Midpoint Generative Models

Daniil Shlenskii[*,1,2] **Nikita Gushchin**[*,2,1] **Lev Novitskiy**[3,2]

Dmitry V. Dylov[2,1] **Alexander Korotin**[2,1]

Abstract

We introduce *Midpoint Generative Models* (MGM), a principled framework for training one-step generative models. MGM is based on a simple symmetry of Flow Matching with linear interpolation: when the two endpoint distributions coincide, the corresponding velocity field vanishes at the midpoint time, $t = 1/2$. We show that the norm of this field defines a valid discrepancy between distributions, which we call the *Midpoint Divergence*. We extend this discrepancy beyond the midpoint by introducing randomly flipped interpolations and further generalize it by replacing deterministic linear Flow Matching interpolations with symmetric stochastic interpolants, yielding a generalized Midpoint Divergence. Finally, we derive a variational formulation of our generalized divergence, yielding a tractable objective for training a one-step generator. The resulting MGM algorithm offers an effective and theoretically grounded approach to generative modeling, achieving competitive performance against existing one-step generative modeling methods.

Code: `https://github.com/daniil-shlenskii/mgm`

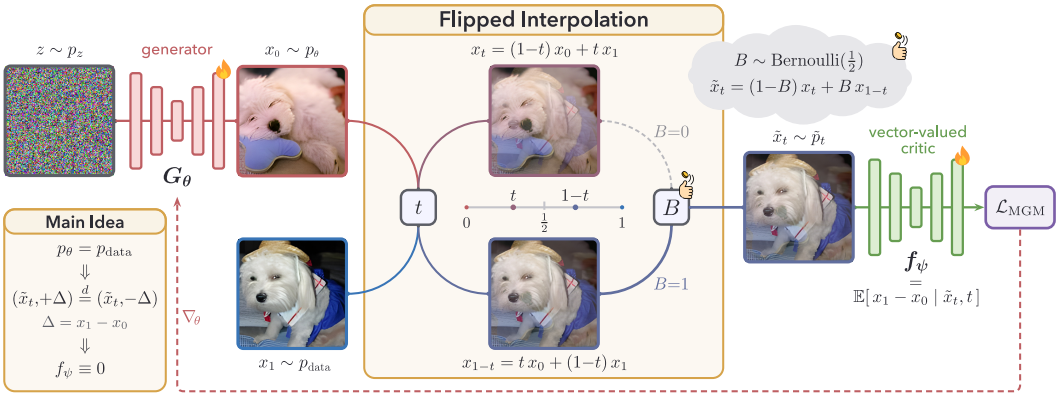


Figure 1: **Overview of our Midpoint Generative Models.** A latent $z \sim p_z$ is mapped by the generator G_θ to a sample $x_0 \sim p_\theta$, which is then paired with a data sample $x_1 \sim p_{\text{data}}$. The two endpoints are joined by a *symmetric stochastic interpolant* (6) at time $t \sim \mathcal{U}[0, 1/2]$, producing both the interpolation $x_t \sim p_t$ and its *time-flipped* counterpart $x_{1-t} \sim p_{1-t}$. A hidden flip B then selects between the two, $\tilde{x}_t = (1 - B)\,x_t + B\,x_{1-t}$, yielding the *symmetrized observation* $\tilde{x}_t \sim \tilde{p}_t$ (the flip B itself is never observed). This flip is the key ingredient of MGM: by restoring endpoint symmetry, it forces the symmetrized displacement field $\tilde{v}_t$ in (5) to vanish whenever $p_\theta = p_{\text{data}}$, making the generalized Midpoint Divergence (8) a definite discrepancy (Theorem 3.3). The vector-valued critic f_ψ approximates $\tilde{v}_t$ by regressing onto $x_1 - x_0$ given $\tilde{x}_t$ and t, while the generator G_θ minimizes the resulting divergence; the two are trained jointly in a minimax fashion through the variational midpoint objective (10). The complete procedure is detailed in Algorithm 1.

[1]AXXX, Russia; [2]Applied AI Institute, Russia; [3]Kandinsky Lab, Russia

*Equal contribution. Correspondence to: `daniil.shlenskii@gmail.com`

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

1 Introduction

Diffusion and flow models are now at the core of many generative systems, enabling a wide range of practical applications: image generation [1, 2], video generation [3], audio generation [4], image editing [5], and image-to-image translation [6]. However, whenever diffusion models are applied, the next question is often the same: how can we make them fast enough for practical use? Several lines of work address this question, including the development of more accurate numerical solvers [7–9], the distillation of pre-trained diffusion or flow models into one- or few-step generators [10–14], and the design of new training schemes that learn one-step generators from scratch using structural properties of diffusion and flow models [11, 15–17].

In this work, we propose a novel and effective framework for one-step generative modeling, built on a symmetry of Flow Matching with linear interpolation [18, 19]. The starting observation is simple: under linear interpolation between identical endpoint distributions $p_0 = p_1$, the velocity field vanishes at the midpoint time $t = 1/2$:

$$v_{1/2}(x) \equiv 0.$$

The converse holds as well: if $v_{1/2} \equiv 0$, then $p_0 = p_1$. This equivalence was recently established by Wang et al. [20] as the *zero-flow criterion* and used to learn sufficient representations through a squared-field loss. Building on this criterion, we define the *Midpoint Divergence* as the expected squared norm of the midpoint velocity: a nonnegative discrepancy that vanishes if and only if $p_0 = p_1$ and serves as a distribution-matching objective for training one-step generators.

The vanishing property, however, is specific to the midpoint $t = 1/2$: at any other time, the velocity field v_t does not vanish even when $p_0 = p_1$, and therefore cannot certify equality of the endpoints on its own. To overcome this, we introduce a random time-flip $t \mapsto 1 - t$ in the interpolation and show that the resulting symmetrized field vanishes at every t whenever $p_0 = p_1$, which extends the construction to a time-integrated Midpoint Divergence. Moreover, the same construction applies beyond deterministic linear paths and yields a generalized Midpoint Divergence for symmetric stochastic interpolants [21, 22]. This divergence is the basis of our training algorithm, Midpoint Generative Models (MGM), illustrated in Figure 1.

Our main contributions are as follows:

- We introduce the *Midpoint Divergence*, a new distributional discrepancy derived from the midpoint symmetry of linear interpolant Flow Matching (§3.1). We prove that this divergence is definite: it is nonnegative and vanishes if and only if the two endpoint distributions are equal. By introducing a random time flip, we further extend this construction to all interpolation times (§3.2) and to symmetric stochastic interpolants (§3.3), preserving definiteness in both cases.
- We develop *Midpoint Generative Models* (MGM), a practical training framework for one-step generative modeling based on the proposed divergence. We derive a variational formulation (§3.4) with a learned vector-valued critic, yielding an effective algorithm for training generators directly from data, without any teacher-model training signal. We show that our MGM is competitive with existing one-step generators on unconditional CIFAR-10, class-conditional ImageNet 64×64 in pixel space, and class-conditional ImageNet 512×512 in latent space (§5).

2 Background

In this section, we briefly review Flow Matching (§2.1) and stochastic interpolants (§2.2), which provide the velocity field and the interpolation paths underlying our construction.

2.1 Flow Matching

Flow Matching [18, 19] learns a time-dependent velocity field that defines an ordinary differential equation (ODE) transporting samples between distributions. Given independent endpoint samples $X_0 \sim p_0$ and $X_1 \sim p_1$, a basic choice is the linear interpolant

$$X_t = (1 - t)X_0 + tX_1, \qquad t \in [0, 1].$$

This interpolant defines a path of marginal distributions $p_t = \mathrm{Law}(X_t)$. These marginals are recovered by the probability flow ODE

$$\frac{dY_t}{dt} = v_t(Y_t), \qquad Y_0 \sim p_0,$$

where the velocity field is given by the conditional expectation

$$v_t(x) = \mathbb{E}_{X_0,X_1}\left[X_1 - X_0 \mid X_t = x\right]. \tag{1}$$

Thus, Flow Matching reduces generative modeling to learning the velocity field $v_t(x)$ that transports samples along the marginal path from p_0 to p_1 through the ODE.

2.2 Stochastic Interpolants

The stochastic interpolants framework is a generalization of the linear one, allowing more flexible, possibly noisy paths between endpoint samples [22]. Let $X_0 \sim p_0$, $X_1 \sim p_1$, and $\epsilon \sim \mathcal{N}(0, \mathrm{Id})$ be independent. A stochastic interpolant is defined as

$$X_t = I_t(X_0, X_1) + \sigma_t \epsilon, \qquad t \in [0,1],$$

where

$$I_0(X_0, X_1) = X_0, \qquad I_1(X_0, X_1) = X_1, \qquad \sigma_0 = \sigma_1 = 0,$$

with I_t twice continuously differentiable in both time and space and $\sigma_t^2 \in C^2([0,1])$.

The standard Flow Matching construction is recovered by setting $I_t(X_0, X_1) = (1-t)X_0 + tX_1$ and $\sigma_t = 0$, giving the linear interpolant above.

3 Midpoint Generative Models

In this section, we introduce **M**idpoint **G**enerative **M**odels (MGM), a principled framework for training one-step generative models. We begin in §3.1 by showing that the velocity field induced by linear Flow Matching contains a discriminative signal at the midpoint $t = 1/2$. In particular, this velocity field vanishes when the endpoint distributions coincide, and its failure to vanish gives rise to the Midpoint Divergence. Next, in §3.2, we extend this idea beyond the midpoint. Since non-midpoint times introduce an orientation bias, we introduce a randomly flipped interpolation that restores symmetry and leads to a time-integrated Midpoint Divergence. In §3.3, we further generalize the construction from deterministic linear interpolations to symmetric stochastic interpolants, yielding a generalized Midpoint Divergence. Finally, in §3.4, we show how this generalized divergence can be used as a training objective for generative modeling. We derive a variational formulation with a learned critic model and summarize the resulting practical training algorithm for one-step generators. All proofs are provided in Appendix A.

3.1 Midpoint Divergence

Consider the Flow Matching linear interpolation between two distributions p_0 and p_1 and the corresponding velocity field v_t in (1). A key property of this field is that it vanishes at the midpoint $t = 1/2$ whenever $p_0 = p_1$ (see Appendix E for an electrostatic interpretation). We illustrate this in Figure 2 and state it formally below.

Proposition 3.1 (Midpoint symmetry)**.** *Let X_0 and X_1 be independent samples from the same distribution p with finite first moment. Then*

$$v_{1/2}(x) = \mathbb{E}_{X_0,X_1}\left[X_1 - X_0 \mid X_{1/2} = x\right] = 0 \quad \textit{for } p_{1/2}\textit{-a.e. } x.$$

This suggests using the failure of the velocity field to vanish at the midpoint as a measure of discrepancy between p_0 and p_1.

Definition 3.1 (Midpoint Divergence)**.** *For distributions p_0 and p_1, define the Midpoint Divergence*

$$D_{\mathrm{mid}}(p_0, p_1) := \mathbb{E}_{X_{1/2}}\left\|\mathbb{E}_{X_0,X_1}\left[X_1 - X_0 \mid X_{1/2}\right]\right\|_2^2. \tag{2}$$

By Proposition 3.1, we immediately have $D_{\mathrm{mid}}(p_0, p_1) = 0$ whenever $p_0 = p_1$. To justify the term "divergence," we now show the converse: the Midpoint Divergence vanishes only when the endpoint distributions coincide.

Theorem 3.1 (Definiteness of the Midpoint Divergence)**.** *Let $X_0 \sim p_0$ and $X_1 \sim p_1$ be independent with sub-exponential tails. Then*

$$D_{\mathrm{mid}}(p_0, p_1) = 0 \quad \Longleftrightarrow \quad p_0 = p_1.$$

Thus, vanishing of the midpoint Flow Matching velocity field characterizes equality of the endpoint distributions. A natural next question is whether analogous discrepancies can be constructed using non-midpoint times.

