MSE Training Induces Diffusions That Are Nearly Flows

Anonymous authors
Paper under double-blind review

Abstract

In the DDPM paper, Ho, Jain, and Abbeel introduced two reversible diffusion processes parameterized by a noise schedule—a generator and an oracle process that the generator learns from—and derived a formula for the Kullback-Leibler divergence (KL) between these processes in the form of a time-weighted Mean Squared Error (MSE). However, they empirically found that omitting the weights improved performance on image synthesis benchmarks—a result later corroborated by many studies. More recently, removing the stochastic component during generation time has proven effective. (1) In this work, we provide a theoretical justification for these practices. We consider a broader class of diffusion processes (not necessarily reversible) parameterized by a noise schedule and a diffusion size b that share the same marginals. Since the weight associated with the MSE depends on b, omitting the weight is equivalent to solving the equation weight(b)=1, which yields a unique "MSE-diffusion". For SOTA models, we checked that b is close to zero—the learned MSE-diffusion is nearly a flow—and we confirm this observation by comparing generators on ImageNet 512×512. These results suggest that flows beat reversible diffusions because training of SOTA models is an implementation of KL minimization for MSE-diffusions, which are nearly flows. (2) Moreover, we obtain a novel representation of the diffusion state as the sum of an explicit linear component, an unweighted pathwise integral of the denoiser, and a noise term. This representation offers the advantages of DPM-solvers while enabling the use of classical ODE methods.

1 Introduction

A typical diffusion model for image generation transforms noise into an image over a few dozen to a few hundred time steps by means of a neural network, known as a denoiser. The denoiser is trained with a time-weighted mean-squared error (MSE) between the noised image and the network's prediction. Averaging is performed over independent trios (time, noise, image), thus no diffusion model is needed to train the denoiser. The weights associated with MSE are often absorbed into the noised image, yielding different parameterizations of the prediction (e.g., noise-, data-, or velocity parameterization). Therefore, the denoiser is determined by the noise schedule and the chosen parameterization.

On the other hand, a diffusion model is specified by its noise schedule and diffusion size (variance or rate coefficient). To train a diffusion model, we need a measure of proximity between 2 diffusion processes: a generator and an oracle process that the generator learns from. The processes are defined by distributions, and MSE does not determine the proximity between them, so we may use the most popular measure for this purpose, that is the Kullback-Leibler divergence (KL) or, equivalently, the maximum likelihood (ML).

This brings us to the problem: how to combine the practical training of the denoiser with the theoretical training of the diffusion process, and then determine the diffusion size for a generator based on the noise schedule and parameterization used in training. This issue has been known since at least the work of Ho et al. (2020). They expressed the KL between reversible diffusion processes as a time-weighted MSE. However, empirical results showed that dropping the weights improved performance—a finding later confirmed by many studies. Since then, the problem has become more entrenched because several different noise schedules and parameterizations have appeared, while

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

diffusion models are defined using a noise schedule and the reversibility condition (Ho et al., 2020; Song et al., 2021c; Nichol & Dhariwal, 2021; Dhariwal & Nichol, 2021; Salimans & Ho, 2021; Song et al., 2021b; Kingma et al., 2021; Ho & Salimans, 2022; Rombach et al., 2022; Kingma & Gao, 2023; Esser et al., 2024). More recently, it has also proven effective to zero out the diffusion size during sampling, replacing the reversible diffusion with a deterministic flow. We validate this claim in the review of state-of-the-art (SOTA) models presented in Section 6.

Main result. We provide a theoretical justification for omitting weights during training and removing the stochastic component during generation. By considering a broader class of (not necessarily reversible) diffusion processes sharing the same marginals, we generalize the KL formulations of Sohl-Dickstein et al. (2015); Ho et al. (2020) in Proposition 4.1, and those of Kingma et al. (2021); Song et al. (2021b) in Proposition 4.2. Subsequently, leveraging Proposition 5.1, we derive closed-form or approximate expressions for the diffusion size of the unique *MSE-diffusion* learned via KL minimization, given a noise schedule and a denoiser parameterization. Moreover, in Section 6, we checked that the MSE-diffusion size is close to zero for current SOTA models; that is, the MSE-diffusion is nearly a flow. We then confirm this observation by comparing generators on the ImageNet 512×512 class-conditional benchmark. Therefore, the practical training of denoisers for generators of deterministic flows can be understood as an implementation of KL minimization for MSE-diffusion processes, which are nearly flows.

Additional results. By applying elementary autoregressive arguments to both discrete- and continuous-time diffusion processes, we derive formulas that are significantly simpler than those in prior work. Specifically, in Proposition 3.1 we provide a novel representation of the diffusion state as the sum of an explicit linear component, an unweighted pathwise integral of the denoiser, and a noise term. This representation offers the advantages of DPM-solvers (Lu et al., 2023; 2025; Gonzalez et al., 2023; Cui et al., 2025; Zhang & Chen, 2023), while additionally enabling the use of classical methods of numerical integration for differential equations.

2 A DENOISER INDUCED BY MSE TRAINING

Assume that we have two positive, continuously differentiable functions of time $t \in [0, 1]$, namely increasing α_t and decreasing σ_t. Let $\mathring{\rho}_t := \alpha_t/\sigma_t$ and $\mathring{\lambda}_t := \log \mathring{\rho}_t$ denote *signal-to-noise ratio* and its logarithm, respectively. We shall also refer to these as the *noise schedule*. The ring accents indicate that these functions are special cases of a more general class of functions, denoted without rings and defined later. Let $t \sim U(0, 1), X \sim p_x$ in $\mathbb{R}^d$ and $\varepsilon \sim \mathcal{N}(0, I_d)$ be independent. We consider the *linear noise generators*

$$\tilde{Y}_t \;:=\; X + \mathring{\rho}_t^{-1}\,\varepsilon \quad \text{and} \quad \tilde{Z}_t \;:=\; \alpha_t \tilde{Y}_t \;=\; \alpha_t\, X + \sigma_t\, \varepsilon. \tag{1}$$

We train a *prediction* $\hat{u}_t : \mathbb{R}^d \to \mathbb{R}^d$ by fitting its parameters, denoted as a hat, according to the *mean squared error* (MSE)

$$\min_{\wedge}\; \mathbb{E}_{t,\varepsilon,X} \left\| \hat{u}_t(\alpha_t \tilde{Y}_t) - u_t \right\|^2 \;=\; \min_{\wedge} \int_0^1 \mathbb{E}_{\varepsilon,X} \left\| \hat{u}_t(\alpha_t \tilde{Y}_t) - u_t \right\|^2 \mathrm{d}t,$$

where $u_t := A_t \tilde{Y}_t + S_t \varepsilon$ is a target and functions A_t, S_t are scaling schedules, with positive and continuous S_t called *parameterization*. Observe that averaging is performed over independent trios (t, ε, X), thus no diffusion process is needed to train $\hat{u}_t$. Note also that in our setting, unlike in the original works α_t is an increasing function.

From a unique $\hat{u}_t$ we recover an estimator of the *noise* using the formula for the target u_t, along with a *denoiser* or a *data* estimator via (1)

$$\hat{\varepsilon}_t(\tilde{Y}_t) :=\; (\hat{u}_t(\alpha_t \tilde{Y}_t) - A_t\, \tilde{Y}_t)/S_t \quad \text{and} \quad \hat{X}_t(\tilde{Y}_t) :=\; \tilde{Y}_t - \mathring{\rho}_t^{-1} \hat{\varepsilon}_t(\tilde{Y}_t).$$

A direct calculation shows that $\hat{u}_t - u_t = S_t(\hat{\varepsilon}_t - \varepsilon) = -\mathring{\rho}_t S_t(\hat{X}_t - X)$. We can also define a target using data: $u_t := B_t \tilde{Y}_t - C_t X$, then for this target learn the network, define $\hat{X}_t$, and then, using (1), define $\hat{\varepsilon}_t$ and set $S_t := C_t/\mathring{\rho}_t$.

For our purposes, the interface between the denoiser and the generator consists of the pair $(\mathring{\lambda}_t, S_t)$, in addition to $\hat{\varepsilon}_t$ or $\hat{X}_t$. These can be viewed as input and output normalization, respectively. In Appendix A, we review popular noise schedules and parameterizations, including those used in current SOTA models.

3 DIFFUSION MODELS INDUCED BY THE DENOISER

A diffusion denoising model generating samples from the estimated distribution p_x is a diffusion process that starts with noise and has a drift estimated by a denoiser. We begin this section by defining a discrete-time noise process, which constitutes an autoregressive model. We will then incorporate a denoiser to obtain a denoising process, and subsequently define an analogous continuous-time process and its numerical solver. Finally, we will state our main problem.

