 34, 11, 24]. Furthermore, we discovered that our algorithm can be adapted to work effectively without an unconditional ODE solver. This broadens the generality of our method for pre-trained CMs as training them can be a complex endeavor. Our algorithm offers a post-hoc way to equip an unguided CM with guidance capabilities, effectively bridging the gap for CT models. We access the effectiveness of guidance by showing significant FID improvements on both CIFAR-10 [28] and ImageNet 64x64 datasets [9] compared to the initial unguided CM. In summary, our contributions are:

- We propose JFDL, a novel post-hoc guidance method for CMs that does not rely on DMs.
- We provide insights to JFDL's effectiveness from a FM's perspective.
- We demonstrate that a pre-trained CM without explicit design for unconditional sampling, remain effective for JFDL and guidance tuning for CMs.
- We demonstrate FID improvements on CIFAR-10 and ImageNet 64x64 by applying our method to CT models.

2 Preliminaries

This section introduces the prelimiaries for the connection between Diffusion models (DMs) and Flow-based models (FMs), Consistency Models (CMs), and Classifier-free Guidance (CFG). We also reuse the established notations for the rest of the paper.

Diffusion Models and Flow-based Models, Two Sides of the Same Coin. DMs are probabilistic generative models that define a forward process which gradually adds noise to a data sample $\mathbf{x}_0 \sim p_{\text{data}}(\mathbf{x})$ over time $t \in [0, 1]$. For the rest of our paper, we focus on the Variance Exploding (VE) scheme [55, 21] that has the stochastic differential equation (SDE), $\mathrm{d}\mathbf{x}_t = \sqrt{2\sigma_t}\mathrm{d}\mathbf{w}_t$, where $\mathbf{w}_t$ is a standard Wiener process, and $\sigma_t = \sigma_{\max} t$ is the noise exploding term with $\sigma_{\max} \gg 1$. The reverse process–generating data from noise–can also be described by an SDE [55]. Notably, there exists a Probability Flow ODE (PF-ODE), that shares the same marginal probability densities as the reverse process, $\mathrm{d}\mathbf{x}_t/\mathrm{d}\sigma_t = -\sigma_t \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) = 1/\sigma_t \cdot (\mathbf{x}_t - \mathbb{E}_{\mathbf{x}_0}[\mathbf{x}_0|\mathbf{x}_t])$, where $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$ is called the score function [26, 53]. A common loss function for predicting the expected value of $\mathbf{x}_0$ has the form:

$$\mathcal{L}_{DM}(\theta) = \mathbb{E}_{t,\mathbf{x}_0,\mathbf{x}_t}\left[w(t)\|D_\theta(\mathbf{x}_t, t) - \mathbf{x}_0\|_2^2\right], \tag{1}$$

where $w(t)$ is a weighting function dependent on time, and θ parametrizes the model D_θ [21]. Other training objectives, such as predicting score, noise, or even a mixture of noise and $\mathbf{x}_0$, can be formulated similarly by reparameterizating the loss term [39, 12, 36].

FMs learn a map between an untractable distribution $p_0(\mathbf{x})$, e.g., $p_{\text{data}}(\mathbf{x})$, and a simple distribution $p_1(\mathbf{x})$, e.g., the standard normal [27, 33, 34]. We define an ODE with the time dependent vector field u_t, and the flow ϕ_t by: $\mathrm{d}\phi_t(\mathbf{x})/\mathrm{d}t = u_t(\phi_t(\mathbf{x}))$, $\phi_0(\mathbf{x}) = \mathbf{x}_0$. Since we do not have access to u_t that satisfies the marginal densities p_t, a per sample aggregation of *conditional vector fields* (CVF)

$u_t(\mathbf{x}_t|\mathbf{x}_0)$ can be used to construct the *conditional flow matching* (CFM) objective [33] as follows:

$$\mathcal{L}_{CFM}(\theta) = \mathbb{E}_{t,\mathbf{x}_0,\mathbf{x}_t}\left[\|F_\theta(\mathbf{x}_t,t) - u_t(\mathbf{x}_t|\mathbf{x}_0)\|_2^2\right], \tag{2}$$

where $\mathbf{x}_t \sim p_t(\mathbf{x}|\mathbf{x}_0)$, and θ parametrizes the model F_θ. If we consider the VE scheme from diffusion, we can construct a *probability path* $p_t(\mathbf{x}|\mathbf{x}_0) = \mathcal{N}(\mathbf{x};\mu_t(\mathbf{x}_0),\sigma_t^2\mathbf{I})$, where $\mu_t(\mathbf{x}) = \mathbf{x}, \sigma_t = \sigma_{\max} t$, and the CVF has the analytical form [33]:

$$u_t(\mathbf{x}_t|\mathbf{x}_0) = \frac{\sigma_t'}{\sigma_t}(\mathbf{x}_t - \mu_t(\mathbf{x}_0)) + \mu_t'(\mathbf{x}_0). \tag{3}$$

With adjustments in parametrization and weighting, previous works have shown that the training target of DMs and FMs are translatable [12, 36]. We will explicitly show the connection of $u_t(\mathbf{x}_t|\mathbf{x}_0)$ and CMs in Section 3.1.

Consistency Models. CMs are a class of generative models designed for one or few-step sampling by learning to directly map any point on the PF-ODE trajectory to its endpoint (the data sample) [53, 26]. In our study, we focus on a family of continuous-time CT models introduced in ECT [13] due to its significantly reduced GPU resources at training. Based on the VE diffusion scheme, the ECT objective is as follows:

$$\mathcal{L}_{ECT}(\theta) = \mathbb{E}_{t,r,\mathbf{x}_0,\mathbf{z}}\left[\, d(G_\theta(\mathbf{x}_0 + \sigma_t\mathbf{z}, t),\ G_{\theta^-}(\mathbf{x}_0 + \sigma_r\mathbf{z}, r))\,\right], \tag{4}$$

where $\mathbf{x}_0 \sim p_{\text{data}}$ is a data sample, $\mathbf{z} \sim \mathcal{N}(\mathbf{0},\mathbf{I})$ is a sampled noise, t, r are consecutive time steps, and $d(\cdot,\cdot)$ is a distance function. G_θ is the online CM, where θ^- denotes the stop gradient of the same CM being the target. The distance $\Delta := |\sigma_t - \sigma_r|$ is approximately σ_t/q^{n}, where n is an integer denoting the training stage. With $q = 2$ and $q = 4$ at the start of CIFAR-10 and ImageNet 64x64 experiments respectively, Δ progressively decreases by a factor of q at training, and the factor at the final stage for both is $q^n = 256$.