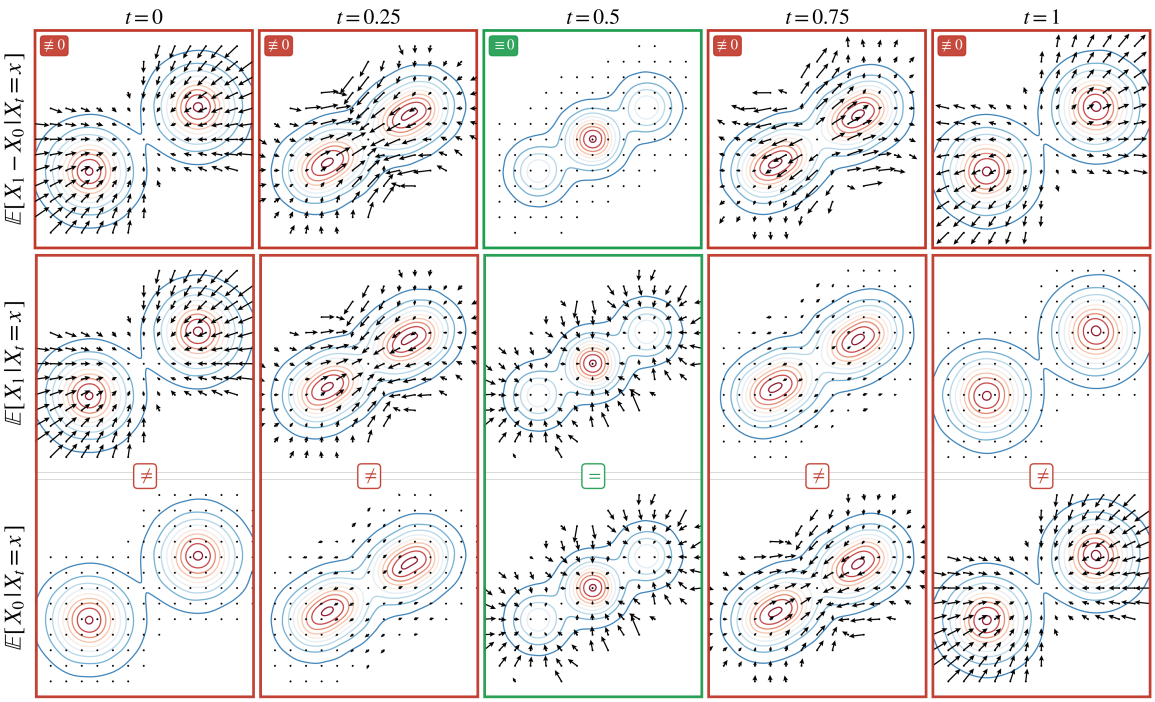


Figure 2: Linear interpolation for a Gaussian toy example. Contours show the interpolated density p_t. *Top row:* The velocity field $v_t(x) = \mathbb{E}[X_1 - X_0 \mid X_t = x]$ vanishes only at $t = 1/2$ and is nonzero away from the midpoint. *Bottom rows:* the endpoint denoisers are related by time reversal, $\mathbb{E}[X_0 \mid X_t = x] = \mathbb{E}[X_1 \mid X_{1-t} = x]$, rather than by equality at a fixed time.

3.2 Time-integrated Midpoint Divergence

A naive way to extend the midpoint construction over the full Flow Matching path is to integrate the squared norm of the Flow Matching velocity field over time:

$$D^{\text{naive}}_{\text{t-mid}}(p_0, p_1) := \int_0^1 \mathbb{E}_{X_t} \left\| \mathbb{E}_{X_0, X_1}[X_1 - X_0 \mid X_t] \right\|_2^2 dt. \tag{3}$$

However, this naive functional is not a valid discrepancy: for $t \neq 1/2$, the Flow Matching velocity field v_t generally does not vanish even when $p_0 = p_1$. Figure 2 (top row) illustrates this failure mode in a Gaussian toy example: under linear interpolation, the velocity field vanishes at the midpoint but remains nonzero at the other displayed times. This can be seen more clearly by rewriting the velocity as a difference of two posterior endpoint means, which we refer to as *denoisers*:

$$v_t(x) = \mathbb{E}_{X_0, X_1}\left[X_1 - X_0 \mid X_t = x\right] = \mathbb{E}_{X_1}\left[X_1 \mid X_t = x\right] - \mathbb{E}_{X_0}\left[X_0 \mid X_t = x\right].$$

Even when $p_0 = p_1$, these two denoisers are generally different away from the midpoint. For example, at $t = 0$ we have $X_t = X_0$, and hence

$$\mathbb{E}_{X_1}\left[X_1 \mid X_0 = x\right] = \mathbb{E}_{X_1} X_1, \qquad \mathbb{E}_{X_0}\left[X_0 \mid X_0 = x\right] = x.$$

Thus, given $X_0 = x$, the squared-error optimal estimator of X_0 is x itself, whereas the optimal reconstruction of the independent endpoint X_1 is only its mean. The same phenomenon occurs at other non-midpoint times: the closer t is to one endpoint, the easier it is to reconstruct that endpoint from X_t, and the harder it is to reconstruct the other endpoint. Hence, the two denoisers are intrinsically asymmetric away from $t = 1/2$. This phenomenon is visually illustrated in Figure 2 (bottom rows). To remove this orientation bias, we introduce a *randomly flipped interpolation*

$$\widetilde{X}_t = \begin{cases} X_t = (1-t)X_0 + tX_1, & B = 0, \\ X_{1-t} = tX_0 + (1-t)X_1, & B = 1, \end{cases} \tag{4}$$

where $B \sim \text{Bernoulli}(1/2)$ is independent of (X_0, X_1). Importantly, the flip variable B is not observed. This randomness makes the two endpoints equally close, in distribution, to the observed interpolation point. Consequently, when $p_0 = p_1$, neither endpoint has an intrinsic reconstruction advantage, and the expected displacement given $\widetilde{X}_t$ vanishes. This is illustrated in Figure 3 and formalized in the following proposition.

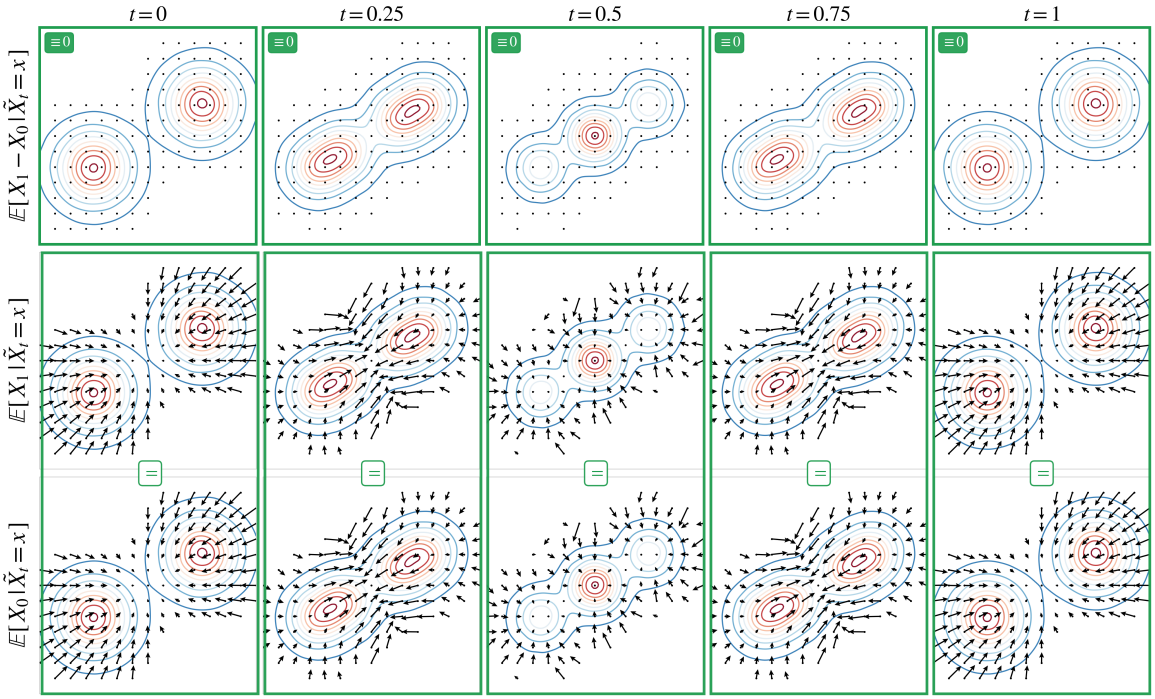


Figure 3: Randomly flipped interpolation on the same Gaussian toy example. Contours show the symmetrized density $\tilde{p}_t$. *Top row:* The symmetrized displacement field $\tilde{v}_t(x) = \mathbb{E}[X_1 - X_0 \mid \tilde{X}_t = x]$ is identically zero for every displayed t. *Bottom rows:* Under the same symmetrized observation model, the endpoint denoisers coincide pointwise for every t, $\mathbb{E}[X_0 \mid \tilde{X}_t = x] = \mathbb{E}[X_1 \mid \tilde{X}_t = x]$.

Proposition 3.2 (Flip-induced symmetry)**.** *Let X_0 and X_1 be independent samples from the same distribution p with finite first moment. Then, for every $t \in [0,1]$, the symmetrized displacement field*

$$\widetilde{v}_t(x) := \mathbb{E}_{X_0,X_1}[X_1 - X_0 \mid \widetilde{X}_t = x] = 0 \quad \textit{for } \widetilde{p}_t\textit{-a.e. } x, \tag{5}$$

where $\widetilde{X}_t \sim \widetilde{p}_t := (p_t + p_{1-t})/2$.

At $t = 1/2$, both branches coincide, so $\widetilde{v}_{1/2} = v_{1/2}$ and the flipped construction reduces to the midpoint one. Since, for $p_0 = p_1$, the symmetrized displacement field vanishes at every t, it is natural to define a discrepancy by integrating its squared norm over time.

Definition 3.2 (Time-integrated Midpoint Divergence)**.** *For any two distributions p_0 and p_1 we define time-integrated Midpoint Divergence as the following functional*

$$D_{\text{t-mid}}(p_0, p_1) := \int_0^{1/2} \mathbb{E}_{\widetilde{X}_t} \left\| \mathbb{E}_{X_0,X_1}[X_1 - X_0 \mid \widetilde{X}_t] \right\|_2^2 dt.$$

We integrate only over $[0, 1/2]$ because, under the random flip construction, the discrepancy at time t is the same as at time $1 - t$; integrating over $[0, 1]$ would therefore count each contribution twice. As in the single-time midpoint case, this functional is a valid divergence.

Theorem 3.2 (Definiteness of the time-integrated Midpoint Divergence)**.** *Let $X_0 \sim p_0$ and $X_1 \sim p_1$ be independent with finite second moments. Then*

$$D_{\text{t-mid}}(p_0, p_1) = 0 \quad \Longleftrightarrow \quad p_0 = p_1.$$

3.3 Generalized Midpoint Divergence

In the flipped construction, the symmetrized displacement field $\widetilde{v}_t(x)$ in (5) is no longer the Flow Matching velocity field in (1). Nevertheless, it can still be interpreted as the difference between two posterior endpoint denoisers under the symmetrized observation model:

$$\widetilde{v}_t(x) = \mathbb{E}_{X_0,X_1}[X_1 - X_0 \mid \widetilde{X}_t = x] = \mathbb{E}_{X_1}[X_1 \mid \widetilde{X}_t = x] - \mathbb{E}_{X_0}[X_0 \mid \widetilde{X}_t = x].$$

This perspective decouples the construction from the specific linear interpolation used in Flow Matching and suggests a broader class of symmetric interpolation paths.

We now extend the construction to stochastic interpolants defined in §2.2:

$$X_t = I_t(X_0, X_1) + \sigma_t \epsilon, \qquad \epsilon \sim \mathcal{N}(0, \mathrm{Id}). \tag{6}$$

We additionally require the interpolant to be symmetric, namely

$$I_t(x_0, x_1) = I_{1-t}(x_1, x_0), \qquad \sigma_t = \sigma_{1-t}.$$

Let $B \sim \mathrm{Bernoulli}(1/2)$ be independent of (X_0, X_1, ϵ). We define the flipped observation by

$$\widetilde{X}_t = \begin{cases} X_t = I_t(X_0, X_1) + \sigma_t \epsilon, & B = 0, \\ X_{1-t} = I_{1-t}(X_0, X_1) + \sigma_{1-t} \epsilon, & B = 1. \end{cases} \tag{7}$$

This symmetrized observation model leads to the following generalized Midpoint Divergence:

$$D^{I,\sigma}_{\text{t-mid}}(p_0, p_1) := \int_0^{1/2} \mathbb{E}_{\widetilde{X}_t} \left\| \mathbb{E}_{X_0, X_1}[X_1 - X_0 \mid \widetilde{X}_t] \right\|_2^2 dt. \tag{8}$$

This reduces to the time-integrated midpoint divergence for the Flow Matching linear interpolant. The next theorem shows that this generalized construction still defines a valid divergence.

Theorem 3.3 (Definiteness of the generalized Midpoint Divergence)**.** *Let $X_0 \sim p_0$ and $X_1 \sim p_1$ be independent with finite second moments. Fix any symmetric stochastic interpolant* (6)*. Then*

$$D^{I,\sigma}_{\text{t-mid}}(p_0, p_1) = 0 \quad \Longleftrightarrow \quad p_0 = p_1.$$

Theorem 3.3 shows that the generalized Midpoint Divergence can serve as a principled training objective: minimizing it against the data distribution identifies the target distribution uniquely. We now use this divergence to construct our Midpoint Generative Models.

3.4 Midpoint Generative Models

The generalized Midpoint Divergence provides a natural approach for generative modeling. Let p_θ be the distribution induced by a generator G_θ applied to a latent prior p_z:

$$Z \sim p_z, \qquad X_0 = G_\theta(Z), \qquad X_0 \sim p_\theta.$$

We pair generated samples with data samples $X_1 \sim p_{\text{data}}$, and define $\widetilde{X}_t$ using the flipped stochastic interpolant in (7). The generalized midpoint divergence then provides a natural training objective

$$\min_\theta D^{I,\sigma}_{\text{t-mid}}(p_\theta, p_{\text{data}}) = \int_0^{1/2} \mathbb{E}_{\widetilde{X}_t} \left\| \mathbb{E}_{X_0, X_1}[X_1 - X_0 \mid \widetilde{X}_t] \right\|_2^2 dt. \tag{9}$$

By Theorem 3.3, the divergence vanishes if and only if $p_\theta = p_{\text{data}}$. Directly optimizing (9), however, is intractable because the inner expectation is unknown. We therefore use the following variational representation.