Stationary, time-inhomogeneous autoregression. Let us fix a grid $0 < t_0 < t_1 < \cdots < t_N = t \leq t_{max} < 1$, where $t_i := t_0 + i(t_{max} - t_0)/N, i = 0, 1, \ldots, N$. Let ρ_t be a positive, increasing, and continuously differentiable function of time $t \in (0, 1)$, which we will refer to as the *diffusion schedule*. We also define $\lambda_t := \log \rho_t$ and $r_t := \rho_t \mathring{\rho}_t$. Note that r_t is likewise a positive, increasing, and continuously differentiable function of time.

Let $\varepsilon_{t_0} := \varepsilon, \xi^N_{t_0}, \ldots, \xi^N_{t_N}$ be i.i.d. $\mathcal{N}(0, I_d)$ and for $t = t_i, s = t_{i-1}$ set

$$\rho_t \varepsilon_t := \rho_s \varepsilon_s + \sqrt{\rho_t^2 - \rho_s^2}\, \xi^N_s. \tag{2}$$

Equation (2) directly implies that $\varepsilon_t \sim \mathcal{N}(0, I_d)$, and that ε_s is independent of ξ^N_s. The correlation and conditional variance are also easily computable for coordinates $j, k = 1, 2, \ldots, d$

$$\operatorname{cor}(\varepsilon_{t,j}, \varepsilon_{s,k}) = \mathbf{1}(j = k)\rho_s/\rho_t \quad \text{and} \quad \operatorname{var}(\varepsilon_{t,j} \mid \varepsilon_{s,k}) = 1 - \mathbf{1}(j = k)\rho_s^2/\rho_t^2.$$

These formulas indicate that ρ_t^2 represents the relative cumulative variance of the autoregression model $\{\varepsilon_{t_i}\}_{i=0}^N$.

Discrete-time diffusion models. Assume that $\{\varepsilon_{t_i}\}_{i=0}^N$ are independent of X. Let us define

$$Y_t := X + \mathring{\rho}_t^{-1}\, \varepsilon_t \;\text{ and }\; Z_t := \alpha_t Y_t = \alpha_t\, X + \sigma_t\, \varepsilon_t. \tag{3}$$

By construction, for any diffusion schedule ρ_t, the random variables Y_t, Z_t are distributed identically to the linear noise generators $\tilde{Y}_t, \tilde{Z}_t$ defined in (1) at times $t = t_i$. Inserting the expressions for $\varepsilon_t, \varepsilon_s$ in terms of Y_t, Y_s or Z_t, Z_s (from (3)) into (2), we get

$$r_t Y_t = r_s Y_s + (r_t - r_s)X + \sqrt{\rho_t^2 - \rho_s^2}\, \xi^N_s, \tag{4}$$

$$\frac{\rho_t}{\sigma_t} Z_t = \frac{\rho_s}{\sigma_s}\, Z_s + (r_t - r_s)X + \sqrt{\rho_t^2 - \rho_s^2}\, \xi^N_s. \tag{5}$$

By substituting the prediction induced by the denoiser, $\hat{X}_s \equiv \hat{X}_s(\hat{Y}_s)$, for X in the oracle process (4), we obtain the *generator*

$$\hat{Y}_{t_0} := \mathring{\rho}_{t_0}^{-1} \varepsilon_{t_0} \quad \text{and} \quad r_t \hat{Y}_t := r_s \hat{Y}_s + (r_t - r_s)\hat{X}_s + \sqrt{\rho_t^2 - \rho_s^2}\, \xi^N_s. \tag{6}$$

Clearly, the processes in (4)–(5) are not realistic generators of the distribution p_x, since they depend on the ground-truth value X. We therefore refer to them as an *oracle*.

In our setting, unlike in many works on generative diffusion models, time runs forward in both the oracle and the generator. Moreover, the function α_t, previously identified as an internal learning function, turns out to be unnecessary for generation. It suffices to generate the process Y_t and, if needed, scale it at the end of generation to obtain $Z_{t_N} = \alpha_{t_N} Y_{t_N}$. Hence, throughout this section, we consider only the process Y_t.

Continuous-time diffusion models. We define the continuous-time diffusion denoising process as any stochastic process that satisfies (7) below.

Proposition 3.1 *Assume that the denoiser $\hat{X}_\tau(y)$ is continuous in τ and Lipschitz in y. Then, for any $t_0 < s < t \leq t_{\max}$, the discrete-time generator admits the continuous-time limit*

$$r_t \hat{Y}_t = r_s \hat{Y}_s + \int_{r_s}^{r_t} \hat{X}_{\tau(r)}\big(\hat{Y}_{\tau(r)}\big)\mathrm{d}r + \sqrt{\rho_t^2 - \rho_s^2}\, \xi_s, \tag{7}$$

where $\hat{Y}_s$ and ξ_s are independent, $\xi_s \sim \mathcal{N}(0, I_d)$, and τ denotes the inverse of $t \mapsto r_t$. Equivalently,

$$\hat{Y}_t = q_{st}\hat{Y}_s + (1 - q_{st})\mathbb{E}_{q\sim U(q_{st},1)}\hat{X}_{\tau(q)}(\hat{Y}_{\tau(q)}) + \beta_{st}\xi_s, \tag{8}$$

where $q_{st} := r_s/r_t$ and $\beta_{st} := \mathring{\rho}_t^{-1}\sqrt{1 - \rho_s^2/\rho_t^2}$.

The coefficient β_{st} is called the *diffusion size*. Proposition 3.1 is proven in Appendix B.

The representation (7)–(8) of the diffusion state—as the sum of an explicit linear component, an unweighted path integral of the denoiser, and a noise term—appears to be new in the context of generative models. Given (7), there is no need to explicitly invoke SDEs. Moreover, the corresponding SDE can be naturally derived from (6). By applying the variation-of-constants method to this SDE, we recover (7), which reduces back to (6) under the simplest discretization scheme–the full derivation is provided in Appendix B.

If $\beta \equiv 0$, then $r \equiv \mathring{\rho}$ and (7) is a deterministic flow equation equivalent to the differential equation

$$\check{Y}'_r = \hat{X}_r(\check{Y}_r), \tag{9}$$

where $\check{Y}_r := r\hat{Y}_{\tau(r)}$, $\hat{X}_r(\check{Y}_r) := \hat{X}_{\tau(r)}(\hat{Y}_{\tau(r)})$. Interestingly, this equation is "dual" to the equation

$$\hat{Y}'_{1/r} = \hat{\varepsilon}_{1/r}(\hat{Y}_{1/r}) \tag{10}$$

utilized by popular generators, such as EDM and EDM2 (Song et al., 2021c; Karras et al., 2022; 2024b).

We also observe that generation based on (8) requires only the quotients of ρ_t—or equivalently, λ_t up to an additive constant—allowing it to be determined solely by the derivative λ'_t, hereafter referred to as the *diffusion rate*.

Diffusion size. The input data X—representing either pixel intensities after preprocessing or their latent/token representations—are typically brought to a "convenient scale" in one of two ways: (1) $\text{var}(X) \approx 1$ in the latent models (Peebles & Xie, 2023; Ma et al., 2024; Esser et al., 2024; Leng et al., 2025; Yu et al., 2025; Leng et al., 2025; Zheng et al., 2025a); (2) $\text{var}(X) \approx 1/4$ in the EDM-style preconditioning (Karras et al., 2022; 2024b;a), as well as the pixel-based models SiD-SiD2 (Hoogeboom et al., 2023; 2025) and JiT (Li & He, 2025). As a consequence of data normalization, the deterministic component of (8) scales with X, as we mix the previous prediction $\hat{Y}_s$ with the mean prediction over the transformed interval (s, t). Given this behavior, the coefficient β of the standard normal noise justifies the name *diffusion size*.

Universal diffusion solver (UDS). The continuous-time model offers the flexibility of being discretized in multiple ways. By replacing the integral in (8) with any numerical ODE integration method, we obtain the UDS family. The simplest cases are the Euler and Heun schemes, where the Heun step depends on a prior estimate from the Euler method

$$\textit{Euler:}\quad \hat{Y}_t := q_{st}\hat{Y}_s + (1 - q_{st})\hat{X}_s(\hat{Y}_s) + \beta_{st}\xi_s, \tag{11}$$

$$\textit{Heun:}\quad \hat{Y}_t := q_{st}\hat{Y}_s + (1 - q_{st})\frac{\hat{X}_s(\hat{Y}_s) + \hat{X}_t(\hat{Y}_t)}{2} + \beta_{st}\xi_s. \tag{12}$$

Observe that EulerUDS coincides with (6). It reduces to the DDPM solver (Ho et al., 2020) for reversible diffusions, where $\beta_{st} = \mathring{\rho}_t^{-1}\sqrt{1 - \mathring{\rho}_s^2/\mathring{\rho}_t^2}$, and to the DDIM solver (Song et al., 2021a) for deterministic flows, where $\beta_{st} = 0$.