Classifier-Free Guidance in DMs and CMs. CFG is a technique used to controlling the guidance of the generation process [16, 40]. In DMs, this is done nearly training-free by extrapolating the conditional and unconditional updates at sampling $D_\theta^{\text{CFG}}(\mathbf{x}_t,t,c,\omega) = \omega\cdot\frac{[\mathbf{x}_t - D_\theta(\mathbf{x}_t,t,c)]}{\sigma_t} + (1-\omega)\cdot\frac{[\mathbf{x}_t - D_\theta(\mathbf{x}_t,t,\emptyset)]}{\sigma_t}$, with $\omega \geq 1$, and $c, \emptyset$ are the conditional and unconditional classes respectively [16]. To achieve CFG in CMs, previous works rely on knowledge distillation of a CFG teacher DM, D_θ^{CFG}, at training, limiting CFG to CD models [63, 40]. We will demonstrate a DM-free CFG method for CMs which uses the pre-trained CM itself as an ODE solver. As a result, we can access the benefits of CFG post-hoc compared to the original CM and equip CT models with guidance.

3 Guidance via Joint-Flow Distribution Learning

We begin by discussing the connection of FMs with CMs. Following, we introduce the naive JFDL for post-hoc guidance tuning, which has a prerequisite of a pre-trained class-conditioned CM that learns the unconditional PF-ODE (with the $\emptyset$ class). We will provide a theoretical analysis of the pseudo-noise in JFDL, supported by experimental verification. In the end, we find out that an adjusted JFDL algorithm works surprisingly well without the need of the $\emptyset$ solver from CM.

3.1 Routine Correspondence between FMs and CMs

Following the VE scheme, let $p_0(\mathbf{x}) = p_{\text{data}}(\mathbf{x})$ be the data distribution, where $\mathbf{x} \in \mathbb{R}^d$, and let $p_0(\mathbf{x}|c) = p_{\text{data}}(\mathbf{x}|c)$ be the class-conditioned data distribution, where $c \in \mathcal{C}$, s.t. $p_0(\mathbf{x}) = \int p_0(\mathbf{x}|c)p(c)\,\mathrm{d}c$.

To train a FM, we begin by sampling $\mathbf{x}_0^c \sim p_{\text{data}}(\mathbf{x}|c)$ and $\mathbf{z} \sim \mathcal{N}(\mathbf{0},\mathbf{I})$. We can construct a class probability path $p_t(\mathbf{x}|c)$ with $p_t(\mathbf{x}|\mathbf{x}_0^c,c) = \mathcal{N}(\mathbf{x};\mathbf{x}_0^c,\sigma_t^2\mathbf{I})$, satisfying $p_t(\mathbf{x}|c) = \int p_t(\mathbf{x}|\mathbf{x}_0^c,c)p_0(\mathbf{x}_0^c|c)\,\mathrm{d}\mathbf{x}_0^c$. Observe that $p_1(\mathbf{x}|\mathbf{x}_0^c,c) \approx \mathcal{N}(\mathbf{x};\mathbf{0},\sigma_{\max}^2\mathbf{I})$ is Gaussian, and $p_0(\mathbf{x}|\mathbf{x}_0^c,c)$ is the class-conditioned data distribution *conditioned on* $\mathbf{x}_0^c$, we can analytically derive the CVF, $u_t(\mathbf{x}_t|\mathbf{x}_0^c,c)$, as the target of the CFM objective. To train a CM, we choose a t, and noisify the sampled data, $\mathbf{x}_t^c = \mathbf{x}_0^c + \sigma_t z$. Then, we can compute the denoising direction as $\mathbf{u} := 1/t\cdot(\mathbf{x}_t^c - \mathbf{x}_0^c)$, and derive the target for the consistency loss as $G_{\theta^-}(\mathbf{x}_t + \mathbf{u}\cdot(\sigma_r - \sigma_t), r, c)$.

Algorithm 1 Naive JFDL for Post-Hoc Guidance

Input: dataset $\mathcal{D}$, pre-trained CM ψ, weighting function $w(t)$, timesteps sampling density $p(t,r)$, total iterations totalIters, max guidance scale ω_{max}, gradnorm layer θ_{gn}, gradnorm function $f(L_{\text{ECT}}, L_{\text{JFDL}} \mid \theta_{\text{gn}})$.
Init: $\theta \leftarrow \psi$, Iters $\leftarrow 0$.
while Iters $<$ totalIters **do**
  Sample $(\mathbf{x}_0^c, c) \sim \mathcal{D}$, $t, r \sim p(t,r)$, $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\omega = 1$;
  $\mathbf{x}_t^c \leftarrow \mathbf{x}_0^c + \sigma_t \mathbf{z}$; $\mathbf{x}_r^c \leftarrow \mathbf{x}_0^c + \sigma_r \mathbf{z}$;
  $L_{\text{ECT}}(\theta) \leftarrow w(t)\, d\big(G_\theta(\mathbf{x}_t^c, t, c, \omega),\ G_{\theta^-}(\mathbf{x}_r^c, r, c, \omega)\big)$; ▷ ECT loss
  Sample $\mathbf{y}_0^c \leftarrow G_{\phi^-}(\mathbf{x}_t^c, t, c)$, $\mathbf{y}_0^{\emptyset,t} \leftarrow G_{\phi^-}(\mathbf{x}_t^c, t, \emptyset)$, $\mathbf{z}' \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\omega \sim \mathcal{U}(1, \omega_{\text{max}})$;
  $\mathbf{y}_t^c \leftarrow \mathbf{y}_0^c + \sigma_t \mathbf{z}'$; $\mathbf{y}_r \leftarrow \mathbf{y}_t^c + \{\omega \left[\frac{\mathbf{y}_t^c - \mathbf{y}_0^c}{\sigma_t}\right] + (1-\omega)\left[\frac{\mathbf{y}_t^c - \mathbf{y}_0^{\emptyset,t}}{\sigma_t}\right]\} \cdot (\sigma_r - \sigma_t)$;
  $L_{\text{JFDL}}(\theta) \leftarrow w(t)\, d\big(G_\theta(\mathbf{y}_t^c, t, c, \omega),\ G_{\theta^-}(\mathbf{y}_r, r, c, \omega)\big)$; ▷ JFDL loss
  $\lambda_{\text{gn}} = f(L_{\text{ECT}}, L_{\text{JFDL}} \mid \theta_{\text{gn}})$;
  $L(\theta) = L_{\text{ECT}}(\theta) + \lambda_{\text{gn}} L_{\text{JFDL}}(\theta)$;
  $\theta \leftarrow \theta - \eta \nabla_\theta L(\theta)$;
  Iters $\leftarrow$ Iters $+ 1$;
end while
return θ