Proposition 3.3 (Variational midpoint objective)**.** *For fixed p_θ, the generalized midpoint divergence admits the variational form*

$$D^{I,\sigma}_{\text{t-mid}}(p_\theta, p_{\text{data}}) = \max_{f_t} \int_0^{1/2} \mathbb{E}_{X_0, X_1, B, \epsilon} \left[2\langle f_t(\widetilde{X}_t), X_1 - X_0 \rangle - \|f_t(\widetilde{X}_t)\|_2^2 \right] dt,$$

where $\widetilde{X}_t$ is defined according to (7) *and the maximum is over square-integrable vector-valued functions f_t. Moreover, the maximizer is given by*

$$f_t^\star(x) = \mathbb{E}_{X_0, X_1} \left[X_1 - X_0 \mid \widetilde{X}_t = x \right] = \widetilde{v}_t(x),$$

for almost every t and $\widetilde{p}_t$-almost every x.

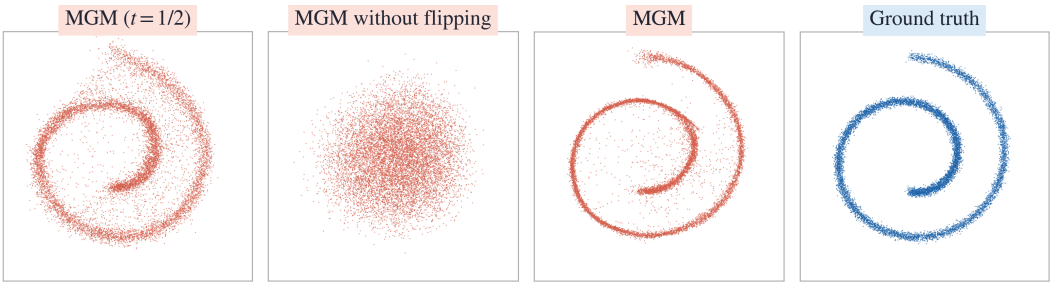


Figure 4: Toy example illustrating the role of the MGM construction. Full MGM recovers the target distribution, while the midpoint-only objective at $t = 1/2$ remains valid but produces noisier samples. In contrast, naively integrating the unflipped midpoint objective over time fails.

Using Proposition 3.3 and parameterizing the symmetrized displacement field by a neural network f_ψ, which we call the vector-valued *critic*, we obtain the following minimax objective:

$$\min_\theta \max_\psi \mathcal{L}_{\mathrm{MGM}}(\theta, \psi) := \int_0^{1/2} \mathbb{E}_{X_0, X_1, B, \epsilon} \left[2\langle f_\psi(\widetilde{X}_t, t), X_1 - X_0 \rangle - \|f_\psi(\widetilde{X}_t, t)\|_2^2 \right] dt. \quad (10)$$

The integrand of (10) can be equivalently rewritten as

$$2\langle f_\psi(\widetilde{X}_t, t), X_1 - X_0 \rangle - \|f_\psi(\widetilde{X}_t, t)\|_2^2 = \|X_1 - X_0\|_2^2 - \|f_\psi(\widetilde{X}_t, t) - (X_1 - X_0)\|_2^2.$$

Since the first term does not depend on ψ, the critic update is exactly Flow Matching regression of $X_1 - X_0$, applied to the symmetrized observation $\widetilde{X}_t$ instead of X_t. The generator, in contrast, must use the original form (10), since $\|X_1 - X_0\|_2^2$ depends on θ through $X_0 = G_\theta(Z)$. In practice, we alternate between updating the critic model to approximate the symmetrized displacement field and updating the generator to minimize the resulting variational midpoint objective. The full stochastic training procedure is summarized in Algorithm 1.

Importance of flipping and time integration. The variational formulation above can be instantiated not only with the generalized Midpoint Divergence, but also with simpler alternatives such as the midpoint-only divergence D_{mid} in (2) and the naive unflipped time-integrated objective $D^{\mathrm{naive}}_{\text{t-mid}}(p_0, p_1)$ in (3); details are provided in Appendix B. These alternatives allow us to ablate the two key ingredients of MGM: random flipping and time integration.

Figure 4 shows the resulting samples on the Swiss roll toy dataset. The midpoint-only objective learns the coarse geometry of the target, but produces noisy samples. One possible explanation is that, at $t = 1/2$, the critic f_ψ only observes the mixed interpolation $x_{1/2} = (x_0 + x_1)/2$, in which the generator output x_0 appears only through an averaged state. This may provide a weaker signal for correcting fine-grained errors in generator samples. In contrast, the flipped time-integrated objective includes times closer to the endpoints. For small t, the symmetrized observation contains samples close to either X_0 or X_1, while preserving endpoint symmetry through the random flip. Thus, time integration gives access to a richer family of observations, including near-generator samples, which can provide a stronger signal for fine-grained details.

The naive unflipped time-integrated objective fails, as expected from the orientation bias of non-midpoint Flow Matching velocities. The flipped time-integrated objective, however, accurately recovers the target distribution. This comparison supports the full MGM construction: random flipping makes the symmetrized displacement field vanish when $p_0 = p_1$, while time integration provides a stronger training signal than the midpoint alone. For the empirical benefit of the stochastic interpolants introduced in §3.3, see §5.2.

4 Related Work

Now we discuss three main axes of work related to MGM: distillation of pre-trained diffusion and flow models, direct training of fast generators from diffusion/flow principles, and GANs.

Distillation from pre-trained diffusion and flow models. A common route to fast generation is to distill a pre-trained diffusion or flow model into a one- or few-step student. One branch exploits PF-ODE consistency: teacher samples follow deterministic trajectories, so a pair (x_t, t) identifies the remaining trajectory including the final point. These methods train a student to reproduce two or

Warmup	FID
0K	34.02
10K	3.73
20K	2.69
50K	**2.34**
100K	2.42
200K	2.66

(a) **Warmup duration**

σ	FID
0.00	5.87
0.01	2.58
0.05	**2.46**
0.20	2.78
0.50	3.55

(b) **Interpolant stochasticity**

Critic σ	FID
0.00	5.87
0.01	**2.27**
0.05	2.34
0.20	2.98
0.50	56.58

(c) **Critic-only noise**

Critic σ	FID
0.00	14.75
0.01	**13.55**
0.05	19.76
0.20	68.62
0.50	120.48

(d) **Midpoint-only training**

Table 1: **Ablation study on unconditional CIFAR-10** 32×32 **generation.** FID-50K is reported for one-step generation, and the best setting within each ablation is in **bold**.

more teacher-sampler steps or to make predictions at different points on the same teacher trajectory agree [10, 11, 15, 23–27]. A second branch parameterizes $p_\theta(x_0)$ through a stochastic generator and optimizes losses derived from a pre-trained teacher [12, 13, 28–35]. These losses usually lead to adversarial optimization and often require an additional estimator for the generator distribution, such as a generator score, generator flow, or critic. Many of these works can be viewed through the inverse-distillation lens of Universal Inverse Distillation [36]. *In contrast, MGM does not distill a pre-trained diffusion or flow teacher; it trains directly from data using the proposed Midpoint Divergence between generated samples and real data, without any teacher signal in training loss.*

Training one- and few-step generators from scratch using diffusion/flow properties. Another line of work trains fast samplers directly from data, without a pretrained teacher as the training signal. These methods use the fact that finite-time updates along the same generative ODE trajectory must be mutually consistent: an update from s to t should agree with updates that pass through intermediate times. Their common goal is to obtain such finite-time integrators, parameterized as endpoint-consistent predictors, two-time trajectory maps, finite-interval updates, or average velocities [11, 15, 23, 24, 16, 17]. MeanFlow [17], for example, learns an average velocity constrained to be consistent with the instantaneous velocity of the generative ODE, so the guidance toward the data is internal to the model itself. Some methods, such as Consistency Models and Consistency Trajectory Models, also support teacher distillation, but their direct-training variants use only data and noise samples [11, 24]. Flow Map Matching gives a general view of many such objectives as learning maps between pairs of times in diffusion or stochastic-interpolant dynamics [37]. MGM is close in spirit, as it also trains without a teacher using diffusion and flow principles. *Unlike these approaches, MGM does not learn a finite-time integrator: its generator directly parameterizes the generated distribution, while the guidance toward the data comes from an auxiliary critic discarded at inference.*

GANs. Classical GANs [38] train a generator against a scalar discriminator, with f-GAN [39] and Wasserstein GAN [40] connecting discriminator-generator minimax objectives to standard divergences. Diffusion-GAN variants [41, 42] combine adversarial objectives with diffusion ideas, for example by discriminating noised samples or modeling large denoising steps adversarially. MGM also has a minimax training form, but uses a vector-valued critic $f_\psi(x, t)$ trained toward endpoint displacements under a symmetrized stochastic interpolant. *Thus, MGM optimizes a regression-based generalized Midpoint Divergence, not a classification-based adversarial loss.*

5 Experiments

We evaluate MGM for one-step unconditional generation on CIFAR-10 [43] and class-conditional generation on ImageNet [44]. We use pixel-space models for CIFAR-10 and ImageNet 64×64, and latent-space models for ImageNet 512×512, following EDM2 [45]. We report FID [46] on 50K generated samples. In §5.1, we describe the stochastic interpolant and initialization used in our experiments. In §5.2, we ablate the main design choices of MGM on CIFAR-10. Finally, in §5.3, we compare MGM against recent one-step generative models. Additional experimental details, further results, and computational cost analysis are provided in Appendices C, D, and F, respectively.

5.1 Experimental Setup

In all experiments, we train MGM with the generalized Midpoint Divergence (§3.3), using the linear interpolant $I_t(x_0, x_1) = (1 - t)x_0 + tx_1$ with the noise schedule $\sigma_t = \sqrt{\sigma t(1 - t)}$. On ImageNet, both the generator and the critic are initialized from pretrained diffusion models: EDM [8] for 64×64

w/ teacher-model training signal		w/o teacher-model training signal	
Method	FID	Method	FID
Unconditional CIFAR-10 32×32			
PD [10]	8.34	ECT [26]	3.60
2-Rectified Flow [19]	4.85	VCT [47]	3.26
TRACT [25]	3.78	IMM [48]	3.20
DFNO [49]	3.78	StyleGAN2 w/ ADA [50]	2.92
DMD [12]	3.77	Stable CT [51]	2.92
sCD [23]	3.66	MeanFlow [17]	2.92
CD [11]	3.55	sCT [23]	2.85
CMT w/ ECT [52]	2.74	iCT [15]	2.83
TCM [27]	2.46	iCT-deep [15]	2.51
CTM [24]	<u>1.98</u>	DiffRatio-DiJS [53]	<u>2.39</u>
SiD [31]	**1.92**	MGM (Ours)	**2.27**
Class-conditional ImageNet 64×64 *(pixel space)*			
sCD [23]	2.44	VCT [47]	4.93
ECD [26]	2.24	iCT [15]	4.02
TCM [27]	2.20	iCT-deep [15]	3.25
CMT w/ ECT [52]	2.02	ECT [26]	2.49
CTM [24]	1.92	Stable CT [51]	2.42
CMT w/ ECD [52]	1.78	sCT [23]	<u>2.04</u>
SiD [31]	<u>1.52</u>	DiffRatio-DiJS [53]	**1.54**
DMD2 [29]	**1.28**	MGM (Ours)	2.50
Class-conditional ImageNet 512×512 *(latent space)*			
ECD [26]	8.47	sCT-S [23]	10.13
CMT-XXL w/ ECD [52]	3.38	ECT [26]	9.98
AYF-S [54]	3.32	sCT-XXL [23]	4.29
sCD-S [23]	<u>3.07</u>	DiffRatio-DiJS-M [53]	**1.41**
sCD-XXL [23]	**2.28**	MGM-S (Ours)	<u>2.36</u>

Table 2: One-step generation on unconditional CIFAR-10 and class-conditional ImageNet, grouped by whether a pretrained model enters the training loss (§5.3). Within each group, the best result is in **bold**, the second best is <u>underlined</u>, and MGM is highlighted. On ImageNet 512×512, suffixes denote EDM2 sizes: S (280M), M (498M), and XXL (1.5B). Baseline values are from the cited papers or as collected by CMT [52].

and EDM2-S [45] for 512×512. On CIFAR-10, we instead initialize from a short denoising warmup to show that MGM does not require a fully trained diffusion model. In all cases, the pretrained network serves only as an initialization and does not enter the training loss. The warmup is ablated in §5.2 and detailed in Appendix C.