Universal schemes such as the Euler-Maruyama method struggle with integrating the rapidly-changing linear term, whereas schemes designed specifically for diffusion, such as DPM-solvers, treat the linear term analytically. However, they require integrating $\hat{\varepsilon}_t$ or $\hat{X}_t$ with an exponential weight, a procedure which is complicate (Lu et al., 2023; 2025; Gonzalez et al., 2023; Cui et al., 2025; Zhang & Chen, 2023). In particular, DPM solvers of order 1-3 are analogous to the Runge-Kutta methods, but the analogue for the Runge-Kutta method of order 4, has not yet been developed. For comparison, our method is both universal and specific to diffusion. A more detailed discussion is provided in Appendix C.

The main problem. Given that $\mathring{\rho}_0 = 0$ and $\mathring{\rho}_1 = \infty$ for popular noise schedules, the generation process is not well-defined at the boundaries, so we must specify an interval $[t_0, t_{max}] \subset (0, 1)$. This leads to the core problem addressed in this work: determining the optimal diffusion rate λ'_t based on the denoiser training parameters $(\mathring{\lambda}_t, S_t)$ within $[t_0, t_{max}]$.

4 A DIFFUSION MODEL INDUCED BY THE PENALIZED MAXIMUM LIKELIHOOD

To choose the optimal diffusion schedule, we require a measure of proximity between the oracle process and the generator. Since these processes are defined by distributions, and a standard MSE does not directly determine the proximity between them, we employ the most common measures for this purpose: maximum likelihood (ML) or, equivalently, the Kullback-Leibler divergence (KL).

4.1 DIFFUSION LOSS

As in the previous sections, we initially consider the processes $Z_t = \alpha_t Y_t$ and $\hat{Z}_t = \alpha_t \hat{Y}_t$ to see that it is enough to consider only Y_t and $\hat{Y}_t$. Let $0 < t_0 < t_1 < \cdots < t_N = t < 1$ be a discretization of the time interval $[t_0, t]$, and set $s := t_{N-1}$. For $x \sim p_x$ we denote latent variables along the path by z_{t_i} and write $z_{t_i:t} = (z_{t_i}, z_{t_{i+1}}, \dots, z_t)$. From the definition of the oracle process (5) and the generator (6) it follows that

$$p_t(z_t|z_s, x) \;=\; \mathcal{N}\Big(z_t|\mu_s(z_s, x), \sigma_t^2(1 - \rho_s^2/\rho_t^2)I_d\Big),$$

$$\text{where}\quad \mu_s(z_s, x) \;:=\; \frac{\sigma_t\rho_s}{\sigma_s\rho_t}z_s + \frac{\sigma_t}{\rho_t}(\rho_t\mathring{\rho}_t - \rho_s\mathring{\rho}_s)X \;=\; \frac{\alpha_t}{\alpha_s}z_s - \sigma_t\Big(\frac{\mathring{\rho}_t}{\mathring{\rho}_s} - \frac{\rho_s}{\rho_t}\Big)\varepsilon_s,$$

and analogously

$$\hat{p}_t(z_t|z_s) \;=\; \mathcal{N}\Big(z_t|\hat{\mu}_s(z_s), \sigma_t^2(1 - \rho_s^2/\rho_t^2)I_d\Big),$$

$$\text{where}\quad \hat{\mu}_s(z_s) \;:=\; \frac{\sigma_t\rho_s}{\sigma_s\rho_t}z_s + \frac{\sigma_t}{\rho_t}(\rho_t\mathring{\rho}_t - \rho_s\mathring{\rho}_s)\hat{X}_s \;=\; \frac{\alpha_t}{\alpha_s}z_s - \sigma_t\Big(\frac{\mathring{\rho}_t}{\mathring{\rho}_s} - \frac{\rho_s}{\rho_t}\Big)\hat{\varepsilon}_s.$$

The KL divergence between these two distributions is

$$\mathbb{D}\big[p_t(.|z_s, x)\big\|\,\hat{p}_t(.|z_s)\big] \;:=\; \mathbb{E}_{Z_t|z_s,x}\log\big[p_t(Z_t|z_s, x)/\,\hat{p}_t(Z_t|z_s)\big] \;=\; \frac{\|\mu_s(z_s, x) - \hat{\mu}_s(z_s)\|^2}{2\sigma_t^2(1 - \rho_s^2/\rho_t^2)} \tag{13}$$

$$=\; w_s^N\,\|\hat{\varepsilon}_s(z_s) - \varepsilon_s\|^2, \quad \text{where}\quad w_s^N \;:=\; \frac{(\mathring{\rho}_t/\mathring{\rho}_s - \rho_s/\rho_t)^2}{2(1 - \rho_s^2/\rho_t^2)}. \tag{14}$$

We define the following conditional and joint distributions

$$\begin{aligned}
p^N_{t_1:t}(z_{t_1:t}|z_{t_0}, x) &:= p_t(z_t|z_s, x)\dots p_{t_1}(z_{t_1}|z_{t_0}, x),\\
\hat{p}^N_{t_1:t}(z_{t_1:t}|z_{t_0}) &:= \hat{p}_t(z_t|z_s)\dots\hat{p}_{t_1}(z_{t_1}|z_{t_0}),\\
p_{t_0,x}(z_{t_0}, x) &:= p_{t_0}(z_{t_0}|x)\,p_x(x),\\
\overline{p}(z_{t_0}, x|z_t) &:= p_{t_0}(z_{t_0})\,\overline{p}_t(x|z_t),\\
p^N_{t_0:t,x}(z_{t_0:t}, x) &:= p^N_{t_1:t}(z_{t_1:t}|z_{t_0}, x)p_{t_0,x}(z_{t_0}, x),\\
\hat{p}^N_{t_0:t,x}(z_{t_0:t}, x) &:= \hat{p}^N_{t_1:t}(z_{t_1:t}|z_{t_0})\overline{p}(z_{t_0}, x|z_t).
\end{aligned}$$

The distribution $\overline{p}_t$ represents the "reconstruction term" which determines how well the image was recovered from the final z_t representation. The bar symbol denotes the parameters of this distribution; its contribution to the divergence between the joint distributions will be referred to as the *bias* and denoted below as $\bar{\mathcal{B}}$. In this work, we focus on fitting a hat, assuming that looking for a bar can be handled using standard techniques.

Two KL decompositions that we shall use are

$$\begin{aligned}
\mathbb{D}\big[p^N_{t_0:t,x}\,\big\|\,\hat{p}^N_{t_0:t,x}\big] &= \mathbb{D}\big[p_x\,\big\|\,\hat{p}^N_x\big] \;+\; \mathbb{E}_X\,\mathbb{D}\big[p^N_{t_0:t}(.|X)\,\big\|\,\hat{p}^N_{t_0:t}(.|X)\big],\\
\mathbb{D}\big[p^N_{t_0:t,x}\,\big\|\,\hat{p}^N_{t_0:t,x}\big] &= \hat{\mathcal{L}}^N(t_0, t, \mathring{\lambda}, \lambda) \;+\; \bar{\mathcal{B}}(t_0, t),
\end{aligned}$$

where

$$\hat{\mathcal{L}}^N(t_0, t, \mathring{\lambda}, \lambda) := \mathbb{E}_{Z_{t_0},X}\, \mathbb{D}\big[p^N_{t_1:t}(.|Z_{t_0}, X) \,\big\|\, \hat{p}^N_{t_1:t}(.|Z_{t_0})\big]$$

and

$$\bar{\mathcal{B}}(t_0, t) := \mathbb{E}_{Z_{t_0},Z_t,X} \log\big[p_{t_0,x}(Z_{t_0}, X)\big/\,\overline{p}(Z_{t_0}, X|Z_t)\big].$$

Both of these decompositions together imply that the *diffusion loss*, denoted as $\hat{\mathcal{L}}^N$, is the objective function for (implicitly) a penalized negative log-likelihood of the estimator $\hat{p}^N_x$ of the density p_x induced by the denoiser with parameters $\wedge$. The diffusion loss for a discrete-time diffusion model can be expressed as a weighted MSE on the grid points. Moreover, as $N \to \infty$, it has a limit analogous to the training loss.