We highlight the correspondence of the routines between FMs and CMs. Under the VE scheme, both CM and FM learns a mapping between the data distribution p_0 and Gaussian p_1. The probability path in FM is equivalent to the noisifying step in CM. Also, we can derive that the CVF is $\sigma_{\text{max}} \cdot \mathbf{z}$ following (3), which equates to the denoising direction in CM. Hence, even without a DM teacher, the ECT objective recovers an accurate approximation of the PF-ODE vector field, just as the CFM objective lets FM estimate the data-marginal velocity field. As we introduce JFDL, we will provide insights from these correspondences into how it effectively captures the mapping associated with the unconditional class distribution.

Joint-Flow Distribution Learning In the variance–exploding (VE) setting, we assume the existence of a *perfect ODE solver*,

$$\text{Solver}\colon\ \mathbb{R}^d \times [0,1]^2 \times \mathcal{C} \longrightarrow \mathbb{R}^d, \quad (\mathbf{x}, t, s, c) \longmapsto \text{Solver}_{t\to s}^c(\mathbf{x}), \tag{5}$$

where the shorthand $\text{Solver}_{t\to s}^c(\mathbf{x}) := \text{Solver}(\mathbf{x}, t, s, c)$ will be used throughout. For class $c \in \mathcal{C}$, time indices $t, s \in [0,1]$, Solver satisfies the push-forward condition

$$\big(\text{Solver}_{t\to s}^c\big)_{\#}\, p_t(\mathbf{x} \mid c) \ =\ p_s(\mathbf{x} \mid c), \tag{6}$$

so that it deterministically transports the class-conditioned distribution from noise level t to level s. Similar to CFG, we introduce a special "null" label $\emptyset \in \mathcal{C}$. Choosing $c = \emptyset$ collapses the conditional path to the *unconditional* path: $p_t(\mathbf{x}|\emptyset) := p_t(\mathbf{x}) \implies (\text{Solver}_{t\to s}^{\emptyset})_{\#} p_t(\mathbf{x}) = p_s(\mathbf{x})$, Thus, the same solver acts as a one-step denoiser (when $s = 0$) for *both* the class-conditioned and $\emptyset$ trajectories. With access to an ideal Solver, we introduce *Joint-Flow Distribution Learning* (JFDL) that constructs a pair of denoising directions, $\mathbf{u}^{\text{cls}}$ and $\mathbf{u}^{\emptyset}$, to embody CFG for the consistency loss. The routine is:

(i) **Draw sample.** Draw a class image $\mathbf{x}_0^c \sim p_0(\mathbf{x}|c)$ and choose a source noise index t.

(ii) **Compute unconditional anchor.** Define $\mathbf{y}_0^{\emptyset,t} := \text{Solver}_{t\to 0}^{\emptyset}\big(\text{Solver}_{0\to t}^c(\mathbf{x}_0^c)\big)$.

(iii) **Noisify.** Add noise by $\mathbf{x}_t^c = \mathbf{x}_0^c + \sigma_t \mathbf{z}$ with $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

(iv) **Compute denoising vectors.** Define $\mathbf{u}^{\text{cls}} := \dfrac{\mathbf{x}_t^c - \mathbf{x}_0^c}{\sigma_t}$ and $\mathbf{u}^{\emptyset} := \dfrac{\mathbf{x}_t^c - \mathbf{y}_0^{\emptyset,t}}{\sigma_t}$.

(v) **Extrapolate guidance.** Extrapolate with ω, define $\mathbf{u}^{\omega} := \omega\, \mathbf{u}^{\text{cls}} + (1-\omega)\, \mathbf{u}^{\emptyset}$.

(vi) **Project to lighter noise for consistency loss.** Pick $r < t$ and let $\mathbf{x}_r = \mathbf{x}_t^c + (\sigma_r - \sigma_t)\, \mathbf{u}^{\omega}$. Compute loss as $d\big(G_\theta(\mathbf{x}_t^c, t, c, \omega),\ G_{\theta^-}(\mathbf{x}_r, r, c, \omega)\big)$.

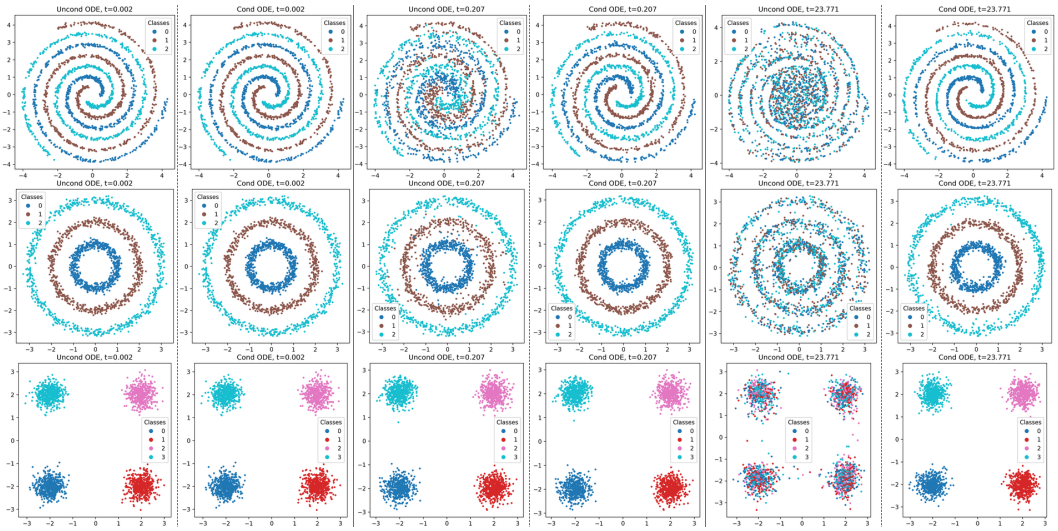


Figure 1: **Comparison of conditional vs. unconditional ODE solutions.** Each row corresponds to a different 2-D toy dataset, *spiral*, *circle*, *Gaussian blob*. Three time steps from left to right, $t = 0.002, 0.207, 23.771$, we show the marginal distribution of a hybrid flow $p(\mathbf{y}_0^{\emptyset,t})$ compared to the marginal distribution of $p(\mathbf{x}_0^c)$.