5.2 Ablation Study

We ablate four design choices in MGM: how long to run the denoising-objective warmup, how much noise to use in the stochastic interpolant, whether the generator should observe noisy interpolants, and whether the objective should integrate over time or use only the midpoint. The corresponding FID results are reported in Table 1.

Warmup duration. We vary the denoising-objective warmup budget before MGM training (Table 1.a). Following Karras et al. [8], we measure warmup budgets in kimgs (thousands of images processed during training). FID improves from 34.02 without warmup to 2.34 after 50K kimgs of warmup (25% of the original EDM training budget), with no further gains from longer warmups. So we use the 50K kimgs checkpoint to initialize all subsequent CIFAR-10 experiments.

Interpolant stochasticity. We vary the noise level σ in the stochastic schedule $\sigma_t = \sqrt{\sigma t(1-t)}$ (Table 1.b). Adding noise improves FID substantially over the deterministic setting $\sigma = 0$, with



Figure 5: One-step ImageNet 512×512 samples from MGM (FID 2.36); more in Appendix G.

small nonzero noise giving the best results. This is consistent with observations in image-to-image translation that adding bridge noise can improve performance [55–59]. Increasing the noise further worsens FID.

Critic stochasticity. We next decouple the generator and critic interpolants (Table 1.c) by adding noise only in the critic loss, while retaining the random time flip in both losses. For small noise levels, this critic-only stochasticity improves FID, giving the best result in our ablations, and for moderate levels it performs comparably to the fully stochastic setting. This suggests that noise is primarily important as a regularizer of the critic. At the large noise level $\sigma = 0.5$, however, training degrades significantly, which may reflect the increased mismatch between the noisy points on which the critic is trained and the noiseless points at which it is evaluated in the generator loss.

Midpoint-only training. We compare the time-integrated objective with a midpoint-only variant that fixes $t = 1/2$ (Table 1.d). In the same critic-only noise setting, the midpoint-only variant is substantially worse across all noise levels, reaching at best FID 13.55 compared to 2.27, and it breaks down already at moderate noise $\sigma = 0.2$. This confirms that extending the divergence to all interpolation times via the random time flip (§3.2) provides a much stronger learning signal than the midpoint alone.

5.3 Baseline Comparison

We compare MGM with recent one-step generative models in Table 2, grouping them by whether a teacher-model training signal is used, i.e., whether a separately pretrained model, or targets and trajectories produced by it, enters the generator loss; using a pretrained model only to initialize the networks does not count as such a signal. On CIFAR-10, MGM achieves FID 2.27, the best result among methods without a teacher-model training signal and comparable to distillation-based approaches. On ImageNet 64×64, MGM reaches FID 2.50, comparable to recent one-step methods. On ImageNet 512×512, MGM with a 280M-parameter generator achieves FID 2.36, outperforming or approaching several 1.5B-parameter models, including teacher-based ones. One-step ImageNet 512×512 samples are shown in Figure 5. Overall, MGM is comparable with recent one-step generative models across all benchmarks without relying on a teacher model during training.

6 Discussion and Limitations

We introduced *Midpoint Generative Models* (MGM), a principled framework for training one-step generators by minimizing the generalized Midpoint Divergence. As in GANs [38], MGM updates the generator using a jointly trained critic. Unlike a GAN discriminator, however, the MGM critic is vector-valued, like the auxiliary networks of distribution-matching distillation methods [12, 28, 31]. In contrast to these methods, the critic alone guides the generator, without a separate pretrained score network for the data distribution. These properties, together with MGM's competitive results, make it a promising direction for one-step generative modeling.

At the same time, MGM has several limitations. First, like other minimax methods, MGM can be unstable. Second, training from scratch is not yet competitive, and MGM relies on a reasonable initialization. Third, a principled adaptation of classifier-free guidance to one-step MGM, which is important for text-to-image generation, remains an open question.

Acknowledgments

The work was supported by the grant for research centers in the field of AI provided by the Ministry of Economic Development of the Russian Federation in accordance with the agreement 000000C313925P4F0002 and the agreement No. 139-10-2025-033.

References

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

[2] 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.

[3] 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.

[4] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=a-xFK8Ymz5J.

[5] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=aBsCjcPu_tE.

[6] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 conference proceedings*, pages 1–10, 2022.

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

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

[9] 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.

[10] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=TIdIXIpzhoI.

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

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

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

[14] Jiatao Gu, Shuangfei Zhai, Yizhe Zhang, Lingjie Liu, and Joshua M. Susskind. BOOT: Data-free distillation of denoising diffusion models with bootstrapping. In *ICML 2023 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2023. URL https://openreview.net/forum?id=ZeM7S01Xi8.

[15] Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=WNzy9bRDvG.

[16] Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=OlzB6LnXcS.

[17] 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.

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

[19] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

[20] Yakun Wang, Leyang Wang, Song Liu, and Taiji Suzuki. Zero-flow encoders. *arXiv preprint arXiv:2602.00797*, 2026.

[21] Michael Samuel Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=li7qeBbCR1t.

[22] Michael Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209): 1–80, 2025.

[23] Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=LyJi5ugyJx.

[24] 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. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=ymjI8feDTD.

[25] David Berthelot, Arnaud Autef, Jierui Lin, Dian Ang Yap, Shuangfei Zhai, Siyuan Hu, Daniel Zheng, Walter Talbott, and Eric Gu. Tract: Denoising diffusion models with transitive closure time-distillation. *arXiv preprint arXiv:2303.04248*, 2023.

[26] 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. URL https://openreview.net/forum?id=xQVxo9dSID.

[27] Sangyun Lee, Yilun Xu, Tomas Geffner, Giulia Fanti, Karsten Kreis, Arash Vahdat, and Weili Nie. Truncated consistency models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=ZYDEJEvCbv.

[28] Weijian Luo, Tianyang Hu, Shifeng Zhang, Jiacheng Sun, Zhenguo Li, and Zhihua Zhang. Diff-instruct: A universal approach for transferring knowledge from pre-trained diffusion models. *Advances in Neural Information Processing Systems*, 36:76525–76546, 2023.

[29] Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and Bill Freeman. Improved distribution matching distillation for fast image synthesis. *Advances in neural information processing systems*, 37:47455–47487, 2024.

[30] Yilun Xu, Weili Nie, and Arash Vahdat. One-step diffusion models with f-divergence distribution matching. *arXiv preprint arXiv:2502.15681*, 2025.

[31] Mingyuan Zhou, Huangjie Zheng, Zhendong Wang, Mingzhang Yin, and Hai Huang. Score identity distillation: Exponentially fast distillation of pretrained diffusion models for one-step generation. In *Forty-first International Conference on Machine Learning*, 2024.

[32] Weijian Luo, Zemin Huang, Zhengyang Geng, J Zico Kolter, and Guo-jun Qi. One-step diffusion distillation through score implicit matching. *Advances in Neural Information Processing Systems*, 37:115377–115408, 2024.

[33] Nikita Gushchin, David Li, Daniil Selikhanovych, Evgeny Burnaev, Dmitry Baranchuk, and Alexander Korotin. Inverse bridge matching distillation. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=UCJSF6Vt0C.

[34] Zemin Huang, Zhengyang Geng, Weijian Luo, and Guo-jun Qi. Flow generator matching. *arXiv preprint arXiv:2410.19310*, 2024.

[35] Daniil Shlenskii and Alexander Korotin. Overclocking electrostatic generative models. 2026. URL https://openreview.net/forum?id=flo449mncA.

[36] Nikita Maksimovich Kornilov, David Li, Tikhon Mavrin, Aleksei Leonov, Nikita Gushchin, Evgeny Burnaev, Iaroslav Sergeevich Koshelev, and Alexander Korotin. Universal inverse distillation for matching models with real-data supervision (no GANs). In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=8NuN5UzXLC.

[37] Nicholas Matthew Boffi, Michael Samuel Albergo, and Eric Vanden-Eijnden. Flow map matching with stochastic interpolants: A mathematical framework for consistency models. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=cqDH0e6ak2.

[38] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[39] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. *Advances in neural information processing systems*, 29, 2016.

[40] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. Pmlr, 2017.

[41] Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion GANs. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=JprM0p-q0Co.

[42] Zhendong Wang, Huangjie Zheng, Pengcheng He, Weizhu Chen, and Mingyuan Zhou. Diffusion-GAN: Training GANs with diffusion. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=HZf7UbpWHuA.

[43] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

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

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

[46] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[47] Gianluigi Silvestri, Luca Ambrogioni, Chieh-Hsin Lai, Yuhta Takida, and Yuki Mitsufuji. VCT: Training consistency models with variational noise coupling. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=CMoX0BEsDs.

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

[49] Hongkai Zheng, Weili Nie, Arash Vahdat, Kamyar Azizzadenesheli, and Anima Anandkumar. Fast sampling of diffusion models via operator learning. In *International conference on machine learning*, pages 42390–42402. PMLR, 2023.

[50] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *Advances in neural information processing systems*, 33:12104–12114, 2020.

[51] Fu-Yun Wang, Zhengyang Geng, and Hongsheng Li. Stable consistency tuning: Understanding and improving consistency models. In *ICLR 2025 Workshop on Deep Generative Model in Machine Learning: Theory, Principle and Efficacy*, 2025. URL https://openreview.net/forum?id=5RoPe2ShXx.

[52] Zheyuan Hu, Chieh-Hsin Lai, Yuki Mitsufuji, and Stefano Ermon. Cmt: Mid-training for efficient learning of consistency, mean flow, and flow map models. *arXiv preprint arXiv:2509.24526*, 2025.

[53] Wenlin Chen, Mingtian Zhang, Jiajun He, Zijing Ou, José Miguel Hernández-Lobato, Bernhard Schölkopf, and David Barber. Diffratio: Training one-step diffusion models without teacher supervision. *arXiv preprint arXiv:2502.08005*, 2025.

[54] Amirmojtaba Sabour, Sanja Fidler, and Karsten Kreis. Align your flow: Scaling continuous-time flow map distillation. In *Advances in Neural Information Processing Systems*, 2025.

[55] Xingchao Liu, Lemeng Wu, Mao Ye, et al. Let us build bridges: Understanding and extending diffusion generative models. In *NeurIPS 2022 Workshop on Score-Based Methods*, 2022.

[56] Linqi Zhou, Aaron Lou, Samar Khanna, and Stefano Ermon. Denoising diffusion bridge models. *arXiv preprint arXiv:2309.16948*, 2023.

[57] Kaiwen Zheng, Guande He, Jianfei Chen, Fan Bao, and Jun Zhu. Diffusion bridge implicit models. *arXiv preprint arXiv:2405.15885*, 2024.

[58] Guande He, Kaiwen Zheng, Jianfei Chen, Fan Bao, and Jun Zhu. Consistency diffusion bridge models. *Advances in Neural Information Processing Systems*, 37:23516–23548, 2024.

[59] Yuang Wang, Siyeop Yoon, Pengfei Jin, Matthew Tivnan, Sifan Song, Zhennong Chen, Rui Hu, Li Zhang, Quanzheng Li, Zhiqiang Chen, et al. Implicit image-to-image schrödinger bridge for image restoration. *Pattern Recognition*, 165:111627, 2025.

[60] Yilun Xu, Ziming Liu, Max Tegmark, and Tommi Jaakkola. Poisson flow generative models. In *Advances in Neural Information Processing Systems*, 2022.

[61] Alexander Kolesov, Stepan Manukhov, Vladimir V. Palyulin, and Alexander Korotin. Field matching: an electrostatic paradigm to generate and transfer data. In *International Conference on Machine Learning*, 2025.

[62] Daniil Shlenskii, Alexander Varlamov, Nazar Buzun, and Alexander Korotin. Unlocking the duality between flow and field matching. In *Advances in Neural Information Processing Systems*, volume 39, 2026.

[63] Stepan Manukhov, Alexander Kolesov, Vladimir V. Palyulin, and Alexander Korotin. Interaction field matching: Overcoming limitations of electrostatic models. In *International Conference on Learning Representations*, 2026.

A Proofs

In this section, we provide proofs for all theorems and propositions in the main part.

A.1 Proof of Proposition 3.1

Let

$$\Delta := X_1 - X_0, \qquad M := X_{1/2} = \frac{X_0 + X_1}{2}.$$

Since X_0 and X_1 are independent samples from the same distribution,

$$(X_0, X_1) \overset{d}{=} (X_1, X_0).$$

Under this exchange, M is unchanged while Δ changes sign. Hence

$$(\Delta, M) \overset{d}{=} (-\Delta, M).$$

Therefore the conditional law of Δ given M is symmetric about the origin, and so

$$\mathbb{E}[\Delta \mid M] = 0 \quad \text{a.s.}$$

Since $\Delta = X_1 - X_0$ and $M = X_{1/2}$, this is exactly

$$v_{1/2}(x) = \mathbb{E}[X_1 - X_0 \mid X_{1/2} = x] = 0$$

for $p_{1/2}$-almost every x.