Proposition 4.1

$$\hat{\mathcal{L}}^N(t_0, t, \mathring{\lambda}, \lambda) = \sum_{i=0}^{N-1} w^N_{t_i} \mathbb{E}_{\varepsilon,X} \big\|\hat{\varepsilon}_{t_i}(\tilde{Y}_{t_i}) - \varepsilon\big\|^2.$$

Proposition 4.2 *Assume that the function $t \mapsto \hat{\varepsilon}_t(\hat{Y}_t)$ is continuous. Then, as $N \to \infty$,*

$$\hat{\mathcal{L}}^N(t_0, t, \mathring{\lambda}, \lambda) = \hat{\mathcal{L}}(t_0, t, \mathring{\lambda}, \lambda') + o(1),$$

where

$$\hat{\mathcal{L}}(t_0, t, \mathring{\lambda}, \lambda') := \int_{t_0}^{t} \frac{\big(\lambda'_\tau + \mathring{\lambda}'_\tau\big)^2}{4\,\lambda'_\tau}\, \mathbb{E}_{\varepsilon,X} \big\|\hat{\varepsilon}_\tau(\tilde{Y}_\tau) - \varepsilon\big\|^2 d\tau. \tag{15}$$

In the above formula, $\tilde{Y}_\tau$ depends on $\mathring{\lambda}_\tau$, and therefore $\hat{\mathcal{L}}$ depends on $\mathring{\lambda}$. Since the loss depends on the diffusion process only through $\mathring{\lambda}_t$ and λ'_t, it suffices, for maximum-likelihood analysis, to consider the processes Y_t and $\hat{Y}_t$. Propositions 4.1–4.2 are proved in Appendix B.

We generalize the generative diffusion model and the formulas for the diffusion loss from Sohl-Dickstein et al. (2015); Ho et al. (2020), Kingma et al. (2021) and Song et al. (2021b). Formula (15) is equivalent to the expressions derived in Huang et al. (2021) and Sabour et al. (2024), but we do not use stochastic calculus in its proof. Despite the generalization, our continuous-time diffusion construction is much simpler than previous ones: time runs forward and there is no need to consider SDEs at all. A more detailed comparison and discussion is provided in Appendix C.

4.2 A DIFFUSION MODEL INDUCED BY THE PENALIZED MAXIMUM LIKELIHOOD OBJECTIVE

Observe that the weights under the integral in (15) are of the form $(\lambda'_t + \mathring{\lambda}'_t)^2/(4\lambda'_t) = \mathring{\lambda}'_t + \chi^2(\lambda'_t, \mathring{\lambda}'_t)/4$, where $\chi^2(\lambda'_t, \mathring{\lambda}'_t) := \big(\lambda'_t - \mathring{\lambda}'_t\big)^2/\lambda'_t$ is the well-known χ^2-distance. Thus, the weights are (up to a constant) the scaled distance between λ'_t and $\mathring{\lambda}'_t$.

Since the function $\hat{\mathcal{L}}$ is (implicitly) a penalized ML objective, we do not change its meaning or difficulty of its calculation, if we add a simple penalty to the weights and define (explicitly) the *penalized maximum likelihood* objective

$$\hat{\mathcal{L}}_c(t_0, t, \mathring{\lambda}, \lambda') := \int_{t_0}^{t} \left[\mathring{\lambda}'_\tau + \frac{\chi^2(\lambda'_\tau, \mathring{\lambda}'_\tau) + c_\tau \lambda'_\tau}{4}\right] \mathbb{E}_{\varepsilon,X} \big\|\hat{\varepsilon}_\tau(\tilde{Y}_\tau) - \varepsilon\big\|^2 d\tau,$$

for some non-negative, continuous function c_t. Indeed, optimizing these penalized weights directly yields

$$\lambda'_{c,t} := \mathring{\lambda}'_t/\sqrt{1 + c_t}. \tag{16}$$

Penalized maximum likelihood encompasses several prominent approaches. For $c_t = 0$, we obtain the maximum (joint) likelihood solution $\lambda'_{c,t} = \mathring{\lambda}'_t$, hereinafter referred as *ML-diffusion*—which, as previously shown, constitutes a reversible diffusion (Ho et al., 2020; Kingma et al., 2021; Kingma & Gao, 2023). Conversely, as $c_t \to \infty$, then $\lambda'_{c,t} \to 0$, and the diffusion process converges to a deterministic flow.

Intermediate solutions in the form $\lambda'_{\gamma,t} = \gamma \mathring{\lambda}'_t$ for $0 \le \gamma \le 1$ have been explored by Huang et al. (2021); Zhang & Chen (2021; 2023) and Cui et al. (2025), making them equivalent to $\lambda'_{c,t}$ with

$c_t = (1-\gamma^2)/\gamma^2$. A similar approach is also employed in the SiT, REPA and REPA-E model families (Ma et al., 2024; Yu et al., 2025; Leng et al., 2025). Specifically, for a *Logis* noise schedule, they defined $\lambda'_{c,t} := 1/(2\sigma_t) = 1/(2(1-t)) \leq 1/(t(1-t)) = \mathring{\lambda}'_t$, so $c_t = (4-t^2)/t^2$. An analogous concept for discrete-time processes gives rise to DDIM variants (Song et al., 2021a; Zhang et al., 2023; Zheng et al., 2025b). Our MSE-diffusion, defined in the following section, likewise represents an intermediate process between a reversible diffusion and a deterministic flow.

5 MSE-DIFFUSION: A DIFFUSION MODEL INDUCED BY MSE TRAINING

Assume that we train the neural network using Stochastic Gradient Descent (SGD) with a batch consisting of a single example. Specifically, we sample time t, noise ε, and an image X, and then update the network weights (a hat) by subtracting the normalized gradient of the training error $||\hat{u}_t - u_t||^2 = S_t^2||\hat{\varepsilon}_t - \varepsilon||^2$. In comparison, the theoretical error given by (15) is $(\mathring{\lambda}'_t + \lambda'_t)^2/(4\lambda'_t)||\hat{\varepsilon}_t - \varepsilon||^2$. For SGD to behave identically under both the practical and theoretical errors, these functions must match, which leads to the following condition.

KL–MSE coherence condition. The diffusion process defined by $(t_0, t_{max}, \mathring{\lambda}, \lambda'_c)$ is coherent with MSE training iff there exist a constant $M \equiv M(t_0, t_{max}, \mathring{\lambda}, S)$ such that $\forall t \in [t_0, t_{max}]$ we have

$$(\mathring{\lambda}'_t + \lambda'_t)^2/(4\lambda'_t) = MS_t^2. \tag{17}$$

The diffusion loss (without subscript c) is invariant to data scaling because it represents an expected divergence, whereas the MSE is scale-dependent. Consequently, the normalization constant M is indeed necessary to equate these two functions. Now we state the main result.

Proposition 5.1 *Let $M := \max_{t\in[t_0,t_{max}]} \mathring{\lambda}'_t/S_t^2$. The KL–MSE coherence condition holds iff*

$$\lambda'_{t,c} = \left(\sqrt{M}S_t - \sqrt{MS_t^2 - \mathring{\lambda}'_t}\right)^2. \tag{18}$$

The diffusion with $\check{\lambda}_t{}' \equiv \lambda'_{t,c}$ is called the *MSE-diffusion*. Proposition 5.1 is proven in Appendix B.

Optimality of MSE-diffusion. We require the diffusion loss and the MSE objective to align not only globally on the interval $[t_0, t_{\max}]$, but also on each of its subintervals, therefore we generalize the MSE to an arbitrary initial interval (t_0, t) as

$$\hat{\mathcal{M}}(t_0, t, \mathring{\lambda}, S) := \int_{t_0}^{t} \mathbb{E}_{\varepsilon,X} \left\| \hat{u}_\tau(\alpha_\tau \tilde{Y}_\tau) - u_\tau \right\|^2 \mathrm{d}\tau = \int_{t_0}^{t} S_\tau^2 \, \mathbb{E}_{\varepsilon,X} \left\| \hat{\varepsilon}_\tau(\tilde{Y}_\tau) - \varepsilon_\tau \right\|^2 \mathrm{d}\tau.$$

The quadratic equation (17) in the variable λ'_t has two roots, but only the smaller one minimizes the penalized maximum-likelihood objective, namely MSE-diffusion $\check{\lambda}_t{}'$ (see the proof of Proposition 5.1). For the square root in (18) to be well defined, M must satisfy

$$M \geq \mathring{\lambda}'_t/S_t^2 \;\text{ for }\; t \in [t_0, t\max]. \tag{19}$$

Now take the smallest M satisfying (19) that is M defined in Proposition 5.1 and let $\tilde{M} > M$ define another diffusion $\tilde{\lambda}'_t$ via (18). Thus $\hat{\mathcal{L}}(\check{\lambda}'_t) = M\hat{\mathcal{M}} < \tilde{M}\hat{\mathcal{M}} = \hat{\mathcal{L}}(\tilde{\lambda}'_t)$. Therefore, among the diffusions coherent with MSE training, the MSE-diffusion minimizes the theoretical error.

