COAL: JACOBIAN-FREE LIKELIHOOD EVALUATION FOR RECTIFIED FLOWS

Anonymous authors
Paper under double-blind review

ABSTRACT

Flow-based generative models achieve strong sample quality, but per-sample likelihood evaluation under the instantaneous change-of-variables formula remains bottlenecked by an $\mathcal{O}(d^3)$ divergence integral. Hutchinson's stochastic trace estimator partially mitigates this, at the cost of K backpropagations through the velocity network per ODE step and a noisy integrand. We show that the divergence of a Rectified Flow velocity admits a deterministic closed form: by Tweedie's formula, it equals the trace of the conditional covariance of the velocity residual, scaled by a known function of t. We amortize this trace with a lightweight per-coordinate variance head trained jointly with the velocity, yielding COAL (covariance-amortized likelihood): a deterministic per-sample likelihood estimator that runs in a single forward pass per ODE step, with no input-space backpropagation and no stochastic probes. On class-conditional ImageNet-256 with a DiT-XL latent rectified flow, COAL matches the Hutchinson reference at $3.5\times$ less evaluation compute.

1 INTRODUCTION

Likelihood evaluation underpins a broad range of statistical modeling and machine learning workflows (Müller et al., 2019; Rehman et al., 2025b; Nalisnick et al., 2019; Rafailov et al., 2023; Schulman et al., 2017; Shao et al., 2024). While likelihood comes for free with autoregressive language models, the dominant approaches for continuous data like images are ODE-based Rectified Flows (Liu et al., 2023; Lipman et al., 2023; Song et al., 2021b;a), which are powerful generators but not directly likelihood-based, limiting any downstream use that depends on reliable likelihood evaluation.

The de facto likelihood estimator for flow models is the instantaneous change-of-variables (ICOV) formula (Chen et al., 2018; Grathwohl et al., 2019), which integrates the divergence of the velocity field along the sample trajectory:

$$\log \pi_1(\boldsymbol{z}_1) = \log \pi_0(\boldsymbol{z}_0) - \int_0^1 \nabla \cdot v_t(\boldsymbol{z}_t)\, \mathrm{d}t, \tag{1}$$

where $\{\boldsymbol{z}_t\}_{t=0}^1$ is obtained by solving $\mathrm{d}\boldsymbol{z}_t/\mathrm{d}t = v_t(\boldsymbol{z}_t)$ backward in time from the query $\boldsymbol{z}_1$, $v : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ is the velocity field, and $\nabla\cdot$ denotes the divergence operator. Though elegant and model-agnostic, exact divergence costs $\mathcal{O}(d^3)$, prohibitive when d reaches the thousands for images. Hutchinson estimators approximate it with K vector-Jacobian products $\xi^\top J_{v_t}\xi$, but introduce two costs: (i) K backpropagations through the velocity network per ODE step, with heavy compute and memory overhead; and (ii) a stochastic integrand with non-vanishing per-step variance. Recent alternatives (Ai et al., 2025; Rehman et al., 2025a) either retain Hutchinson estimation during training or still require $\mathcal{O}(d^3)$ determinant evaluation at inference and both restrict to *flow map models* only.

In this work, we show that the divergence of a Rectified Flow velocity admits a *deterministic, closed-form* expression that requires no backpropagation through the network at inference time. The key observation is that a Rectified Flow velocity is highly structured: its marginals are Gaussian convolutions of the data distribution under a prescribed schedule, so its divergence need not be computed generically. By Tweedie's formula, the divergence of the optimal velocity then reduces to computing the trace of the conditional covariance of the velocity residual. We then amortize the cost of this covariance estimation with a lightweight diagonal-variance head that predicts a per-coordinate variance whose sum recovers the required trace; the head is trained jointly with

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.

the velocity under a per-coordinate Gaussian negative log-likelihood, requiring neither Jacobian computation nor Hutchinson probes. We call the resulting estimator COAL (**Co**variance-**A**mortized **L**ikelihood).

This reformulation has two concrete consequences. First, asymptotic per-step cost matches the velocity network itself, $\mathcal{O}(d^\omega)$ rather than the $\mathcal{O}(d^{\omega+1})$ of evaluating the exact divergence, and evaluation avoids input-space backpropagation entirely. Second, the estimator is deterministic and single-pass: the likelihood integral is accumulated as a scalar auxiliary state alongside the ODE solve, with no Monte Carlo probes. To the best of our knowledge, COAL is the first per-sample likelihood estimator for general Rectified Flow models that requires no Jacobian computation or Hutchinson probes. We evaluate COAL on DiT-XL latent Rectified Flow models trained on ImageNet-256 (d=4096). COAL Pareto-dominates Hutchinson on the compute-versus-accuracy frontier, reaching matched per-sample BPD-RMSE at roughly $3.5\times$ less compute.