When $\omega = 1$, the extrapolated direction $\mathbf{u}^\omega = \mathbf{u}^{\text{cls}}$ is no different than the denoising direction for unguided CMs. One one hand, for small noise index t, the *hybrid* ODE solution $\mathbf{y}_0^{\emptyset,t} \approx \mathbf{x}_0^c$, while for large t, the difference is increased, suggesting the distribution deviates more. On the other hand, for any chosen t, the distribution of the hybrid ODE solution integrated over $\mathcal{C}$ is always the data distribution $p_0(\mathbf{x})$. See illustration of Fig 1. Formally, we state that:

Proposition 1 (Hybrid flow preserves the marginal data distribution)**.** *Fix any noise index $t \in [0, 1]$. Draw a class label $c \sim p(c)$ and a clean sample $\mathbf{x}_0^c \sim p_0(\mathbf{x} \mid c)$. Define the hybrid ODE solution $\mathbf{y}_0^{\emptyset,t} := \text{Solver}_{t\to 0}^{\emptyset}\Big(\text{Solver}_{0\to t}^{c}(\mathbf{x}_0^c)\Big)$. We denote the density of $\mathbf{y}_0^{\emptyset,t}$ as $p(\mathbf{y}|c)$, which we define as:*

$$p(\mathbf{y} \mid c) := \big(\text{Solver}_{t\to 0}^{\emptyset}\big)_{\#}\big(\text{Solver}_{0\to t}^{c}\big)_{\#} p_0(\mathbf{x} \mid c). \tag{7}$$

Then the marginal law of $\mathbf{y}_0^{\emptyset,t}$ over the class prior $p(c)$ coincides with the unconditional-class data distribution:

$$\int_{\mathcal{C}} p(c)\, p(\mathbf{y} \mid c)\, dc = p_0(\mathbf{x}). \tag{8}$$

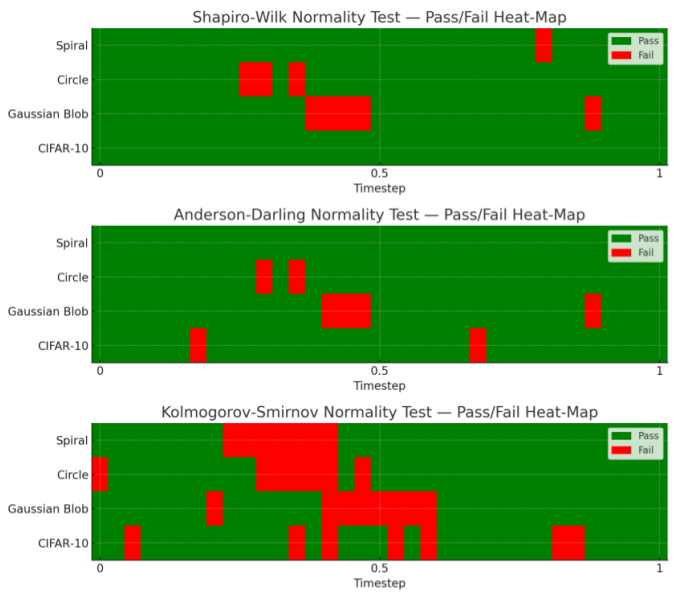
We provide the proof of Prop. 1 in Appendix A. To match JFDL with the FM routine, the denoising direction $\mathbf{u}^\emptyset$ points to the data distribution $p_0(\mathbf{x})$, for any $t \in [0, 1]$. We can further establish its probability path $p_\tau(\mathbf{x}|\mathbf{y}_0^{\emptyset,t})$ to be,

$$\mathcal{N}(\mathbf{x}; \mu_\tau(\mathbf{y}_0^{\emptyset,t}), \sigma_\tau^2\mathbf{I}), \text{ where } \mu_\tau(\mathbf{y}_0^{\emptyset,t}) = \mathbf{y}_0^{\emptyset,t} + \frac{\mathbf{x}_0^c - \mathbf{y}_0^{\emptyset,t}}{t} \cdot \tau,\ \sigma_\tau = \sigma_{\max} \cdot \tau, \tag{9}$$

such that when $\tau = t$, it coincides with $\mathcal{N}(\mathbf{x}; \mathbf{x}_0^c, \sigma_t^2\mathbf{I})$, i.e., *joining* the denoising direction from $p_0(\mathbf{x}|\mathbf{x}_0^c, c)$ at $\mathbf{x}_t^c$. So far, we are left with matching the p_1 distribution. If we can show that $p_1(\mathbf{x}|\mathbf{y}_0^{\emptyset,t}) \approx \mathcal{N}(\mathbf{0}, \sigma_{\max}^2\mathbf{I})$ following the probability path $p_\tau(\mathbf{x}|\mathbf{y}_0^{\emptyset,t})$ in (9), then with Prop. 1, we can imply that the denoising direction $\mathbf{u}^\emptyset$ in JFDL is the CVF from Gaussian to the $\emptyset$-class data distribution. Consequently, JFDL achieves CFG like guidance.