Example. Consider the *Logis* noise schedule with the *Vel* parametrization: $\alpha_t := t, \sigma_t := 1-t, \mathring{\lambda}'_t = 1/[t(1-t)], S_t = 1/t$ and $g_t := \sqrt{(t_{max}-t)/(t_{max}(1-t))}$. Thus $M = t_{max}/(1-t_{max})$ and $\check{\lambda}_t{}' = \left(\sqrt{M} - \sqrt{M - t/(1-t)}\right)^2 \Big/ t^2$. In this case, we obtain a compact expression for $\check{\lambda}_t$

$$\check{\lambda}_t = -\log\Big(\frac{t}{1-t}\Big) - \log\Big(\frac{1+g_t}{1-g_t}\Big) + \frac{2\,t_{max}}{1-t_{max}} \, \frac{1-t}{t} \, \big(g_t - 1\big) + c.$$

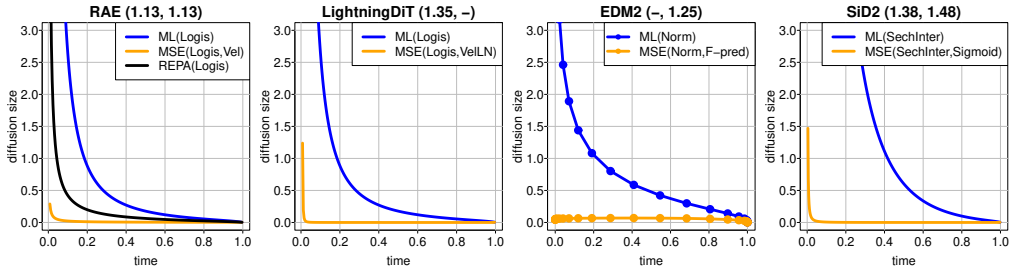


Figure 1: **Diffusion sizes of current SOTA models.** The FID scores for the ImageNet 256×256 and ImageNet 512×512 class-conditional benchmarks are indicated in parentheses next to the model names. In the case of EDM2, the generator uses only 32 steps on a non-uniform grid, which we have marked with dots. In all other cases, the generators operate on a uniform grid, with default settings of 250 steps for RAE and LightningDiT, and 512 steps for SiD2.

Discrete time. The KL–MSE coherence condition and the MSE-diffusion each have an analogue in the discrete model. Let us define $\eta_{st} := \sqrt{1-\rho_s^2/\rho_t^2} = \mathring{\rho}_t\beta_{st}$ and $s := t-\Delta t$, $\Delta t := (t_{max}-t_0)/N$. Rewriting (2) we see that η_{st}^2 is the proportion of new noise ξ_s to the total noise ε_t

$$\varepsilon_t \;=\; \sqrt{1-\eta_{st}^2}\,\varepsilon_s \;+\; \eta_{st}\,\xi_s.$$

From (14)

$$w_t^N(\eta_{st}) = \eta_{st}^{-2}\big(\sqrt{\gamma_{st}} \;-\; \sqrt{1-\eta_{st}^2}\big)^2, \;\text{ where }\; \gamma_{st} := \mathring{\rho}_t^2/\mathring{\rho}_s^2.$$

A straightforward calculation shows that

$$\mathring{\eta}_{st}^N := \arg\min_{\eta} w_t^N(\eta) = \sqrt{1-\gamma_{st}^{-1}} \quad\text{and}\quad \check{\eta}_{st}^N := \frac{\gamma_{st}-1}{\sqrt{2MS_t^2\gamma_{st}}+\sqrt{2MS_t^2+1-\gamma_{st}}},$$

where $M := \max_{t\in\{t_0,\dots,t_{max}\}}(\gamma_{st}-1)/(2S_t^2)$. In the next section, we plot the diffusion size β_{st} using the approximation based on both λ_t' and η_{st}^N

$$2\Delta t\lambda_t' \approx 2(\lambda_t-\lambda_s) \approx (1+\rho_s/\rho_t)(1-\rho_s/\rho_t) \approx (\eta_{st}^N)^2, \quad\text{so}\quad \sqrt{2\Delta t\lambda_t'}/\mathring{\rho}_t \approx \beta_{st} \approx \eta_{st}^N/\mathring{\rho}_t.$$

6 Experiment and Discussion

In this section, we present a concise review of SOTA models and argue, based on experimental evidence, that their corresponding MSE-diffusions are almost flows. We also discuss MSE-diffusion as a generator benchmark. Let ML($\mathring{\lambda}_t$) and MSE($\mathring{\lambda}_t, S_t$) denote the diffusion size $\beta_{t-\Delta t,t}$ for ML-diffusion and MSE-diffusion, respectively.

Current open-source SOTA "diffusion models" generate deterministic flows. It is natural to group SOTA models for image generation according to their neural network architecture—either ViT (Dosovitskiy et al., 2021) or U-Net (Ronneberger et al., 2015). This applies to both ImageNet class-conditioned models and text-to-image models. The largest group consists of ViT-based models, which can be conveniently divided into two subgroups depending on whether Saining Xie is listed as a co-author. The subsequent models by Xie include DiT (Peebles & Xie, 2023), SiT (Ma et al., 2024), REPA (Yu et al., 2025), REPA-E (Leng et al., 2025) and RAE (Zheng et al., 2025a). The original DiT employed the noise schedule, parameterization, and generator from DDPM, whereas the later models use the *Logis* noise schedule and *Vel* parameterization. The latest RAE model generates a flow ($\beta_{t-\Delta t,t}=0$). Other ViT-based models include DDT (Wang et al., 2025), LightningDiT (Yao et al., 2025), JiT (Li & He, 2025), and popular text-to-image models such as Stable Diffusion 3 (SD3) (Esser et al., 2024) and PixArt (Chen et al., 2024b;a). All of them generate flows: DDT employs *Logis* noise schedule and *Vel* parameterization; SD3, LightningDiT, and JiT models use the *Logis* with *VelLN*; while PixArt uses a beta-linear noise schedule and *Noise* parameterization. Thus, the latest SOTA "diffusion models" based on ViT generate deterministic flows.

Table 1: **FID scores of different generators using the EDM2 model trained on ImageNet 512×512.** We run generators in the EDM2-S without CFG. NFE: number of function evaluations. Reported values are means of 4 runs.

	HeunEDM2	HeunUDS, (12)			EulerUDS, (11)		
NFE	Flow	Flow	MSE	ML	Flow	MSE	ML
63	2.63	2.63	2.66	3.40	2.84	2.88	3.97
255	2.57	2.57	2.57	2.88	2.59	2.60	2.97

SOTA "diffusion models" based on U-Net originate from the DDPM network and its mature modification, ADM (Dhariwal & Nichol, 2021), which operated with the *Sech* noise schedule and *Noise* parameterization and used the DDPM generator. These models can be further divided according to the research group. The first subgroup consists of models developed at Stability AI: LDM-8, Stable Diffusion 1, see Rombach et al. (2022), and subsequently modified at NVIDIA: EDM (Karras et al., 2022) and EDM2 (Karras et al., 2024b;a). These recent models use a *Normal* noise schedule, *F-pred* parameterization, and generate flows. The second subgroup consists of models developed at Google: SiD (Hoogeboom et al., 2023) and SiD2 (Hoogeboom et al., 2025). These models use a *SechInter* noise schedule, and *Noise* and *Sigmoid* parameterization, respectively. They are the only SOTA "diffusion models" known to us that generate reversible diffusion (approximately, due to interpolation between the prior and posterior variance), but they are not publicly available.