2 BACKGROUND

Rectified Flow (RF) Let π_1 be the data distribution on $\mathbb{R}^d$ and $\pi_0 = \mathcal{N}(0, \mathbf{I})$ the standard Gaussian source. RF (Liu et al., 2023; Lipman et al., 2023; Albergo & Vanden-Eijnden, 2023) learns a velocity field $v : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ such that the ODE

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{Z}_t = v_t(\boldsymbol{Z}_t), \quad \boldsymbol{Z}_0 \sim \pi_0,$$

transports π_0 to π_1 over $t \in [0,1]$. In practice v is parametrized by a neural network v_θ (e.g., a Diffusion Transformer (Peebles & Xie, 2023; Ma et al., 2024)) and trained on the stochastic interpolant $\boldsymbol{X}_t = t\boldsymbol{X}_1 + (1-t)\boldsymbol{X}_0$ with the MSE objective

$$\mathcal{L}(\theta) \coloneqq \mathbb{E}\left[\|v_\theta(\boldsymbol{X}_t, t) - (\boldsymbol{X}_1 - \boldsymbol{X}_0)\|^2\right], \quad (\boldsymbol{X}_0, \boldsymbol{X}_1) \sim \pi_0 \times \pi_1,\ t \sim \mathrm{Unif}(0,1), \tag{2}$$

whose minimizer is $v_t^*(\boldsymbol{x}) = \mathbb{E}[\boldsymbol{X}_1 - \boldsymbol{X}_0 \mid \boldsymbol{X}_t = \boldsymbol{x}]$; under v^*, the marginals of $\boldsymbol{Z}_t$ match those of $\boldsymbol{X}_t$. Since $\boldsymbol{X}_0 \sim \mathcal{N}(0, \mathbf{I})$, we have $\boldsymbol{X}_t \mid \boldsymbol{X}_1 \sim \mathcal{N}(t\boldsymbol{X}_1, (1-t)^2\mathbf{I})$, and the marginal of $\boldsymbol{X}_t$ is a Gaussian convolution of the data:

$$\mathrm{Law}(\boldsymbol{X}_t) \triangleq \pi_t(\boldsymbol{x}) = \int \frac{1}{(1-t)^d}\,\rho\left(\frac{\boldsymbol{x} - t\boldsymbol{x}_1}{1-t}\right)\pi_1(\mathrm{d}\boldsymbol{x}_1) = \mathrm{Law}(\boldsymbol{Z}_t),$$

where ρ is the standard Gaussian density.

Likelihood Evaluation Evaluating (1) requires computing $\nabla \cdot v_t$ at each ODE step, which costs $\mathcal{O}(d^3)$ for Transformer-based flows (d backward passes of cost $\mathcal{O}(d^2)$ each to assemble J_{v_t}), impractical even for moderate dimensions $d \gtrsim 4096$. The standard workaround is the Hutchinson trace estimator: for $J_{v_t}(\boldsymbol{x}) \coloneqq \nabla_{\boldsymbol{x}} v_t(\boldsymbol{x})$,

$$\nabla \cdot v_t(\boldsymbol{x}) = \mathrm{tr}(J_{v_t}(\boldsymbol{x})) = \mathbb{E}_\xi\left[\xi^\top J_{v_t}(\boldsymbol{x})\,\xi\right] \approx \frac{1}{K}\sum_{k=1}^{K} \xi_k^\top J_{v_t}(\boldsymbol{x})\,\xi_k,$$

where $\xi \sim \mathrm{Rademacher}$ ($\xi_i \in \{\pm 1\}$). This reduces per-step cost but retains two fundamental limitations: K vector–Jacobian products per ODE step (one backpropagation each), and a stochastic integrand whose per-step variance does not vanish with finer integration steps.

3 COAL: COVARIANCE-AMORTIZED LIKELIHOOD

Computing the divergence generically treats the velocity field as a black box. Yet *a Rectified Flow velocity is far from arbitrary: its marginals are Gaussian convolutions of the data distribution.* By Tweedie's formula, this structure collapses the divergence to a closed-form scalar: the conditional covariance trace of $\boldsymbol{X}_1 - \boldsymbol{X}_0$, requiring neither a Jacobian nor a Monte Carlo probe. COAL exploits this structure by amortizing the trace into a lightweight auxiliary output during training, so that likelihood evaluation at inference matches the backbone's own $\mathcal{O}(d^\omega)$ forward cost rather than the $\mathcal{O}(d^{\omega+1})$ of generic divergence computation.