A.2 Proof of Theorem 3.1

Let

$$\Delta := X_1 - X_0, \qquad M := X_{1/2} = \frac{X_0 + X_1}{2}.$$

Then, by Definition 3.1,

$$D_{\mathrm{mid}}(p_0, p_1) = \mathbb{E}\big[\|\mathbb{E}[\Delta \mid M]\|_2^2\big].$$

If $p_0 = p_1$, then Proposition 3.1 gives

$$\mathbb{E}[\Delta \mid M] = 0 \quad \text{a.s.}$$

and hence $D_{\mathrm{mid}}(p_0, p_1) = 0$.

Conversely, suppose that

$$D_{\mathrm{mid}}(p_0, p_1) = 0.$$

Since the integrand is nonnegative,

$$\mathbb{E}[\Delta \mid M] = 0 \quad \text{a.s.}$$

Since X_0 and X_1 are assumed to have sub-exponential tails, i.e., $\mathbb{E}\, e^{\alpha \|X_j\|_2} < \infty$ for $j \in \{0, 1\}$ and some $\alpha > 0$, X_0 and X_1 have finite moments of all orders, so Δ is integrable. Thus, by the tower property, for every measurable function φ such that $\Delta\, \varphi(M)$ is integrable, we have

$$\mathbb{E}[\Delta\, \varphi(M)] = 0.$$

For every $u \in \mathbb{R}^d$ with $\|u\|_2 < 2\alpha$ (we take α that works for both X_0 and X_1), the function

$$\varphi_u(m) := e^{\langle u, m \rangle}$$

satisfies $\|\Delta\, \varphi_u(M)\|_2 \le (\|X_0\|_2 + \|X_1\|_2)\, e^{\|u\|_2 \|X_0\|_2/2}\, e^{\|u\|_2 \|X_1\|_2/2}$. Since $\|x\|_2\, e^{\|u\|_2 \|x\|_2/2} \le C\, e^{\alpha \|x\|_2}$ for some constant C, the right-hand side is integrable by independence of X_0 and X_1. Therefore, applying the identity above with $\varphi = \varphi_u$ and recalling that $M = (X_0 + X_1)/2$,

$$\mathbb{E}\left[(X_1 - X_0) e^{\langle u, (X_0 + X_1)/2 \rangle}\right] = 0.$$

Using independence of X_0 and X_1, this becomes

$$\mathbb{E}\Big[X_1 e^{\langle u, X_1 \rangle/2}\Big]\, \mathbb{E}\Big[e^{\langle u, X_0 \rangle/2}\Big] = \mathbb{E}\Big[X_0 e^{\langle u, X_0 \rangle/2}\Big]\, \mathbb{E}\Big[e^{\langle u, X_1 \rangle/2}\Big].$$

Define the moment generating functions

$$G_j(v) := \mathbb{E}\Big[e^{\langle v, X_j\rangle}\Big], \qquad j \in \{0,1\}.$$

For $\|v\|_2 < \alpha$, we have $e^{\langle v,x\rangle} \le e^{\alpha\|x\|_2}$, so G_0 and G_1 are finite on the ball $B_\alpha := \{v \in \mathbb{R}^d : \|v\|_2 < \alpha\}$. Moreover, for every $a < \alpha$, the gradient $x\, e^{\langle v,x\rangle}$ is dominated, uniformly over $\|v\|_2 \le a$, by the integrable function $C_a\, e^{\alpha\|x\|_2}$; hence differentiation under expectation is justified, and on B_α

$$\nabla G_j(v) = \mathbb{E}\Big[X_j e^{\langle v, X_j\rangle}\Big].$$

Setting $v = u/2$, the preceding identity yields

$$G_0(v)\nabla G_1(v) = G_1(v)\nabla G_0(v) \qquad \text{for all } v \in B_\alpha.$$

Since $G_0(v) > 0$ and $G_1(v) > 0$,

$$\nabla \log G_1(v) = \nabla \log G_0(v).$$

Hence $\log G_1 - \log G_0$ is constant on the connected set B_α. Evaluating at $v = 0$ gives

$$G_0(0) = G_1(0) = 1,$$

so the constant is zero. Thus

$$G_0(v) = G_1(v) \qquad \text{for all } v \in B_\alpha.$$

By the uniqueness theorem for moment generating functions finite on a neighborhood of the origin, we obtain $p_0 = p_1$. Therefore

$$D_{\mathrm{mid}}(p_0, p_1) = 0 \quad \Longleftrightarrow \quad p_0 = p_1.$$

A.3 Proof of Proposition 3.2

Fix $t \in [0,1]$ and let

$$\Delta := X_1 - X_0.$$

We need to show that

$$\mathbb{E}[\Delta \mid \widetilde{X}_t] = 0.$$

Since X_0 and X_1 are independent samples from the same distribution and $B \sim \mathrm{Bernoulli}(1/2)$ is independent of them,

$$(X_0, X_1, B) \overset{d}{=} (X_1, X_0, 1-B).$$

Under this transformation, Δ changes sign. On the other hand, the flipped observation

$$\widetilde{X}_t = (1-B)\big((1-t)X_0 + tX_1\big) + B\big(tX_0 + (1-t)X_1\big)$$

is unchanged. Therefore

$$(\Delta, \widetilde{X}_t) \overset{d}{=} (-\Delta, \widetilde{X}_t).$$

It follows that the conditional law of Δ given $\widetilde{X}_t$ is symmetric about the origin, and hence

$$\mathbb{E}[\Delta \mid \widetilde{X}_t] = 0 \quad \text{a.s.}$$

Equivalently,

$$\widetilde{v}_t(x) = \mathbb{E}[X_1 - X_0 \mid \widetilde{X}_t = x] = 0$$

for $\widetilde{p}_t$-almost every x.

A.4 Proof of Theorem 3.2

Theorem 3.2 is the deterministic linear interpolant special case of Theorem 3.3. Indeed, take

$$I_t(x_0, x_1) := (1-t)x_0 + tx_1, \qquad \sigma_t := 0.$$

Then the flipped stochastic interpolant in (7) reduces to the flipped linear interpolation in (4):

$$\widetilde{X}_t = \begin{cases} (1-t)X_0 + tX_1, & B = 0, \\ tX_0 + (1-t)X_1, & B = 1. \end{cases}$$

Consequently,

$$D^{I,\sigma}_{t\text{-mid}}(p_0, p_1) = D_{t\text{-mid}}(p_0, p_1).$$

The claim therefore follows directly from Theorem 3.3.

A.5 Proof of Theorem 3.3

Let

$$\Delta := X_1 - X_0, \qquad m_t(x) := \mathbb{E}[\Delta \mid \widetilde{X}_t = x].$$

By definition,

$$D^{I,\sigma}_{t\text{-mid}}(p_0, p_1) = \int_0^{1/2} \mathbb{E}\big[\|m_t(\widetilde{X}_t)\|_2^2\big]\, dt.$$

Nonnegativity is immediate.

First suppose that $p_0 = p_1$. Then X_0 and X_1 are exchangeable. Since $B \sim \mathrm{Bernoulli}(1/2)$ is independent of (X_0, X_1, ϵ), the transformation

$$(X_0, X_1, B, \epsilon) \mapsto (X_1, X_0, 1 - B, \epsilon)$$

preserves the joint distribution. Under this transformation, Δ changes sign. On the other hand, the flipped observation (7), which due to symmetry could be rewritten as

$$\widetilde{X}_t = \begin{cases} I_t(X_0, X_1) + \sigma_t \epsilon, & B = 0, \\ I_t(X_1, X_0) + \sigma_t \epsilon, & B = 1, \end{cases}$$

is unchanged. Hence

$$(\Delta, \widetilde{X}_t) \overset{d}{=} (-\Delta, \widetilde{X}_t).$$

Therefore the conditional law of Δ given $\widetilde{X}_t$ is symmetric about the origin, so

$$m_t(\widetilde{X}_t) = \mathbb{E}[\Delta \mid \widetilde{X}_t] = 0 \quad \text{a.s.}$$

for every $t \in [0, 1/2]$. Thus

$$D^{I,\sigma}_{t\text{-mid}}(p_0, p_1) = 0.$$

Conversely, suppose that

$$D^{I,\sigma}_{t\text{-mid}}(p_0, p_1) = 0.$$

Since the integrand is nonnegative,

$$m_t(\widetilde{X}_t) = \mathbb{E}[\Delta \mid \widetilde{X}_t] = 0 \quad \text{a.s.}$$

for almost every $t \in [0, 1/2]$. Choose a sequence $t_n \downarrow 0$ from this full-measure set. By the endpoint continuity of the stochastic interpolant,

$$\widetilde{X}_{t_n} \longrightarrow Y \quad \text{a.s.},$$

where

$$Y := \begin{cases} X_0, & B = 0, \\ X_1, & B = 1. \end{cases}$$

Let $\varphi : \mathbb{R}^d \to \mathbb{R}$ be bounded and continuous. For each n, since

$$m_{t_n}(\widetilde{X}_{t_n}) = \mathbb{E}[\Delta \mid \widetilde{X}_{t_n}] = 0 \quad \text{a.s.},$$

we have, by the tower property,

$$\begin{aligned} \mathbb{E}\big[\Delta\, \varphi(\widetilde{X}_{t_n})\big] &= \mathbb{E}\big[\mathbb{E}[\Delta \mid \widetilde{X}_{t_n}]\varphi(\widetilde{X}_{t_n})\big] \\ &= \mathbb{E}\big[m_{t_n}(\widetilde{X}_{t_n})\varphi(\widetilde{X}_{t_n})\big] \\ &= 0. \end{aligned}$$

Since $\Delta \in L^1$ and φ is bounded, dominated convergence gives

$$\mathbb{E}[\Delta\, \varphi(Y)] = 0.$$

By a monotone-class argument, the same identity holds for every bounded Borel function φ.

Expanding over the two values of B, we obtain

$$0 = \frac{1}{2}\mathbb{E}\big[(X_1 - X_0)\varphi(X_0)\big] + \frac{1}{2}\mathbb{E}\big[(X_1 - X_0)\varphi(X_1)\big].$$

Let

$$\bar{x}_0 := \mathbb{E}[X_0], \qquad \bar{x}_1 := \mathbb{E}[X_1].$$

Using independence of X_0 and X_1, the previous identity becomes

$$\int \varphi(x)(\bar{x}_1 - x)\, p_0(dx) + \int \varphi(x)(x - \bar{x}_0)\, p_1(dx) = 0$$

for every bounded Borel φ. Taking $\varphi \equiv 1$ gives

$$\bar{x}_0 = \bar{x}_1.$$

Denote the common value by $\bar{x}$. Then

$$\int \varphi(x)(x - \bar{x})\, (p_1 - p_0)(dx) = 0$$

for every bounded Borel φ.

Let

$$\mu := p_1 - p_0.$$

We have shown that

$$(x - \bar{x})\, \mu(dx) = 0$$

as a vector-valued signed measure. We now show that this implies $\mu = 0$.

Fix a coordinate $j \in \{1, \dots, d\}$ and an integer $n \geq 1$, and define

$$A_{j,n} := \{x \in \mathbb{R}^d : \ |x_j - \bar{x}_j| \geq 1/n\}.$$

For any Borel set $A \subseteq A_{j,n}$, the function

$$\varphi(x) := \frac{\mathbf{1}_A(x)}{x_j - \bar{x}_j}$$

is bounded and Borel. Applying the j-th coordinate of the signed-measure identity gives

$$\mu(A) = \int \varphi(x)(x_j - \bar{x}_j)\, \mu(dx) = 0.$$

Thus μ vanishes on every Borel subset of $A_{j,n}$. Since

$$\mathbb{R}^d \setminus \{\bar{x}\} = \bigcup_{j=1}^{d} \bigcup_{n=1}^{\infty} A_{j,n},$$

we have

$$\mu(\mathbb{R}^d \setminus \{\bar{x}\}) = 0.$$

Hence μ is supported on $\{\bar{x}\}$. But μ is the difference of two probability measures, so

$$\mu(\mathbb{R}^d) = p_1(\mathbb{R}^d) - p_0(\mathbb{R}^d) = 0.$$

Therefore $\mu(\{\bar{x}\}) = 0$, and consequently $\mu \equiv 0$. Hence $p_1 \equiv p_0$.