MSE-diffusions are nearly flows. Figure 1 illustrates the diffusion size $\beta_{t-\Delta t,t}$ for ML- and MSE-diffusion, evaluated on current SOTA models for class-conditional generation on the ImageNet benchmark (Deng et al., 2009). It can be observed that the MSE-diffusions are close to flows for all models except at the initial stage of generation, which should not matter given that the generator starts from a normal distribution and the diffusion noise is also normal. In the later stages of generation, the diffusion size is below 1% of the image size in all models, with the exception of EDM2, where it remains at 10% (5% in the Figure corresponds to 10% of the image size, as $\text{var}(X) \approx 1/4$). Therefore, we decided to verify whether such values imply that the flow generator and the MSE-diffusion are practically indistinguishable in terms of FID in image generation. We conducted an experiment comparing generators in the EDM2 environment using version S without CFG, applying the selection of time steps, specific to this model. The results are shown in Table 1. The flow version of HeunUDS (12) yields the same results as the original HeunEDM2, suggesting the interchangeability of the "dual" equations (9) and (10). The MSE diffusion performs nearly like a flow, confirming that β is a reliable predictor of generation performance. We also confirmed this result for a different architecture, JiT-B/16 trained for 600 epochs; see Li & He (2025), Table 6.

MSE-diffusion as a generator benchmark. The left panel of Figure 1 shows the diffusion size of Xie's models: SiT, REPA, REPA-E, and RAE. Although the noise schedules and parameterizations are identical, their underlying generators differ substantially. The generator of the earliest model, SiT, has a diffusion size close to ML(*Logis*) (blue). The REPA and REPA-E generator is defined by $\lambda'_t := 1/(2(1-t))$ (black), whereas the most recent model, RAE, generates a deterministic flow. Each successive generator moves closer to MSE-diffusion (orange)—the generator coherent with MSE training. Although these models differ in architecture and training objectives, this trend suggests that the diffusion size used by the generator should be close to the diffusion size induced by training. A similar evolution can be observed from EDM to EDM2 and from Stable Diffusion 1 to Stable Diffusion 3. Together with the experimental result, this suggests that the diffusion size provides a useful operational characteristic for comparing models.

Conclusion. Our findings reinforce previous results on the connection between training and generation. The MSE diffusion and the diffusion size, viewed as an operating characteristic, provide a basis for comparing diffusions and flows. In image synthesis, empirical evidence for the dominance of flows is now widespread, so our theoretical support may not substantially affect current image synthesis practice. Nevertheless, it helps explain why flows outperform diffusions. In image restoration, diffusions and flows currently appear more evenly matched (Zhou et al., 2024; Zheng et al., 2025b; Sobieski et al., 2025), and our work may therefore have practical implications in this setting.

AI USE STATEMENT

Generative AI was used to assist with the technical conversion of the manuscript's LaTeX source to the ICLR 2027 format and with language and stylistic editing. The authors reviewed and verified the resulting text and source.

REFERENCES

Michael S. Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *International Conference on Learning Representations*, 2023.

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

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

Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, et al. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *International Conference on Learning Representations*, 2024b.

Qinpeng Cui, Xinyi Zhang, Qiqi Bao, and Qingmin Liao. Elucidating the solution space of extended reverse-time sde for diffusion models. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *Advances in neural information processing systems*, 2021.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Patrick Esser, Suneet Kulal, Andreas Blattmann, Rameen Entezari, Jürgen Müller, Himanshu Saini, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning*, 2024.

Martin Gonzalez, Nelson Fernandez, Thuy Vinh Dinh Tran, Elies Gherbi, Hatem Hajri, and Nader Masmoudi. Seeds: Exponential sde solvers for fast high-quality sampling from diffusion models. In *Advances in Neural Information Processing Systems*, 2023.

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

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020.

Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In *International Conference on Machine Learning*, 2023.

Emiel Hoogeboom, Thomas Mensink, Jonathan Heek, Kay Lamerigts, Ruiqi Gao, and Tim Salimans. Simpler diffusion: 1.5 fid on imagenet512 with pixel-space diffusion. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025.

Chin-Wei Huang, Jae Hyun Lim, and Aaron C Courville. A variational perspective on diffusion-based generative models and score matching. In *Advances in Neural Information Processing Systems*, 2021.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, 2022.

Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a diffusion model with a bad version of itself. In *Advances in Neural Information Processing Systems*, 2024a.

Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 24174–24184, 2024b.

Diederik Kingma and Ruiqi Gao. Understanding diffusion objectives as the elbo with simple data augmentation. In *Advances in Neural Information Processing Systems*, 2023.

Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. In *Advances in Neural Information Processing Systems*, 2021.

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

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. *arXiv preprint arXiv:2511.13720*, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023.

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. In *Advances in Neural Information Processing Systems*, 2023.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *Machine Intelligence Research*, 22: 730–751, 2025.

Nanye Ma, Mark Goldstein, Michael S. Albergo, Nicholas M. Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, 2024.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, 2021.

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

Robin Rombach, Andreas Blattmann, Dan Lorenz, Patrick Esser, and Bjorn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2022.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015.

Arman Sabour, Sanja Fidler, and Karsten Kreis. Align your steps: Optimizing sampling schedules in diffusion models. In *International Conference on Machine Learning*, 2024.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2021.

Andrea Schioppa and Tim Salimans. Covariance-aware sampling for diffusion models. *arXiv preprint arXiv:2605.13910*, 2026.

Bartlomiej Sobieski, Matthew Tivnan, Yuang Wang, Siyeop Yoon, Pengfei Jin, Dufan Wu, Quanzheng Li, and Przemyslaw Biecek. System-embedded diffusion bridge models. In *Advances in Neural Information Processing Systems*, 2025.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in neural information processing systems*, 2019.

Yang Song, Chris Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. In *Advances in Neural Information Processing Systems*, 2021b.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021c.

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

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

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *International Conference on Learning Representations*, 2025.

Qinsheng Zhang and Yongxin Chen. Diffusion normalizing flow. In *Advances in Neural Information Processing Systems*, 2021.

Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. In *International Conference on Learning Representations*, 2023.

Qinsheng Zhang, Molei Tao, and Yongxin Chen. gddim: Generalized denoising diffusion implicit models. In *International Conference on Learning Representations*, 2023.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. *arXiv preprint arXiv:2510.11690*, 2025a.

Kaiwen Zheng, Guande He, Jianfei Chen, Fan Bao, and Jun Zhu. Diffusion bridge implicit models. In *International Conference on Learning Representations*, 2025b.

Linqi Zhou, Aaron Lou, Samar Khanna, and Stefano Ermon. Denoising diffusion bridge models. In *International Conference on Learning Representations*, 2024.

A NOISE SCHEDULES AND PARAMETERIZATIONS

Noise schedules. Kingma & Gao (2023) demonstrated that three popular noise schedules $\mathring{\lambda}_t$ can be derived uniformly as quantile functions of bell-shaped densities: normal, logistic, and hyperbolic secant.

Normal. $\mathring{\lambda}_t := \Phi^{-1}(t) + 0.4,\ \alpha_t := 1,\ \sigma_t = \exp(-\mathring{\lambda}_t)$, where Φ is the standard normal cumulative distribution function. This schedule is used in the EDM2 model (Karras et al., 2024b;a).

Logis. $\alpha_t := t,\ \sigma_t := 1 - t,\ \mathring{\lambda}_t = \log(t/(1-t))$, that is, the quantile of the (standard) logistic distribution. This schedule is inspired by optimal transport between the noise and data distributions (Lipman et al., 2023; Liu et al., 2023; Albergo & Vanden-Eijnden, 2023; Albergo et al., 2025; Ma et al., 2024; Esser et al., 2024; Yao et al., 2025).

Sech. $\alpha_t := \sin(\pi t/2),\ \sigma_t := \cos(\pi t/2),\ \mathring{\lambda}_t = \log(\tan(\pi t/2))$, so $(2/\pi)\,\mathring{\lambda}_t$ is the quantile function of the hyperbolic secant distribution (Nichol & Dhariwal, 2021; Dhariwal & Nichol, 2021; Kingma et al., 2021; Salimans & Ho, 2021; Hoogeboom et al., 2023; 2025).

SechInter. Hoogeboom et al. (2023; 2025) proposed the shifted Sech schedule interpolation, which can be expressed in a simplified form as $\mathring{\lambda}_t := \log(\tan(\pi t/2)) + (\log 16)(t-1),\ \alpha_t := (1 + \exp(-2\mathring{\lambda}_t))^{-1/2},\ \sigma_t^2 = 1 - \alpha_t^2$.

Parameterizations. Kingma & Gao (2023) showed that MSE training is determined by $\mathring{\lambda}_t$ and a weight $\omega_t = S_t^2/\mathring{\lambda}'_t$ equivalent to our S_t parametrization.

Noise. The network predicts ε, thus $S_t = 1$. This parameterization is very popular (Ho et al., 2020; Song et al., 2021a;c; Nichol & Dhariwal, 2021; Dhariwal & Nichol, 2021; Rombach et al., 2022; Peebles & Xie, 2023; Hoogeboom et al., 2023; Chen et al., 2024b;a).