3.2 Pseudo-Noise Analysis and Experimental Verification

In this section, our goal is to show that $p_1(\mathbf{x}|\mathbf{y}_0^{\emptyset,t})$ approximates $\mathcal{N}(\mathbf{0}, \sigma_{\max}^2\mathbf{I})$. In other words, for any t, we sample $\mathbf{x}_0^c \sim p_{\text{data}}(\mathbf{x}|c)$ and compute $\mathbf{y}_0^{\emptyset,t}$ to be the solution of its hybrid ODE following



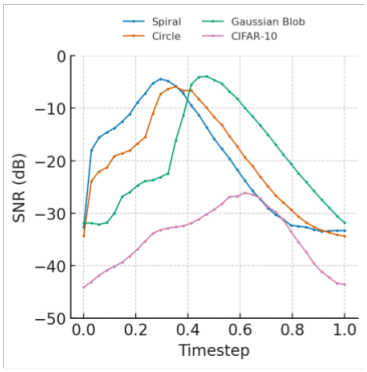


Figure 2: **Normality of pseudo-noise across timesteps.** Heat-maps show pass (green) or fail (red) at $\alpha = 0.05$ for **(top)** Shapiro–Wilk, **(middle)** Anderson–Darling, and **(bottom)** Kolmogorov–Smirnov tests. Rows correspond to the four datasets, *spiral, circle, Gaussian blob, CIFAR-10*. With only a handful of isolated rejections, as well as extremely low SNR ratio, the pseudo-noise is effectively Gaussian at almost every t, supporting the normality assumption.

the JFDL routine. Evaluating the probability path in (9) at $\tau = 1$ gives the random variable,

$$\overbrace{\underbrace{\mathbf{y}_0^{\emptyset,t} + \frac{\mathbf{x}_0^c - \mathbf{y}_0^{\emptyset,t}}{t}}_{\text{mixed-signals}} + \sigma_{\max}\,\mathbf{z}}^{\text{pseudo-noise}}, \qquad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{10}$$

which we will show how $\sigma_{\max}\mathbf{z}$ dominates analytically. Then, we will verify the Gaussianity of (10) experimentally on 2D toy datasets and CIFAR-10 with the signal-to-noise ratio and Gaussianity tests.

Theoretical analysis. Without loss of generality, the clean variables $\mathbf{y}_0^{\emptyset,t}$ and $\mathbf{x}_0^c$ are centered at zero and confined to the hyper-cube $[-1, 1]^d$. With a VE schedule, we can make $\sigma_{\max}$ arbitrarily large, so for any fixed $t \gg 0$ the Gaussian term $\sigma_{\max}\mathbf{z}$ in (10) already dominates, forcing the conditional law at $\tau = 1$ to be close to $\mathcal{N}(\mathbf{0}, \sigma_{\max}^2\mathbf{I})$.

The non-trivial regime emerges in the limit $t \to 0$, where the deterministic drift term $\mu_1(t) := \mathbf{y}_0^{\emptyset,t} + [\mathbf{x}_0^c - \mathbf{y}_0^{\emptyset,t}]/t$ becomes comparable in magnitude. To analyze its contribution, we introduce a function $f(t) := \mathbf{y}_0^{\emptyset,t}$ representing the time-dependent evolution of the hybrid flow, and express the mixed-signals as follows:

Proposition 2 (Mixed-signals as a function of t)**.** *Let* $f(t) := \mathbf{y}_0^{\emptyset,t} = \text{Solver}_{t\to 0}^{\emptyset}\big(\text{Solver}_{0\to t}^{c}(\mathbf{x}_0^c)\big)$ *denote the hybrid flow, we can define the mixed-signals* $g(t)$ *as:*

$$g(t) := f(0) + \frac{f(0) - f(t)}{t}. \tag{11}$$

then, from Taylor expansion to the 2nd order, we have:

$$g(t) \approx f(0) + \frac{t}{2}f''(0) = \mathbf{x}_0^c - \frac{t}{2}(\nabla_{\mathbf{x}_0^c} \log p(c|\mathbf{x}_0^c)). \tag{12}$$

The resulting Taylor expansion of the mixed-signal is surprisingly interpretable near $t = 0^+$. As $\mathbf{x}_0^c$ genuinely belongs to class c, we expect the $p(c|\mathbf{x}_0^c) \approx 1$. The gradient represents how quickly

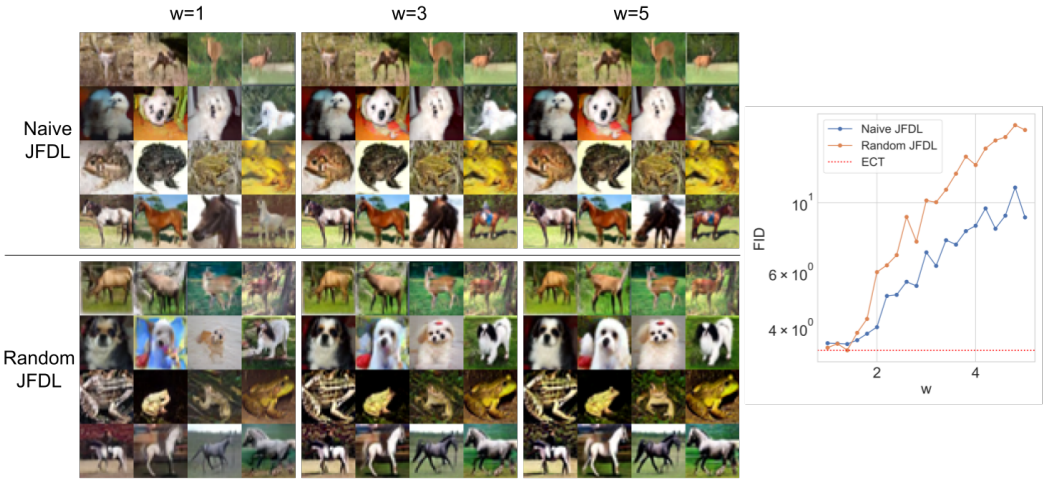


Figure 3: **Preliminary results tuning L_{JFDL} only.** CIFAR-10 samples from Naive JFDL (**top left**) v.s. Random JFDL (**bottom left**). FID w.r.t. ω plot (**right**) reflects the stronger guidance effect from Random JFDL compared to Naive, causing the FID to diverge faster.

the probability would change if the data were perturbed slightly, which would be small if $p(c|\mathbf{x}_0^c)$ is close to the local maximum. We provide the proof of Prop. 2 in Appendix A. Our theoretical support concludes that for any t, $p_1(\mathbf{x}|\mathbf{y}_0^{\emptyset,t})$ approximates a Gaussian.

Experiment Verification In this section, we verify experimentally that for any t, the mixed-signal in (10) will be dominated by the noise term $\sigma_{\max}\mathbf{z}$. To evaluate this claim, we trained a DM and CM as an ODE solver on the 2D toy datasets and CIFAR-10 respectively. Notably, a typical DM/CM is used to solving the ODE backward from $t \to 0$, not forward. To construct a class c sample and its unconditional anchor pair, we adjust the steps of JFDL as follows:

(i) **Draw and noisify.** Draw $\mathbf{x}_0^c \sim p_0(\mathbf{x}|c)$, $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, choose a source noise index t, then noisify $\mathbf{x}_t^c = \mathbf{x}_0^c + \sigma_t \mathbf{z}$.