Combining both directions,

$$D^{I,\sigma}_{t\text{-mid}}(p_0, p_1) = 0 \quad \Longleftrightarrow \quad p_0 = p_1.$$

A.6 Proof of Proposition 3.3

Let

$$\Delta := X_1 - X_0, \qquad m_t(x) := \mathbb{E}[\Delta \mid \widetilde{X}_t = x].$$

Then

$$D^{I,\sigma}_{t\text{-mid}}(p_\theta, p_{\text{data}}) = \int_0^{1/2} \mathbb{E}\big[\|m_t(\widetilde{X}_t)\|_2^2\big]\, dt.$$

Fix $t \in [0, 1/2]$ and let f_t be any square-integrable vector-valued function. Recall that

$$m_t(\widetilde{X}_t) = \mathbb{E}[\Delta \mid \widetilde{X}_t].$$

Since $f_t(\widetilde{X}_t)$ is measurable with respect to $\widetilde{X}_t$, the tower property gives

$$\begin{aligned}\mathbb{E}\big[\langle f_t(\widetilde{X}_t), \Delta\rangle\big] &= \mathbb{E}\Big[\mathbb{E}\big[\langle f_t(\widetilde{X}_t), \Delta\rangle \mid \widetilde{X}_t\big]\Big] \\ &= \mathbb{E}\Big[\Big\langle f_t(\widetilde{X}_t), \mathbb{E}[\Delta \mid \widetilde{X}_t]\Big\rangle\Big] \\ &= \mathbb{E}\big[\langle f_t(\widetilde{X}_t), m_t(\widetilde{X}_t)\rangle\big].\end{aligned}$$

The second term does not involve Δ, so it is unchanged. Therefore

$$\mathbb{E}\left[2\langle f_t(\widetilde{X}_t), \Delta\rangle - \|f_t(\widetilde{X}_t)\|_2^2\right] = \mathbb{E}\left[2\langle f_t(\widetilde{X}_t), m_t(\widetilde{X}_t)\rangle - \|f_t(\widetilde{X}_t)\|_2^2\right].$$

Completing the square,

$$2\langle f_t, m_t\rangle - \|f_t\|_2^2 = \|m_t\|_2^2 - \|f_t - m_t\|_2^2.$$

Therefore

$$\mathbb{E}\left[2\langle f_t(\widetilde{X}_t), \Delta\rangle - \|f_t(\widetilde{X}_t)\|_2^2\right] = \mathbb{E}\big[\|m_t(\widetilde{X}_t)\|_2^2\big] - \mathbb{E}\big[\|f_t(\widetilde{X}_t) - m_t(\widetilde{X}_t)\|_2^2\big].$$

The first term is nonnegative and does not depend on f_t, so the expression is maximized when

$$f_t(\widetilde{X}_t) = m_t(\widetilde{X}_t) \quad \text{a.s.}$$

Equivalently,

$$f_t^\star(x) = m_t(x) = \mathbb{E}[X_1 - X_0 \mid \widetilde{X}_t = x]$$

for $\widetilde{p}_t$-almost every x.

Integrating the pointwise variational identity over $t \in [0, 1/2]$ yields

$$D^{I,\sigma}_{t\text{-mid}}(p_\theta, p_{\text{data}}) = \max_{f_t} \int_0^{1/2} \mathbb{E}\left[2\langle f_t(\widetilde{X}_t), X_1 - X_0\rangle - \|f_t(\widetilde{X}_t)\|_2^2\right] dt.$$

This proves the variational representation and identifies the maximizer.

B Algorithm

In this section, we provide a detailed description of the Midpoint Generative Models training procedure. Algorithm 1 gives pseudocode, while Algorithm 2 provides a corresponding Python implementation sketch.

Critic-only noise. In this setting, the generator loss uses the randomly flipped interpolant with its noise term removed, while the critic loss retains the chosen noise schedule. In Algorithm 2, this corresponds to passing `sigma=0` to `mgm_loss_G` and the chosen `sigma` to `mgm_loss_f`.

Simpler variants of our objective. The same training template can also be adapted to simpler variants of our objective, including the midpoint-only divergence D_{mid} in (2) and the naive unflipped time-integrated objective $D^{\text{naive}}_{\text{t-mid}}$ in (3). To recover these functionals exactly, we use the linear interpolant $I_t(x_0, x_1) = (1-t)x_0 + tx_1$ and set $\sigma_t \equiv 0$ in both losses. To obtain the midpoint-only variant, one fixes $t = 1/2$ throughout training instead of sampling $t \sim \mathcal{U}[0, 1/2]$. To obtain the naive unflipped time-integrated variant, one removes the random flip by setting $B \equiv 0$ and samples t over the full interval $[0, 1]$ rather than $[0, 1/2]$. Retaining nonzero noise yields the corresponding stochastic training variants.

C Additional Experimental Details

In this section we provide details specifying the experimental setup of §5.

Algorithm 1 MGM training procedure

Require: Generator G_θ, critic f_ψ, symmetric interpolant (I_t, σ_t)

repeat

 # Update critic f_ψ

Sample $z \sim p_z$, $x_1 \sim p_{\text{data}}$, $t \sim \mathcal{U}[0, 1/2]$, $b \sim \text{Bernoulli}(1/2)$, $\epsilon \sim \mathcal{N}(0, \text{Id})$.

Set $x_0 = G_\theta(z)$.

Define

$$x_t = I_t(x_0, x_1) + \sigma_t \epsilon, \qquad x_{1-t} = I_{1-t}(x_0, x_1) + \sigma_{1-t}\epsilon.$$

Set the flipped observation $\tilde{x}_t = (1 - b)x_t + b x_{1-t}$.

Update ψ by a gradient descent step on

$$\hat{L}_\psi = \|f_\psi(\tilde{x}_t, t) - (x_1 - x_0)\|_2^2.$$

 # Update generator G_θ

Sample a fresh batch and recompute x_0, x_1, and $\tilde{x}_t$ as above.

Update θ by a gradient descent step on

$$\hat{L}_\theta = 2\langle f_\psi(\tilde{x}_t, t), x_1 - x_0\rangle - \|f_\psi(\tilde{x}_t, t)\|_2^2.$$

until convergence

Return G_θ

C.1 General Details

Implementation. Our implementation builds on the SiD codebase[1] and is publicly available at `https://github.com/daniil-shlenskii/mgm`.

Data and architectures. On CIFAR-10 32×32 and ImageNet 64×64, MGM operates in pixel space. On CIFAR-10, the generator and the critic use the DDPM++ architecture from EDM [8] (56M parameters); on ImageNet 64×64, they use the ADM architecture [1] as trained in EDM (296M parameters). On ImageNet 512×512, we follow the latent-space setup of EDM2 [45]: images are encoded by the Stable Diffusion VAE [2] into $64 \times 64 \times 4$ latents, both networks use the EDM2-S architecture (280M parameters), and generated latents are decoded by the frozen VAE decoder before evaluation. In all cases, the generator and the critic share the same architecture and are initialized from the same pretrained network (Appendix C.2).

Evaluation. We report FID computed on 50K generated samples against the reference statistics of the training set provided by EDM [8] and EDM2 [45], using the Inception network from the EDM evaluation code. We evaluate EMA checkpoints of the generator and report the lowest FID obtained during training.

Hyperparameters. Training hyperparameters are summarized in Table 3. The generator and the critic share the same optimizer settings. We maintain an exponential moving average (EMA) of the generator weights only, and the critic is discarded after training. We do not use data augmentation during MGM training.

Resources. A single MGM training run takes approximately 2 days on 2 NVIDIA H100 GPUs for CIFAR-10, 3 days on 8 H100 GPUs for ImageNet 64×64, and 4 days on 8 H100 GPUs for ImageNet 512×512, all in full precision and excluding the initialization stage.

C.2 Initialization and Preconditioning

Pretrained denoiser. In all experiments, both networks use the EDM-style denoiser parameterization [8]

$$D_\eta(x, \sigma) = c_{\text{skip}}(\sigma)\, x + c_{\text{out}}(\sigma)\, F_\eta\big(c_{\text{in}}(\sigma)\, x,\ c_{\text{noise}}(\sigma)\big),$$

where F_η is the raw network, with the standard EDM preconditioning $c_{\text{skip}}(\sigma) = \sigma_{\text{data}}^2/(\sigma^2 + \sigma_{\text{data}}^2)$, $c_{\text{out}}(\sigma) = \sigma\sigma_{\text{data}}/\sqrt{\sigma^2 + \sigma_{\text{data}}^2}$, $c_{\text{in}}(\sigma) = 1/\sqrt{\sigma^2 + \sigma_{\text{data}}^2}$, $c_{\text{noise}}(\sigma) = \frac{1}{4}\log\sigma$, and $\sigma_{\text{data}} = 0.5$. On CIFAR-10, D_η is obtained by the short denoising warmup described in Appendix C.3; on

[1] `https://github.com/mingyuanzhou/SiD`

Algorithm 2 PyTorch-style implementation of the MGM losses

```
# G:   generator network,  z -> x_0
# f:   critic network,     (x, t) -> symmetrized displacement
# sigma: interpolant noise level, sigma_t = sqrt(sigma * t * (1 - t))

def flipped_interpolant(x_0, x_1, sigma):
  t    = rand(N, 1) * 0.5       # [N, 1], uniform in [0, 1/2]
  b    = (rand(N, 1) < 0.5).float() # [N, 1], Bernoulli(1/2)
  noise = randn(N, D)           # [N, D]

  # the two branches of the random time flip (sigma_t = sigma_{1-t})
  sigma_t = sqrt(sigma * t * (1 - t))
  x_t  = (1 - t) * x_0 + t * x_1 + sigma_t * noise
  x_1mt = t * x_0 + (1 - t) * x_1 + sigma_t * noise

  # hide which branch was observed
  x_tilde = (1 - b) * x_t + b * x_1mt # [N, D]
  return x_tilde, t


def mgm_loss_f(G, f, sigma):
  # -------- critic: Flow Matching regression on x_tilde --------
  x_0 = stop_grad(G(sample_noise(N))) # [N, D]
  x_1 = sample_data(N)          # [N, D]
  x_tilde, t = flipped_interpolant(x_0, x_1, sigma)

  target = x_1 - x_0            # [N, D]
  pred = f(x_tilde, t)          # [N, D]
  return (pred - target).pow(2).sum(-1).mean()


def mgm_loss_G(G, f, sigma):
  # -------- generator: variational midpoint objective --------
  x_0 = G(sample_noise(N))      # [N, D]
  x_1 = sample_data(N)          # [N, D]
  x_tilde, t = flipped_interpolant(x_0, x_1, sigma)

  target = x_1 - x_0            # [N, D]
  pred = f(x_tilde, t)          # [N, D]
  return (pred * (2 * target - pred)).sum(-1).mean()
```

ImageNet 64×64 and 512×512, we use the publicly available EDM and EDM2-S checkpoints, respectively. For class-conditional models, both networks additionally receive the class label.

Generator. Following SiD [31], the generator is initialized as a one-step denoiser applied to scaled Gaussian noise:

$$G_\theta(z) = D_\theta(\sigma_{\text{init}} z,\, \sigma_{\text{init}}), \qquad z \sim \mathcal{N}(0, \text{Id}), \qquad \theta \leftarrow \eta,$$

with $\sigma_{\text{init}} = 2.5$, matching the maximal noise level used during warmup (Appendix C.3).

Critic. The critic reuses the same network and preconditioning with two modifications:

$$f_\psi(x, t) = D_\psi(x,\, \sigma_{\min} + t) - x, \qquad \psi \leftarrow \eta,$$

where $\sigma_{\min} = 0.002$ matches the default minimum sampling noise level in EDM. First, since c_{noise} is undefined at $\sigma = 0$, we shift the noise level by $\sigma_{\min}$, so that the critic is well defined for all $t \in [0, 1]$. Second, the critic outputs the residual $D_\psi - x$ without time-dependent rescaling. In Flow Matching, the velocity is obtained from the denoiser output by rescaling $D - x$ with a time-dependent factor. Under the random time flip, however, this factor depends on the unobserved flip B: it differs between the original and the flipped branch. To avoid committing to either branch, we omit the rescaling and let the network absorb the scale.

	CIFAR-10	**ImageNet** 64×64	**ImageNet** 512×512
Space	pixel	pixel	latent (SD VAE)
Architecture	DDPM++ (56M)	ADM (296M)	EDM2-S (280M)
Initialization	50K kimgs warmup	EDM checkpoint	EDM2-S checkpoint
Batch size	256	2048	256
Adam (β_1, β_2)		$(0, 0.999)$	
Learning rate		10^{-5}	
Critic:generator updates		1:1	
Generator EMA decay		0.999	
Noise σ	0.01	0.2	0.5
Critic-only noise		Yes	

Table 3: Training hyperparameters used for the baseline comparison in §5.3.

C.3 Denoising Warmup

On CIFAR-10, we obtain the pretrained denoiser D_η (Appendix C.2) by a short denoising warmup: we train the DDPM++ network from scratch with the EDM objective and training hyperparameters [8] for 50K kimgs (25% of the EDM training budget), unless stated otherwise (Table 1.a). This warmup is used only to obtain a reasonable initialization; we do not aim to train a full diffusion model.

The only modification of EDM training concerns the sampled noise levels. EDM samples log-noise values

$$n \sim \mathcal{N}(-1.2,\, 1.2^2), \qquad \sigma = \exp(n),$$

which covers noise levels far beyond those used by MGM: the generator is evaluated at $\sigma_{\text{init}} = 2.5$, while the critic uses conditioning noise levels in $[\sigma_{\min}, \sigma_{\min} + 1/2]$. We therefore restrict the noise levels to $\sigma \leq \sigma_{\max}$ with $\sigma_{\max} = \sigma_{\text{init}} = 2.5$ by reflecting the log-noise value around $\log \sigma_{\max}$:

$$s = \begin{cases} n, & n \leq \log \sigma_{\max}, \\ 2 \log \sigma_{\max} - n, & n > \log \sigma_{\max}, \end{cases} \qquad \sigma = \exp(s).$$

This folded log-normal distribution reallocates the high-noise tail to noise levels below $\sigma_{\max}$, concentrating the warmup budget on the range relevant for MGM.