Data. The network is trained to predict X, so $S_t = \mathring{\rho}_t^{-1}$. This natural schedule is not popular, but has recently gained importance in high-resolution diffusion models operating in pixel space (Hoogeboom et al., 2025; Li & He, 2025).

V-pred. The network predicts $V_t := \alpha_t\varepsilon - \sigma_t X$, thus $S_t = \alpha_t^{-1}$. This parametrization was employed by Salimans & Ho (2021) and Hoogeboom et al. (2023).

F-pred. The target for this prediction is $F_t := \sqrt{4 + \mathring{\rho}_t^2}\,X - \mathring{\rho}_t^2/\sqrt{4 + \mathring{\rho}_t^2}\,Y_t$, thus $S_t = \sqrt{4\mathring{\rho}_t^{-2} + 1}$. It is a schedule used in the EDM and EDM2 models (Karras et al., 2022; 2024b;a).

Vel. The target is the velocity of $\tilde{Z}_t$, that is, $v_t := \alpha'_t X + \sigma'_t\varepsilon = \alpha'_t\tilde{Y}_t - \mathring{\lambda}'_t\sigma_t\varepsilon$. Then (2) yields $\hat{v}_t(\alpha_t\tilde{Y}_t) = \alpha_t\tilde{Y}_t - \mathring{\lambda}'_t\sigma_t\hat{\varepsilon}_t(\tilde{Y}_t)$, so $S_t = \mathring{\lambda}'_t\sigma_t$. This very popular parameterization is motivated by considerations from optimal transport theory and is closely related to the logistic noise schedule (Lipman et al., 2023; Liu et al., 2023; Albergo & Vanden-Eijnden, 2023; Albergo et al., 2025; Ma et al., 2024; Yu et al., 2025; Leng et al., 2025; Zheng et al., 2025a).

VelLN. $S_t := \mathring{\lambda}'_t\sigma_t\,\exp\!\big(-\frac{1}{4}\big(\log\frac{t}{1-t}\big)^2\big)/\sqrt{\sqrt{2\pi}\frac{t}{1-t}}$. This parameterization results from a velocity-type prediction that employs a logit-normal distribution for time sampling (Esser et al., 2024; Yao et al., 2025; Li & He, 2025).

Sigmoid. $S_t := \sqrt{2\mathring{\lambda}'_t/\big(1 + \exp(2\mathring{\lambda}_t + 3)\big)}$. This parameterization was introduced in Kingma & Gao (2023) and is employed in the SiD2 model (Hoogeboom et al., 2025).

B PROOFS

Proof of Proposition 3.1.

The construction is standard: it starts from Brownian motion and then proceeds analogously to the proof of Picard's theorem. We give only a proof sketch, indicating how the final formula arises.

By iterating the recursion (6), for any grid points $t_i < t_j$ we obtain

$$r_{t_j}\hat{Y}_{t_j} = r_{t_i}\hat{Y}_{t_i} + (r_{t_{i+1}} - r_{t_i})\hat{X}_{t_i} + \ldots + (r_{t_j} - r_{t_{j-1}})\hat{X}_{t_{j-1}} + \sqrt{\rho_{t_j}^2 - \rho_{t_i}^2}\,\xi_{t_i},$$

where $\hat{Y}_{t_i}$ and ξ_{t_i} are independent, and

$$\xi_{t_i} := \frac{\sqrt{\rho_{t_{i+1}}^2 - \rho_{t_i}^2}\,\xi_{t_i}^N + \ldots + \sqrt{\rho_{t_j}^2 - \rho_{t_{j-1}}^2}\,\xi_{t_{j-1}}^N}{\sqrt{\rho_{t_j}^2 - \rho_{t_i}^2}} \sim \mathcal{N}(0, I_d).$$

As $N \to \infty$,

$$\sum_{k=i}^{j-1}(r_{t_{k+1}} - r_{t_k})\hat{X}_{t_k} \to \int_s^t \hat{X}_\tau(\hat{Y}_\tau)\,\mathrm{d}r(\tau) = \int_s^t \hat{X}_\tau(\hat{Y}_\tau)\,r'(\tau)\,\mathrm{d}\tau = \int_{r_s}^{r_t} \hat{X}_{\tau(r)}(\hat{Y}_{\tau(r)})\,\mathrm{d}r,$$

which yields (7). Dividing both sides of (7) by r_t and rewriting the integral in terms of $q_{st} := r_s/r_t$ gives (8) with the stated coefficients q_{st} and $\hat{\beta}_{st}$.

From the generator (6) *to SDE and back.*

The equation (6) for the simplest generator is equivalent to

$$\hat{Y}_t - \hat{Y}_s = \Big(\frac{r_s - r_t}{\Delta t\; r_t}\hat{Y}_s + \frac{r_t - r_s}{\Delta t\; r_t}\hat{X}_s\Big)\Delta t + \mathring{\rho}_t^{-1}\sqrt{\frac{\rho_t^2 - \rho_s^2}{\Delta t\;\rho_t^2}}\sqrt{\Delta t}\;\xi_s^N. \tag{20}$$

A direct manipulation with Taylor expansions yields, for $s = t - \Delta t$ and $|\delta_1|, |\delta_2| = o(1)$,

$$\frac{r_t - r_s}{\Delta t\, r_t} = (\log r_s)'(1 + \delta_1) \quad \text{and} \quad \frac{\rho_t + \rho_s}{\rho_t}\,\frac{\rho_t - \rho_s}{\Delta t\,\rho_t} = 2(\log \rho_s)'(1 + \delta_2).$$

Hence (20) takes the form

$$\hat{Y}_t - \hat{Y}_s = (\log r_s)'\big(\hat{X}_s - \hat{Y}_s\big)\,\Delta t + \mathring{\rho}_s^{-1}\sqrt{2(\log \rho_s)'}\sqrt{\Delta t}\;\xi_s^N + \mathcal{O}_P(\Delta t). \tag{21}$$

Let W_t be a standard d-dimensional Wiener process and assume that $\hat{X}_t$ is sufficiently regular. Then the difference equation (21) converges to the Itô SDE

$$\begin{aligned}\mathrm{d}\hat{Y}_t &= (\log r_t)'\big(\hat{X}_t - \hat{Y}_t\big)\mathrm{d}t + \mathring{\rho}_t^{-1}\sqrt{2(\log \rho_t)'}\;\mathrm{d}W_t, \\ &= (\lambda_t' + \mathring{\lambda}_t')\big(\hat{X}_t - \hat{Y}_t\big)\mathrm{d}t + \mathring{\rho}_t^{-1}\sqrt{2\lambda_t'}\;\mathrm{d}W_t, \\ &= \mathring{\rho}_t^{-1}(\lambda_t' + \mathring{\lambda}_t')\hat{\varepsilon}_t\mathrm{d}t + \mathring{\rho}_t^{-1}\sqrt{2\lambda_t'}\;\mathrm{d}W_t.\end{aligned} \tag{22}$$

Fix $s < t$. By substituting $\tilde{Y}_t = (r_t/r_s)\hat{Y}_t$ for $\hat{Y}_t$ in (22) we get

$$r_s\mathrm{d}\tilde{Y}_t = \; r_t'\hat{X}_t\,\mathrm{d}t \;+\; \rho_t\sqrt{2(\log \rho_t)'}\;\mathrm{d}W_t. \tag{23}$$

By integrating (23) and returning to $\hat{Y}_t$, we obtain (7), which, with the simplest discretization, takes the form of (6).

Note that the substitution leading to formula (23) uses the method of variation of constants—the same approach is used to derive DPM solvers. Thanks to the construction of the diffusion process by (7), SDEs are not needed.

Proof of Proposition 4.1.