(ii) **Compute class anchor.** Define $\mathbf{y}_0^c := \text{Solver}_{t\to 0}^c(\mathbf{x}_t^c)$.

(iii) **Compute unconditional anchor.** Define $\mathbf{y}_0^{\emptyset,t} := \text{Solver}_{t\to 0}^{\emptyset}(\mathbf{x}_t^c)$.

With a pre-trained DM/CM as an ODE solver, the routine aligns with the naive JFDL in Alg. 1. Moreover, $\mathbf{y}_0^c$ replaces $\mathbf{x}_0^c$ in the pseudo-noise term. To assess the normality of the pseudo-noise for varying t, we employed standard statistical tests [8, 60], including the Shapiro-Wilk [50], Anderson-Darling [1], and Kolmogorov-Smirnov tests [41], each conducted at a significance level of $\alpha = 0.05$. We also show the log signal-to-noise ratio (SNR), $10 \times \log_{10} \frac{\|\text{mixed-signals}\|^2}{\|\text{pseudo-noise}-\text{mixed-signals}\|^2}$. Our results (Fig. 2) indicate that we can successfully show Gaussianity of the pseudo-noise constructed by the probability path in (9). We will provide the training and testing details of the toy datasets in Appendix C.

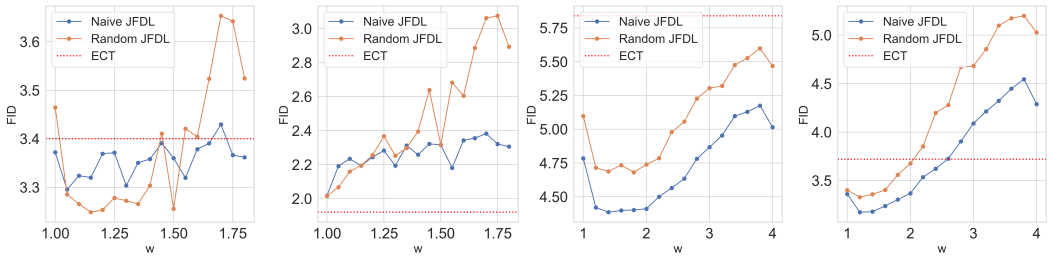
3.3 Preliminary Results

To practice Naive JFDL, we first require a CM that is properly trained to solve for the $\emptyset$–class ODE. We therefore train an ECT baseline that converged with one-step FID of 3.40. Starting from these ECT weights, we appended lightweight guidance embedding layers and fine-tuned exclusively on the JFDL loss. Our preliminary results show this simple adaptation equips CFG style control post-hoc to the model (Fig. 3).

However, since most CMs do not explicitly learn the $\emptyset$–class ODE, Naive JFDL's applicability is limited. Inspired by recent results that dispense with the need of a $\emptyset$-class for guidance [**?**], we propose Random JFDL, which replaces the $\emptyset$ label with a random class drawn from the data

Table 1: **Results on CIFAR-10 and ImageNet 64x64.** We train ECT first on a budget of 25.6M images, then tune it with Naive/Random JFDL on a budget of 1.92M images. The table shows the gains of FID posthoc compared to an unguided CM, with ω denoting the lowest recorded FID.

Method	**CIFAR-10**		**ImageNet 64×64**	
	1-step FID	2-step FID	1-step FID	2-step FID
ECT (baseline)	3.40	1.92	5.84	3.72
ECT + Naive JFDL	3.29 (ω=1.05)	2.18 (ω=1.55)	4.38 (ω=1.80)	3.17 (ω=1.40)
ECT + Random JFDL	3.24 (ω=1.15)	2.06 (ω=1.05)	4.68 (ω=1.80)	3.32 (ω=1.20)



(a) CIFAR-10 1-step (b) CIFAR-10 2-step (c) ImageNet 64x64 1-step (d) ImageNet 64x64 2-step

Figure 4: **FID to guidance strength progression.**

distribution. Surprisingly, this variant generates higher contrast images that aligns even closer with CFG. We provide the full Random JFDL in Appendix B.

4 Experiments

We first discuss our experimental setup. Then, we report quantitative FID results demonstrating that JFDL consistently outperforms the unguided ECT baseline. Finally, we visualize the guidance effect in the qualitative evaluation section.

Setup. Our preliminary results show that guidance yields FID improvements primarily when the guidance scale $\omega < 2$, a trend consistent with CFG in DMs. Therefore, our main experiments sample $\omega \in [1, 2]$ and $\omega \in [1, 4]$ on CIFAR-10 and ImageNet 64x64 respectively. We also observed that optimizing JFDL alongside the ECT objective leads to better FID scores. As shown in Alg. 1, we optimize a multi-task loss combining L_{ECT} and L_{JFDL} with adaptive weighting via GradNorm [4]. For both experiments, we use the last $\texttt{out\_conv}$ layer to compute the gradient norm of the two losses. Inspired by recent work on truncated CMs [29], we further shift the time sampling distribution to a higher log-normal mean, with -0.5 for CIFAR-10 and -0.4 for ImageNet 64×64. Following ECT, we adopt the EDM architecture for CIFAR-10, and zero-initialize the guidance embedding weights following ControlNet [62]. In ImageNet 64x64, we adopt the EDM2-S architecture and set the magnitude preserving coefficient to 1e-3 for the attached guidance layers to stablize fine-tuning. Both experiments use a batch size of 64 and converge after processing roughly 1.92 million images (30 k iterations). Compared to the lightweight ECT baseline, JFDL's fine-tuning consumes only about 7.5% of ECT's training data and around 30% of its GPU hours. We present detailed ablation studies of our design choices in Appendix. C.4.