D Additional Results

In this section we provide additional results extending experiments of §5.

D.1 Trigonometric Interpolant

To assess the sensitivity of MGM to the choice of the interpolation path, we replace the linear deterministic component $I_t(x_0, x_1) = (1-t)x_0 + tx_1$ with the trigonometric interpolant [21]

$$I_t(x_0, x_1) = \cos\left(\tfrac{\pi t}{2}\right) x_0 + \sin\left(\tfrac{\pi t}{2}\right) x_1.$$

It satisfies the endpoint conditions and the symmetry $I_t(x_0, x_1) = I_{1-t}(x_1, x_0)$ required in §3.3, since $\cos(\pi(1-t)/2) = \sin(\pi t/2)$. We keep the noise schedule and all other settings of the CIFAR-10 configuration used in the main comparison fixed. The trigonometric interpolant reaches FID 2.52 compared to 2.27 for the linear one, indicating that MGM is not tied to a particular interpolation path.

D.2 Critic-to-Generator Update Ratio

Since MGM is trained with a minimax objective, we examine its sensitivity to the number of critic updates per generator update, keeping all other settings of the CIFAR-10 configuration used in the main comparison fixed. With update ratios 1:1, 2:1, and 4:1, MGM reaches FID 2.27, 1.99, and 3.21, respectively. The effect is non-monotonic: a moderate increase in critic updates improves FID, whereas a larger one degrades it. Thus, MGM does not require multiple critic updates per generator

update, although tuning the ratio can further improve performance. We use the 1:1 ratio throughout the main experiments for consistency.

D.3 Seed Variability

To quantify run-to-run variability, we repeat the CIFAR-10 configuration used in the main comparison with two additional random seeds. The three runs reach FID 2.27, 2.24, and 2.29, i.e., 2.27 ± 0.02 (mean $\pm$ standard deviation). This indicates low sensitivity to the random seed within the used training regime.

D.4 Noise Level on ImageNet 64×64

We repeat the noise ablations of §5.2 on class-conditional ImageNet 64×64 (Table 4). The results follow the same qualitative pattern as on CIFAR-10: the deterministic setting $\sigma = 0$ performs poorly, moderate noise gives the best results in both settings, and large noise $\sigma = 0.5$ degrades training, most severely when noise is added only to the critic. Compared to CIFAR-10, however, the optimal noise level is larger: $\sigma = 0.1$–0.2 instead of 0.01–0.05. With critic-only noise at $\sigma = 0.2$, MGM achieves FID 2.50, which we report in Table 2.

σ	0.00	0.01	0.05	0.10	0.20	0.50
Both interpolants	5.40	4.90	2.67	**2.54**	2.74	15.71
Critic only	5.62	3.53	3.18	2.65	**2.50**	> 100

Table 4: FID-50K on class-conditional ImageNet 64×64 for different noise levels σ, with noise added to both the generator and critic interpolants or to the critic interpolant only. The best result in each row is in **bold**.

E Electrostatic Interpretation for the Midpoint Divergence

Recently, electrostatics has inspired a new family of generative models [60, 61, 35]. In this section, we show that the midpoint symmetry of Flow Matching established in §3 has a natural connection to the method of images in electrostatics. We first recall the electrostatic view of generative modeling. We then turn to the method of images: a plane is equipotential whenever the charges on its two sides are mirror images of each other with opposite signs. Finally, using the duality between flow and field matching [62], we show how this mirror symmetry naturally suggests the vanishing midpoint velocity of Flow Matching. This discussion is intended as intuition and is not used in our proofs.

Electrostatic basics. EFM [61] augments the data space $\mathbb{R}^d$ with an extra coordinate z. Points of $\mathbb{R}^{d+1}$ are written as $\tilde{x} = (x, z)$, and a vector field splits into its data and z-components, $\mathbf{E} = (E_x, E_z)$. A unit point charge at $\tilde{y} \in \mathbb{R}^{d+1}$ creates the Coulomb field

$$\mathbf{E}_{\tilde{y}}(\tilde{x}) = \frac{1}{S_d} \frac{\tilde{x} - \tilde{y}}{\|\tilde{x} - \tilde{y}\|^{d+1}},$$

where S_d is the surface area of the unit sphere in $\mathbb{R}^{d+1}$. The Coulomb field is conservative, meaning that it can be written as the negative gradient of a scalar potential. For $d \geq 2$, this potential is

$$\varphi_{\tilde{y}}(\tilde{x}) = \frac{1}{(d-1)S_d} \frac{1}{\|\tilde{x} - \tilde{y}\|^{d-1}}, \qquad \mathbf{E}_{\tilde{y}} = -\nabla \varphi_{\tilde{y}}.$$

By the superposition principle, a charge density q creates the field and potential

$$\mathbf{E}(\tilde{x}) = \int \mathbf{E}_{\tilde{y}}(\tilde{x})\, q(\tilde{y})\, d\tilde{y}, \qquad \varphi(\tilde{x}) = \int \varphi_{\tilde{y}}(\tilde{x})\, q(\tilde{y})\, d\tilde{y},$$

with $\mathbf{E} = -\nabla \varphi$.

EFM represents two distributions as plates of a capacitor, placing p_0 as positive charge on the plane $z = 0$ and p_1 as negative charge on the plane $z = L$. The resulting charge distribution is

$$q(y, z) = p_0(y)\, \delta(z) - p_1(y)\, \delta(z - L).$$

Samples are transported between the plates along the field lines. Inside the layer $0 < z < L$, the positive plate repels and the negative plate attracts, so both contribute a positive z-component. Hence $E_z > 0$, and z increases strictly along each field line and can serve as its parameter. For a field line $(x(\tau), z(\tau))$ satisfying $d(x, z)/d\tau = \mathbf{E}(x, z)$, the chain rule gives

$$\frac{dx}{dz} = \frac{dx/d\tau}{dz/d\tau} = \frac{E_x(x, z)}{E_z(x, z)}.$$

Some field lines of the capacitor leave the layer, but only the field inside the layer matters for the discussion below.

Method of images. The classical method of images pairs charges with mirror charges of opposite signs. Every point on the reflection plane is equally distant from the two charges in each pair, so their potential contributions cancel there. When $p_0 = p_1$, the EFM plates form exactly this configuration with respect to the midplane $z = L/2$. The midplane is therefore equipotential, and the tangential field vanishes on it,

$$\varphi(x, L/2) = 0, \qquad E_x(x, L/2) = -\nabla_x \varphi(x, L/2) = 0.$$

Since $E_z > 0$, the field-line slope dx/dz also vanishes at the midplane, providing the electrostatic counterpart of $v_{1/2} = 0$.

The same mirror configuration gives $\varphi(x, L - z) = -\varphi(x, z)$ throughout the layer. Differentiating in x and z yields

$$E_x(x, L - z) = -E_x(x, z), \qquad E_z(x, L - z) = E_z(x, z).$$

The field-line slope therefore satisfies

$$\frac{E_x(x, L - z)}{E_z(x, L - z)} = -\frac{E_x(x, z)}{E_z(x, z)}.$$

From Field Matching to Flow Matching. Shlenskii et al. [62] establish a duality between flow-based generative models and forward-only interaction fields in an augmented space. Under this duality, Flow Matching corresponds to an interaction field [63] on the layer $\mathbb{R}^d \times [0, 1]$, where the time t plays the role of the extra coordinate z and $L = 1$:

$$\mathbf{E}(x, t) = \big(v_t(x)\, p_t(x),\ p_t(x)\big), \qquad \frac{E_x(x, t)}{E_t(x, t)} = v_t(x) \quad \text{wherever } p_t(x) > 0.$$

This field is not a Coulomb field and, in general, not the gradient of any potential, so equipotentiality itself does not carry over to Flow Matching. The slope of the field lines, however, plays the same role in both models: it is E_x/E_z in EFM and the velocity v_t in Flow Matching. For linear Flow Matching with independent endpoints and $p_0 = p_1$, the analogous mirror symmetry is

$$p_{1-t} = p_t, \qquad v_{1-t} = -v_t,$$

where the velocity antisymmetry was established by Wang et al. [20]. At $t = 1/2$, this antisymmetry gives $v_{1/2} = 0$, recovering Proposition 3.1.

F Computational Cost

In this section, we compare the training cost of MGM with the 21 CIFAR-10 baselines in Table 2. A direct comparison is difficult: most papers omit accelerator count or wall-clock time, architectures differ in size, and training may require additional networks, initialization, trajectory sampling, or multiple stages. Moreover, a single loss evaluation may involve several network calls, ODE integration, or Jacobian-vector products (JVPs). Raw iteration and example counts are therefore not directly comparable.

F.1 Training Cost Estimation

We report compute in millions of forward-pass equivalents of a 56M-parameter network (**M FPE56**). We use 56M because EDM DDPM++ [8] is the most common backbone in this CIFAR-10 comparison. We assign the following costs:

$$F = 1, \qquad B = 2, \qquad F_B = 3, \qquad F_{\mathrm{JVP}} = 2, \qquad F_{\mathrm{JVP},B} = 4,$$

where F and B denote a forward and a backward pass, F_B a forward-backward pass, F_{JVP} a joint forward-JVP evaluation, and $F_{\mathrm{JVP},B}$ its evaluation with a backward pass. For a network with P million parameters, we scale the cost by $P/56$:

$$\text{M FPE56} = \frac{\text{iterations} \times \text{batch size}}{10^6} \times \text{operation cost} \times \frac{P}{56}. \tag{11}$$

We sum this expression over the required stages and network calls. A frozen network still incurs a forward pass, and we count F_B when gradients must propagate through it to its input. Near-56M backbones are rounded to 56M. We scale iCT-deep (112M), the StyleGAN2 generator and discriminator (19.95M and 20.72M), the DFNO student and teacher (65.77M and 60M), and the VCT encoder (1.6M) separately.

Initialization and post-initialization cost. *Post-init* is the work performed after the required checkpoints are available. *Init* is the work used to obtain these checkpoints, including teacher pretraining where applicable. We report Total = Post-init+Init. The standard 56M EDM initialization costs $200 \times 3 = 600$ M FPE56, corresponding to 200M processed training images. Image counts include repeated presentations of the training data. We include online teacher calls and offline trajectory generation when their budgets are available. The estimates describe a single training pipeline and exclude hyperparameter searches, additional seeds, and FID evaluation.

Scope of the estimates. FPE56 provides an approximate estimate of computational cost. It does not directly measure FLOPs, GPU-hours, or wall-clock time. In particular, scaling by parameter count does not capture all differences in architectures, memory access, precision, or implementations. When a budget is not reported, we mark it as unknown. The notation $\geq$ denotes a lower bound under this accounting: the corresponding LPIPS, VGG, DSM, GAN, feature-loss, or R1 computations are not included. These omissions are specified in Tables 6 and 7. Our measured training times are reported separately in Appendix C.1.

F.2 MGM Training Cost

We consider the main CIFAR-10 configuration with FID 2.27 and one critic update per generator update. The denoising warmup processes 50M images, one quarter of the 200M-image EDM initialization budget, and trains a single 56M network. Its weights initialize both the generator and the critic, so this warmup is counted once:

$$C_{\mathrm{init}} = 50 \times 3 = 150 \quad \text{M FPE56}.$$

The subsequent MGM stage uses $78{,}125$ iterations with batch size 256, corresponding to 20M examples for each update branch in Algorithm 1. The critic update requires a stop-gradient generator forward and a critic forward-backward pass, costing $F + F_B = 4$ per example. The generator update uses a fresh batch and requires a generator forward-backward pass and differentiation through the critic to its input, costing $2F_B = 6$. Thus,

$$C_{\mathrm{post}} = 20 \times (4 + 6) = 200, \qquad C_{\mathrm{total}} = 150 + 200 = 350 \quad \text{M FPE56}.$$

This estimate includes both networks and their recomputation between updates. It applies to the 1:1 configuration, rather than the 2:1 update-ratio ablation in Appendix D.2.

F.3 Comparison with Baselines

Table 5 summarizes the estimates, and Tables 6 and 7 give the schedules and per-operation calculations. As in Table 2, we distinguish whether a pretrained model provides a training signal, rather than whether initialization is used. Within this comparison, MGM achieves the best FID without a teacher-model training signal and the third-best FID overall, behind CTM and SiD. MGM has the lowest complete post-initialization estimate at 200 M FPE56. The closest estimates are 204.8 for ECT, at least 204.8 for CMT w/ ECT, and 209.2 for VCT. Including initialization gives 350 M FPE56 for MGM, compared with 804.8, at least 804.8, and 809.2 M FPE56, respectively. Under this accounting, MGM reduces the total training cost by at least 56% relative to these three alternatives. This comparison uses the reported training configurations and does not establish a ranking at a matched training budget or a corresponding wall-clock speedup.