By the chain rule for KL along the grid $t_0 < t_1 < \cdots < t_N = t$ we obtain

$$\begin{aligned}&\mathbb{D}\big[p_{t_1:t}^N(.|z_{t_0}, x)\,\big\|\,\hat{p}_{t_1:t}^N(.|z_{t_0})\big] = \mathbb{D}\big[p_{t_1}(.|z_{t_0}, x)\,\big\|\,\hat{p}_{t_1}(.|z_{t_0})\big] + \mathbb{E}_{Z_{t_1}}\,\mathbb{D}\big[p_{t_2}(.|Z_{t_1}, x)\,\big\|\,\hat{p}_{t_2}(.|Z_{t_1})\big] \\ &+ \cdots + \mathbb{E}_{Z_{t_{N-1}}}\,\mathbb{D}\big[p_t(.|Z_{t_{N-1}}, x)\,\big\|\,\hat{p}_t(.|Z_{t_{N-1}})\big].\end{aligned} \tag{24}$$

In our setting, from (14) this can be rewritten in terms of denoising errors with weights $w_{t_j}^N$:

$$(24) = w_{t_0}^N\;\|\hat{\varepsilon}_{t_0} - \varepsilon_{t_0}\|^2 + w_{t_1}^N\,\mathbb{E}_{\varepsilon_{t_1}}\big\|\hat{\varepsilon}_{t_1} - \varepsilon_{t_1}\big\|^2 + \cdots + w_{t_{N-1}}^N\,\mathbb{E}_{\varepsilon_{t_{N-1}}}\big\|\hat{\varepsilon}_{t_{N-1}} - \varepsilon_{t_{N-1}}\big\|^2.$$

So

$$\mathbb{E}_{Z_{t_0},X}\mathbb{D}\big[p^N_{t_1:t}(.|Z_{t_0},X)\,\big\|\,\hat{p}^N_{t_1:t}(.|Z_{t_0})\big]=\sum_{i=0}^{N-1}\mathbb{E}_{\varepsilon_{t_i},X}w^N_{t_i}\left\|\hat{\varepsilon}_{t_i}-\varepsilon_{t_i}\right\|^2=\sum_{i=0}^{N-1}\mathbb{E}_{\varepsilon,X}w^N_{t_i}\left\|\hat{\varepsilon}_{t_i}-\varepsilon\right\|^2.$$

Proof of Proposition 4.2.

A direct manipulation with Taylor expansions yields, for $s=t-\Delta\tau$,

$$\begin{aligned}\bar{w}^N_{t-\Delta\tau}:=\frac{w^N_{t-\Delta\tau}}{\Delta\tau}&=\left(\frac{\mathring{\rho}_t-\mathring{\rho}_s}{\Delta\tau\,\mathring{\rho}_s}+\frac{\rho_t-\rho_s}{\Delta\tau\,\rho_t}\right)^2\Bigg/\left(2\,\frac{\rho_t-\rho_s}{\Delta\tau\,\rho_t}\,\frac{\rho_t+\rho_s}{\rho_t}\right)\\&=\big(\mathring{\lambda}'_t\,(1+\delta_1)+\lambda'_t\,(1+\delta_2)\big)^2/\big(4\,\lambda'_t\,(1+\delta_3)\big)=\bar{w}_t+o(1),\end{aligned}$$

where $\bar{w}_t:=\big(\lambda'_t+\mathring{\lambda}'_t\big)^2/(4\,\lambda'_t)$ and $|\delta_1|,|\delta_2|,|\delta_3|=o(1)$.

For $\tau\in[t_0,t]$ define $t^N(\tau):=\max\{t_i:t_i\le\tau\}$ and $m_\tau:=\mathbb{E}_{\varepsilon,X}\big\|\hat{\varepsilon}_\tau-\varepsilon\big\|^2$. From Proposition 4.1 we obtain

$$\begin{aligned}\hat{\mathcal{L}}^N(t_0,t,\mathring{\lambda},\lambda)&=\sum_{i=0}^{N-1}\frac{w^N_{t_i}}{\Delta\tau}m_{t_i}\Delta\tau=\int_{t_0}^{t}\bar{w}^N_{t^N(\tau)}m_{t^N(\tau)}\,d\tau=\int_{t_0}^{t}\bar{w}_\tau(1+o(1))m_\tau(1+o(1))\,d\tau\\&=\int_{t_0}^{t}\bar{w}_\tau m_\tau\,d\tau+o(1).\end{aligned}$$

Proof of Proposition 5.1.

Let us fix t and simplify notation $b:=\lambda'_t, b_c:=\lambda'_{c,t}, \mathring{b}:=\mathring{\lambda}'_t, s:=\sqrt{M}S_t$. Therefore, from the KL-MSE coherence condition (17)

$$(b+\mathring{b})^2/(4\,b)=s^2. \tag{25}$$

The definition of the constant M implies that $s^2\ge\mathring{b}$, thus equation (25) has 2 roots

$$b_-=\left(s-\sqrt{s^2-\mathring{b}}\right)^2\quad\text{and}\quad b_+=\left(s+\sqrt{s^2-\mathring{b}}\right)^2.$$

Observe that $b_-b_+=\mathring{b}^2$, so $b_-\le\mathring{b}\le b_+$. By comparison with (16) $b_-=b_{c_-}, b_+=b_{c_+}$, we obtain formulas for the penalty constants $c_-=(\mathring{b}/b_-)^2-1,\ \ c_+=(\mathring{b}/b_+)^2-1$ and $c_+\le 0\le c_-$. Hence only b_- optimizes the penalized maximum likelihood objective. Sufficiency is obvious.

C Additional Discussion and Comparisons

We see no significant difference between score-based frameworks (Song & Ermon, 2019; Song et al., 2021c;b; Huang et al., 2021; Karras et al., 2022; 2024b; Lu et al., 2023; 2025; Gonzalez et al., 2023) and stochastic interpolants (Kingma et al., 2021; Kingma & Gao, 2023; Albergo & Vanden-Eijnden, 2023; Albergo et al., 2025; Lipman et al., 2023; Liu et al., 2023; Hoogeboom et al., 2023; 2025) that generate processes on $(0,1)$. It is crucial that $\mathring{\rho}_t$ and ρ_t maintain strictly positive values within the closed interval of actual generation. This positivity condition is essential for ensuring the realism of the analysis, a requirement clearly demonstrated in the proofs of global convergence for numerical ODE solvers (Lu et al., 2023; 2025; Gonzalez et al., 2023).

Cui et al. (2025) provide diffusion-state representations as general as our Proposition 3.1. However, their formulas contain weights under the integral and are defined implicitly through a function ϕ. As a result, their solvers require an additional numerical integration subroutine. In our representation, the integrand is just the denoiser, which, beyond its computational advantages, yields a simple interpretation of the coefficient β as the diffusion size.

Formula (15) is equivalent to the KLUB expression introduced in equation (5) of Sabour et al. (2024) and to the average ELBO expression in equation (51) of Huang et al. (2021); the latter note in Footnote 9 that their λ may be time-dependent. Unlike these works, however, our proof does not rely

on stochastic calculus. Despite this generalization, our construction of the continuous-time diffusion loss is considerably simpler than previous ones.

Our modeling choice consists in selecting the smaller root of equation (17). We make this choice because we want the diffusion induced by training to admit a penalized maximum-likelihood interpretation. Moreover, to the best of our knowledge, image generation has not previously been performed using a diffusion whose noise is substantially larger than that of the corresponding reversible diffusion. Nevertheless, the diffusion determined by the larger root of equation (17) has the same marginal distributions and the same diffusion loss.

Implementing diffusion models using α_t and σ_t has become common practice, despite the largely simulation-based arguments of Karras et al. (2022), suggesting that α_t is unnecessary. We formalize these arguments as follows: α_t serves merely as input normalization within the denoiser, which is redundant for the generation process. Moreover, since the loss function depends on the diffusion process solely through $\mathring{\lambda}_t$ and λ'_t, one can—in the context of maximum likelihood analysis—consider the processes Y_t and $\hat{Y}_t$ without loss of generality.

Based on a recent paper by the Google team working on SiD2 (Schioppa & Salimans, 2026), we can conclude that SiD2 is not a SOTA model in the practically relevant regime considered here, where the number of function evaluations is typically 50–100, rather than 500–1000. Figure 1 of that paper shows that, on ImageNet 512×512, the new SiD2 generators achieve $FID > 2$ at approximately 100 NFE, whereas RAE and EDM2 achieve $FID < 1.25$.

D SENSITIVITY ANALYSIS

In Appendix A, we present several popular noise schedules and parameterizations. We also compared them graphically, together with the corresponding MSE-diffusion sizes for a fixed diffusion size under different parameterizations. In particular, we found that:

- the *Logis* and *Sech* noise schedules are almost indistinguishable;
- the noise schedule used in SiD2, namely *SechInter*, is shifted considerably upward relative to the other schedules;
- the MSE-diffusion sizes corresponding to the *V-pred*, *F-pred*, and *Vel* parameterizations are very similar.

We find it particularly interesting that, for **current** SOTA models, the MSE-diffusions induced by training are closer to the diffusions used by the generators than they were in **earlier** SOTA models. Thus, as the technology develops, the training and generation processes appear to become increasingly aligned. This pattern can be observed in the following model families: (SiT, RAE), (EDM, EDM2), and (Stable Diffusion 1, Stable Diffusion 3).