Quantitative Results. We evaluate generation quality using the standard FID-50k metric w.r.t. different ω scales. For each guidance weight ω, we conducted three FID runs and report the mean in Tab. 1. Our post-hoc tuning framework enables a direct comparison between guided models and the unguided ECT baseline. Compared to the preliminary results in Fig. 3, jointly optimizing L_{ECT} and L_{JFDL} significantly reduces the FID compared to the baseline. Under one-step sampling, both Naive JFDL and Random JFDL consistently lower FID on CIFAR-10 and ImageNet 64×64. The resulting FID curve over ω closely mirrors CFG's behavior in DMs, exhibiting improvements for small scales before degrading. For two-step sampling, JFDL worsened FID on CIFAR-10, where

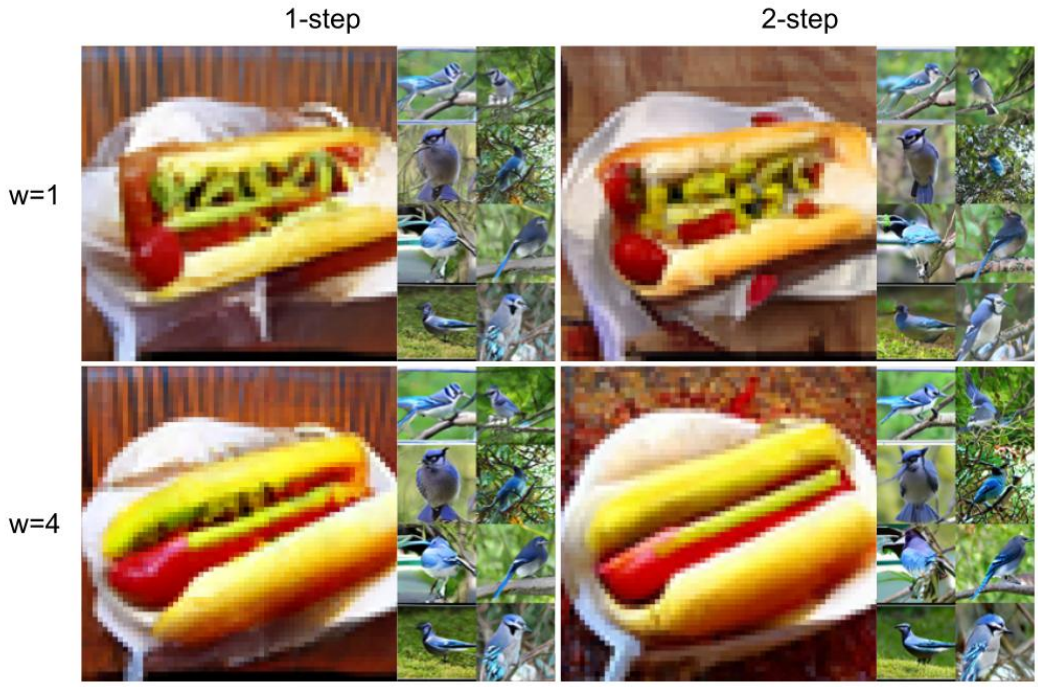


Figure 5: **ImageNet 64x64 sample results.** The classes shown are "jay" and "hotdog", generated by ECT + Naive JFDL. Rows are guidance strength. Columns are sampling steps.

the strong baseline meant any contrast gains were outweighed by diversity loss. However, JFDL improved FID on ImageNet 64×64, where there was more room for perceptual enhancement with few-step sampling under guidance. Notably, our experiments shows that the ω=1 case deviates from the ECT baseline, even though in the case of CFG it is expected to be exactly the same as unguided generation. This can be explained by the fact that fine-tuning the original model (e.g., via JFDL) alters its behavior, whereas CFG is a training-free method for guiding DMs.

Qualitative Results. Fig. 5 presents ImageNet 64×64 samples generated by JFDL under varying guidance scales and number of sampling steps. As the guidance scale ω increases (e.g. $\omega = 4$), perceptual quality improves as the mustard appears crisper and the bun is more well-defined. We can also observe the diminishing sample diversity, as seen in the loss of certain hotdog toppings. This same fidelity-diversity trade-off persists when using two-step sampling with high guidance on both steps. To conclude, JFDL reproduces the effects of CFG, higher contrast and enhanced perceptual fidelity, demonstrating its promise as a post-hoc guidance method for CMs.

5 Discussion and Future Work

We introduced JFDL, a fully post-hoc guidance framework for CMs that requires no DM teacher. By unifying perspectives from DMs, FMs and CMs, we derived theoretical guarantees for JFDL and validated them empirically. We further demonstrated that JFDL can equip an unguided ECT model with adjustable, CFG style guidance, yielding significant FID improvements on CIFAR-10 and ImageNet 64×64.

In our work, we intentionally obscured the theoretical groundings for applying guidance under the multi-step sampling settings, but still applied it following previous works as in [40]. In the context of CMs, one direction for future work is to explore the interaction between multi-step sampling and guidance, and develop adaptive guidance schemes tailored to multi-step solvers.

References

[1] Theodore W. Anderson and Donald A. Darling. Asymptotic theory of certain "goodness of fit" criteria based on stochastic processes. *Annals of Mathematical Statistics*, 23(2):193–212, 1952.

[2] Arwen Bradley and Preetum Nakkiran. Classifier-free guidance is a predictor-corrector. *arXiv preprint arXiv:2408.09000*, 2024.

[3] Junsong Chen, Yue Wu, Simian Luo, Enze Xie, Sayak Paul, Ping Luo, Hang Zhao, and Zhenguo Li. Pixart-δ: Fast and controllable image generation with latent consistency models, 2024.

[4] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks, 2018.

[5] Hyungjin Chung, Jeongsol Kim, Geon Yeong Park, Hyelin Nam, and Jong Chul Ye. Cfg++: Manifold-constrained classifier free guidance for diffusion models, 2024.

[6] Wenxun Dai, Ling-Hao Chen, Jingbo Wang, Jinpeng Liu, Bo Dai, and Yansong Tang. Motionlcm: Real-time controllable motion generation via latent consistency model, 2024.

[7] Quan Dao, Khanh Doan, Di Liu, Trung Le, and Dimitris Metaxas. Improved training technique for latent consistency models, 2025.

[8] Keya Rani Das and AHMR Imon. A brief review of tests for normality. *American Journal of Theoretical and Applied Statistics*, 5(1):5–12, 2016.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[10] Zhengcong Fei, Mingyuan Fan, and Junshi Huang. Music consistency models, 2024.

[11] Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models, 2024.

[12] Ruiqi Gao, Emiel Hoogeboom, Jonathan Heek, Valentin De Bortoli, Kevin P. Murphy, and Tim Salimans. Diffusion meets flow matching: Two sides of the same coin. 2024.