Method	Post-init	Init	Total
Without teacher-model training signal			
MGM (ours)	**200**	**150**	**350**
DiffRatio-DiJS [53]	Unknown	600	Unknown (≥ 600)
iCT-deep [15]	3276.8	0	3276.8
iCT [15]	1638.4	0	1638.4
sCT [23]	Unknown	614.4	Unknown (≥ 614.4)
MeanFlow [17]	3276.8	0	3276.8
Stable CT [51]	614.4	600	1214.4
StyleGAN2+ADA+tuning [50]	≥ 475.5	0	≥ 475.5
IMM [48]	6553.6	0	6553.6
VCT [47]	209.2	600	809.2
ECT [26]	204.8	600	804.8
With teacher-model training signal			
CTM [24]	≥ 802.1	600	≥ 1402.1
SiD [31]	6500	600	7100
TCM [27]	1331.2	600	1931.2
CMT w/ ECT [52]	≥ 204.8	600	≥ 804.8
CD [11]	≥ 2457.6	614.4	≥ 3072
sCD [23]	Unknown	614.4	Unknown (≥ 614.4)
DMD [12]	≥ 1311.1	600	≥ 1911.1
DFNO [49]	≥ 909.4	329.1	≥ 1238.5
TRACT [25]	1536	600	2136
2-Rectified Flow [19]	Unknown	Unknown	Unknown
PD [10]	$[480, 2048]$	614.4	$[1094.4, 2662.4]$

Table 5: **Training-cost estimates on unconditional CIFAR-10.** All costs are in M FPE56. The MGM initialization is one quarter of the standard EDM initialization. Unknown denotes an unreported budget, and $\geq$ indicates an incomplete cost estimate with the omissions listed in Tables 6 and 7. FID values are reported in Table 2.

Method	Post-initialization schedule and calculation
MGM	$78{,}125 \times 256 = 20$M examples per update branch. One critic update and one generator update cost $F + 3F_B = 10$ per example, giving $20 \times 10 = 200$.
DiffRatio-DiJS	Batch size 64. The number of iterations and the checkpoint used for the reported FID are not specified, so post-init cost is unknown.
iCT-deep	$400\text{k} \times 1024 = 409.6$M examples. Each iteration uses $F + F_B = 4$, with a 112M-parameter network: $409.6 \times 4 \times 112/56 = 3276.8$.
iCT	$400\text{k} \times 1024 = 409.6$M examples. Each iteration uses $F + F_B = 4$, giving $409.6 \times 4 = 1638.4$.
sCT	Batch size 512, with $F_B + F_{\mathrm{JVP}} = 5$ per example. The number of training iterations is not reported. Initialization uses 204.8M examples, giving $204.8 \times 3 = 614.4$.
MeanFlow	$800\text{k} \times 1024 = 819.2$M examples. Each iteration uses $F_{\mathrm{JVP},B} = 4$, giving $819.2 \times 4 = 3276.8$.
Stable CT	$300\text{k} \times 512 = 153.6$M examples. Each iteration uses $F + F_B = 4$, giving $153.6 \times 4 = 614.4$.
StyleGAN2 +ADA+tuning	$1{,}562{,}500 \times 64 = 100$M examples. The generator uses $F + F_B = 4$ and the discriminator uses $3F_B = 9$. With 19.95M and 20.72M parameters, respectively, the estimate is $\geq 100 \times (4 \times 19.95 + 9 \times 20.72)/56 = 475.5$, excluding R1.
IMM	$400\text{k} \times 4096 = 1638.4$M examples. Each iteration uses $F + F_B = 4$, giving $1638.4 \times 4 = 6553.6$.
VCT	$400\text{k} \times 128 = 51.2$M examples. The backbone uses $F + F_B = 4$, and the 1.6M-parameter encoder uses $F_B = 3$: $51.2 \times (4 \times 56 + 3 \times 1.6)/56 = 209.2$.
ECT	$400\text{k} \times 128 = 51.2$M examples. Each iteration uses $F + F_B = 4$, giving $51.2 \times 4 = 204.8$.

Table 6: **Cost accounting without a teacher-model training signal.** All computed costs are in M FPE56, $\mathrm{k} = 10^3$, and M denotes 10^6 processed examples. Method references and initialization costs are given in Table 5.

Method	Post-initialization schedule and calculation
CTM	$100\text{k} \times 256 = 25.6\text{M}$ examples. Heun step counts $k \in \{1, \ldots, 17\}$ are sampled proportionally to k, giving $2\sum k^2 / \sum k \approx 23.333$ teacher forwards per example. The total is $25.333F + 2F_B = 31.333$, giving $\geq 25.6 \times 31.333 = 802.1$, excluding DSM, GAN, and feature losses.
SiD	$1{,}953{,}125 \times 256 = 500\text{M}$ examples. An update cycle uses $F + 4F_B = 13$, giving $500 \times 13 = 6500$.
TCM	$250\text{k} \times 512 + 200\text{k} \times 1024 = 332.8\text{M}$ examples across two stages. Each stage uses $F + F_B = 4$, giving $332.8 \times 4 = 1331.2$. Stage 2 uses either the frozen Stage-1 or current Stage-2 target.
CMT w/ ECT	$300\text{k} \times 128 + 100\text{k} \times 128 = 51.2\text{M}$ examples. Mid-training obtains 16 training pairs per 16-NFE teacher trajectory, averaging one teacher forward per pair. ECT post-training instead uses one target forward per example. Both stages additionally require a student forward-backward pass, giving $F + F_B = 4$ per example and $\geq 51.2 \times 4 = 204.8$ in total, excluding LPIPS.
CD	$800\text{k} \times 512 = 409.6\text{M}$ examples. Each iteration uses $3F + F_B = 6$, giving $\geq 409.6 \times 6 = 2457.6$, excluding LPIPS.
sCD	Batch size 512, with $F + F_{\text{JVP}} + F_B = 6$ per example. The number of training iterations is not reported.
DMD	$300\text{k} \times 392 = 117.6\text{M}$ examples, with $2F + 3F_B = 11$ per example, plus 0.5M offline pairs at $35F$ per pair. The estimate is $\geq 117.6 \times 11 + 0.5 \times 35 = 1311.1$, excluding LPIPS.
DFNO	$400\text{k} \times 256 = 102.4\text{M}$ student examples with $F_B = 3$, plus 1M teacher trajectories at $512F$ each. Scaling the 65.77M student and 60M teacher separately gives $\geq 102.4 \times 3 \times 65.77/56 + 512 \times 60/56 = 909.4$, excluding VGG. Teacher initialization costs 329.1.
TRACT	$500\text{k} \times 512 = 256\text{M}$ examples. Each iteration uses $3F + F_B = 6$, giving $256 \times 6 = 1536$.
2-Rectified Flow	The reflow stage uses 300k iterations, but its batch size, initial 1-RF training budget, pair-generation NFE, and final distillation budget are not reported.
PD	In Consistency Models [11], the authors state: "For progressive distillation, we used the same training settings as those described in Salimans & Ho (2022)." However, their released configuration uses a batch size of 512 and approximately 800k updates. The original CIFAR-10 schedule uses a batch size of 128 and 750k updates across all stages for one-step generation [10]. With $2F + F_B = 5$ per example, these schedules give 480 and 2048 M FPE56, respectively. To reflect this discrepancy, we report a post-initialization cost range of $[480, 2048]$ M FPE56, or $[1094.4, 2662.4]$ M FPE56 including initialization.

Table 7: **Cost accounting with a teacher-model training signal.** All computed costs are in M FPE56. CD, sCD, and PD use a 204.8M-example initialization, costing 614.4 M FPE56. The other initialization costs and method references are given in Table 5.

G Additional Samples from MGM



Figure 6: One-step unconditional CIFAR-10 32×32 samples generated by MGM (FID=2.27).



Figure 7: One-step class-conditional ImageNet 64×64 samples generated by MGM (FID=2.50).

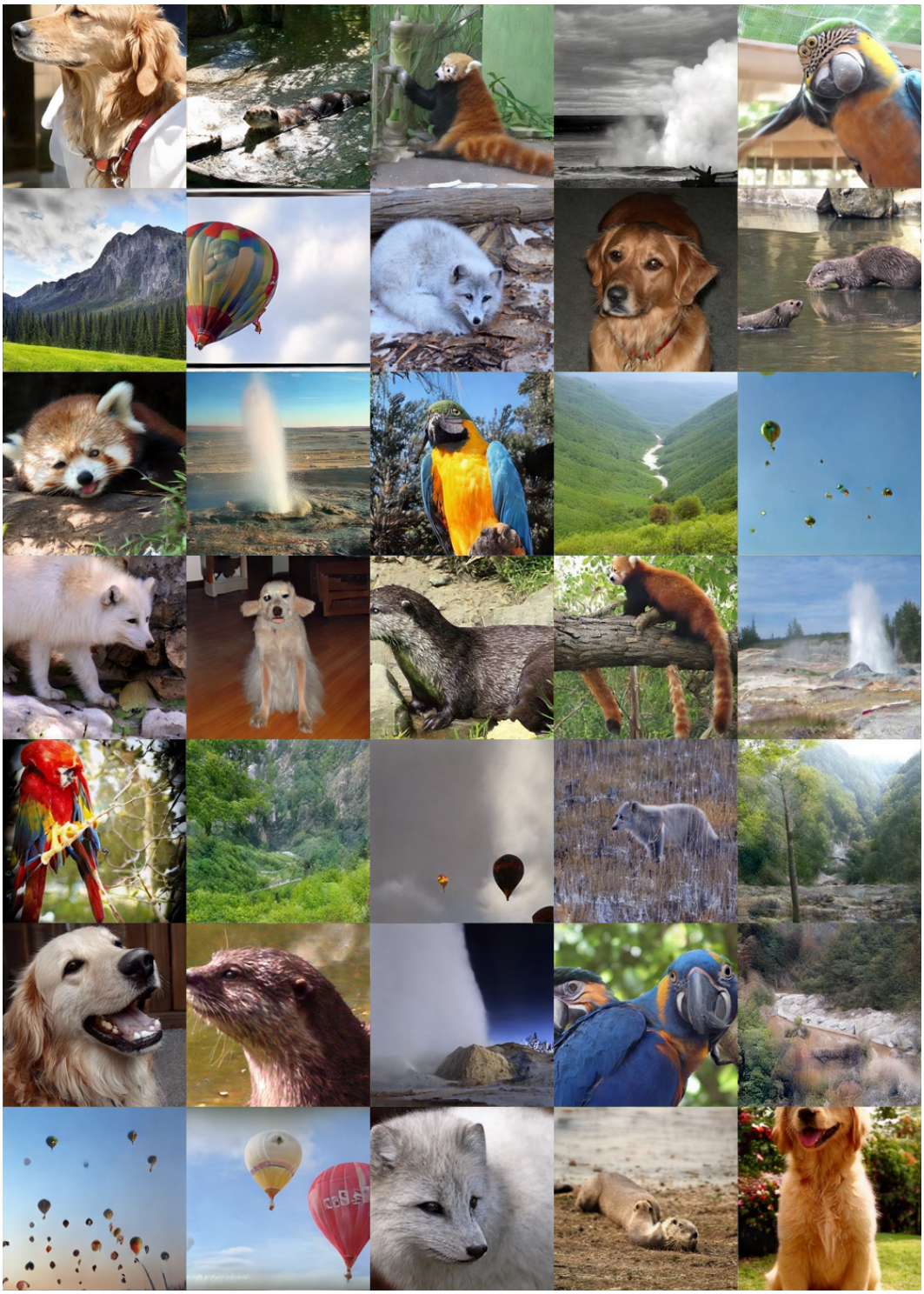


Figure 8: One-step class-conditional ImageNet 512×512 samples generated by MGM (FID=2.36).

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 claims are supported by the theory in §3 and experiments in §5, with limitations discussed in §6.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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: Limitations are discussed in §6.

   Guidelines:

   - The answer [N/A] 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: Assumptions and results are stated in §3, with proofs in Appendix A.

Guidelines:

- The answer [N/A] 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: Algorithms and settings are given in Appendices B and C.

   Guidelines:

   - The answer [N/A] 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: Code and reproduction instructions are available in the repository linked in Appendix C.1.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Experimental settings are described in §5 and Appendix C, with hyperparameters in Table 3.

   Guidelines:

   - The answer [N/A] 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: Appendix D.3 reports the mean and standard deviation across three CIFAR-10 training seeds. Other experiments omit error bars due to computational cost.

   Guidelines:

   - The answer [N/A] 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: Hardware and runtime per MGM run are reported in Appendix C.1.

   Guidelines:

   - The answer [N/A] 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: The work conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] 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: [No]

    Justification: The paper focuses on methodology and benchmarks (§3 and §5) and does not discuss societal impacts.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Experiments use standard image benchmarks (§5); no high-risk assets are released.

    Guidelines:

    - The answer [N/A] 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: [No]

    Justification: Assets are credited in §5 and Appendix C.1, but licenses and terms are not explicitly listed.

    Guidelines:

    - The answer [N/A] 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: [Yes]

    Justification: The MGM implementation is documented in the repository linked in Appendix C.1.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: No crowdsourcing or human-subject studies are involved.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: No human-subject studies are involved.

    Guidelines:

    - The answer [N/A] 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 rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were used only for writing, editing, and formatting.

    Guidelines:

    - The answer [N/A] 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 in the NeurIPS handbook for what should or should not be described.