[13] Zhengyang Geng, Ashwini Pokle, William Luo, Justin Lin, and J. Zico Kolter. Consistency models made easy, 2024.

[14] Guande He, Kaiwen Zheng, Jianfei Chen, Fan Bao, and Jun Zhu. Consistency diffusion bridge models, 2024.

[15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

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

[17] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022.

[18] Chia-Hong Hsu, Shiu-hong Kao, and Randall Balestriero. Beyond and free from diffusion: Invertible guided consistency training. *arXiv preprint arXiv:2502.05391*, 2025.

[19] Rongjie Huang, Max WY Lam, Jun Wang, Dan Su, Dong Yu, Yi Ren, and Zhou Zhao. Fastdiff: A fast conditional diffusion model for high-quality speech synthesis. *arXiv preprint arXiv:2204.09934*, 2022.

[20] Rongjie Huang, Zhou Zhao, Huadai Liu, Jinglin Liu, Chenye Cui, and Yi Ren. Prodiff: Progressive fast diffusion model for high-quality text-to-speech. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 2595–2605, 2022.

[21] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models, 2022.

[22] Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a diffusion model with a bad version of itself. *Advances in Neural Information Processing Systems*, 37:52996–53021, 2024.

[23] Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models, 2024.

[24] Beomsu Kim, Yu-Guan Hsieh, Michal Klein, Marco Cuturi, Jong Chul Ye, Bahjat Kawar, and James Thornton. Simple reflow: Improved techniques for fast flow models, 2024.

[25] Beomsu Kim, Jaemin Kim, Jeongsol Kim, and Jong Chul Ye. Generalized consistency trajectory models for image manipulation. *arXiv preprint arXiv:2403.12510*, 2024.

[26] Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ode trajectory of diffusion. *arXiv preprint arXiv:2310.02279*, 2023.

[27] Ivan Kobyzev, Simon J.D. Prince, and Marcus A. Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):3964–3979, November 2021.

[28] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research).

[29] Sangyun Lee, Yilun Xu, Tomas Geffner, Giulia Fanti, Karsten Kreis, Arash Vahdat, and Weili Nie. Truncated consistency models, 2025.

[30] Liangchen Li and Jiajun He. Bidirectional consistency models, 2025.

[31] Zongrui Li, Minghui Hu, Qian Zheng, and Xudong Jiang. Connecting consistency distillation to score distillation for text-to-3d generation. In *European Conference on Computer Vision*, pages 274–291. Springer, 2024.

[32] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation, 2023.

[33] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023.

[34] Qiang Liu. Rectified flow: A marginal preserving approach to optimal transport, 2022.

[35] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow, 2022.

[36] Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models, 2025.

[37] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022.

[38] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*, 2022.

[39] Calvin Luo. Understanding diffusion models: A unified perspective, 2022.

[40] Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference, 2023.

[41] Frank J. Massey. The kolmogorov-smirnov test for goodness of fit. *Journal of the American Statistical Association*, 46(253):68–78, 1951.

[42] Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14297–14306, 2023.

[43] Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models, 2021.

[44] 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.

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

[46] 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*, pages 10684–10695, 2022.

[47] Seyedmorteza Sadat, Manuel Kansy, Otmar Hilliges, and Romann M Weber. No training, no problem: Rethinking classifier-free guidance for diffusion models. *arXiv preprint arXiv:2407.02687*, 2024.

[48] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

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

[50] S. S. SHAPIRO and M. B. WILK. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3-4):591–611, dec 1965.

[51] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2022.

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

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

[54] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution, 2020.

[55] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2021.

[56] Fu-Yun Wang, Zhengyang Geng, and Hongsheng Li. Stable consistency tuning: Understanding and improving consistency models, 2024.

[57] Felix Wimbauer, Bichen Wu, Edgar Schoenfeld, Xiaoliang Dai, Ji Hou, Zijian He, Artsiom Sanakoyeu, Peizhao Zhang, Sam Tsai, Jonas Kohler, Christian Rupprecht, Daniel Cremers, Peter Vajda, and Jialiang Wang. Cache me if you can: Accelerating diffusion models through block caching, 2024.

[58] Zhen Xing, Qijun Feng, Haoran Chen, Qi Dai, Han Hu, Hang Xu, Zuxuan Wu, and Yu-Gang Jiang. A survey on video diffusion models. *ACM Computing Surveys*, 57(2):1–42, 2024.

[59] Ling Yang, Zixiang Zhang, Zhilong Zhang, Xingchao Liu, Minkai Xu, Wentao Zhang, Chenlin Meng, Stefano Ermon, and Bin Cui. Consistency flow matching: Defining straight flows with velocity consistency, 2024.

[60] Bee Wah Yap and Chiaw Hock Sim. Comparisons of various types of normality tests. *Journal of Statistical Computation and Simulation*, 81(12):2141–2155, 2011.

[61] Ye Yuan, Jiaming Song, Umar Iqbal, Arash Vahdat, and Jan Kautz. Physdiff: Physics-guided human motion diffusion model. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16010–16021, 2023.

[62] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023.

[63] Jianbin Zheng, Minghui Hu, Zhongyi Fan, Chaoyue Wang, Changxing Ding, Dacheng Tao, and Tat-Jen Cham. Trajectory consistency distillation: Improved latent consistency distillation by semi-linear consistency function with trajectory mapping. *arXiv preprint arXiv:2402.19159*, 2024.

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: Section 4 and Section 3.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

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

   Answer: [Yes]

   Justification: Section 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

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: Section 3.1 and Appendix A.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

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: [Yes]

   Justification: Section 4 Appendix C.4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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: [Yes]

Justification: We declare that we will release the code at completion of the rebuttal for completeness.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Section 4, Appendix C and C.4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

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: [Yes]

   Justification: Section 3 reported the p/α for normality tests, and Section 4 we reported the statistical averaged results. We did not report error bars, but we included appropriate information about the statistical significance of our experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

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: In Section 4 we reported our GPU hours in comparison with previous works. We will disclose specific GPU type and concrete GPU hours in Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

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: [Yes]

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Appendix D.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

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., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our experiments trained on CIFAR-10 and ImageNet 64x64 would not pose such risk.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

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: [Yes]

    Justification: Section 1.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [NA]

    Justification: No new assets yet.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

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: [NA]

    Justification: Does not include crowdsourcing.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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: [NA]

    Justification: Does not involve any of these.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

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 rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: Not required.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.