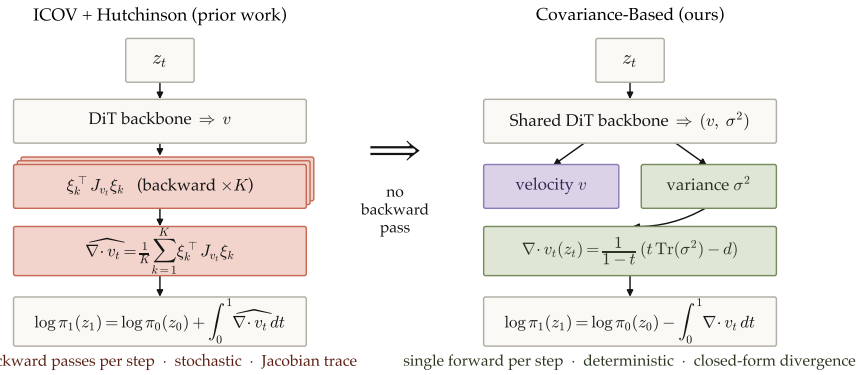


Figure 1: ICOV with Hutchinson (left) requires K backward passes per ODE step and yields a stochastic divergence estimate. COAL (right) reuses the flow backbone with a lightweight variance head, turning the divergence into a closed-form scalar of σ_θ^2 and integrating the likelihood jointly with the state, using only one forward evaluation of the network at every ODE step.

3.1 DIVERGENCE AS CONDITIONAL COVARIANCE

Concretely, at every $t \in [0, 1)$ the divergence of v^* is a known affine function of the trace of the conditional covariance of the velocity residual:

Proposition 3.1 (Divergence as conditional covariance)**.** *Consider the Rectified Flow stochastic interpolant $\boldsymbol{X}_t = (1-t)\boldsymbol{X}_0 + t\boldsymbol{X}_1$ with $\boldsymbol{X}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\boldsymbol{X}_0 \perp \boldsymbol{X}_1$, and assume the marginal density π_t is strictly positive and twice continuously differentiable on $\mathbb{R}^d$ for every $t \in [0, 1)$. Then the spatial divergence of the optimal velocity field $v_t^*(\boldsymbol{x}) \coloneqq \mathbb{E}[\boldsymbol{X}_1 - \boldsymbol{X}_0 \mid \boldsymbol{X}_t = \boldsymbol{x}]$ satisfies*

$$\nabla \cdot v_t^*(\boldsymbol{x}) \;=\; \frac{1}{1-t}\left(t \operatorname{Tr}\left(\operatorname{Cov}\left(\boldsymbol{X}_1 - \boldsymbol{X}_0 \mid \boldsymbol{X}_t = \boldsymbol{x}\right)\right) - d\right), \qquad t \in [0, 1). \tag{3}$$

Moreover, at any boundary point $\boldsymbol{x}$ with $\pi_1(\boldsymbol{x}) > 0$ where the posterior moments admit the first-order expansion in Appendix A.1, the right-hand side extends continuously as $t \uparrow 1$ and its limit equals $\nabla \cdot v_1^*(\boldsymbol{x}) = d$.

The proof applies the Tweedie-Miyasawa formula to express the score $\nabla \log \pi_t(\boldsymbol{x})$ in terms of $\mathbb{E}[\boldsymbol{X}_1 \mid \boldsymbol{X}_t = \boldsymbol{x}]$, differentiates once more to obtain the trace of the conditional covariance, and evaluates the terminal limit by L'Hôpital's rule; the full derivation is provided in Appendix A.1. This result shows that the instantaneous expansion or contraction of the probability flow is governed entirely by the model's total internal uncertainty, i.e., the trace of the conditional covariance of the velocity target. Indeed, the trace of this covariance equals the conditional residual of the Rectified Flow objective in (2):

$$\operatorname{Tr}\left(\operatorname{Cov}\left(\boldsymbol{X}_1 - \boldsymbol{X}_0 \mid \boldsymbol{X}_t = \boldsymbol{x}\right)\right) \;=\; \mathbb{E}\left[\left\|\boldsymbol{X}_1 - \boldsymbol{X}_0 - v_t^*(\boldsymbol{X}_t)\right\|^2 \,\middle|\, \boldsymbol{X}_t = \boldsymbol{x}\right].$$

3.2 AMORTIZING THE COVARIANCE TRACE VIA A VARIANCE HEAD

While Eq. (3) provides an elegant formulation, evaluating the conditional covariance directly at inference time is not computationally feasible. Therefore, we extend the network with a lightweight variance head $\sigma_\theta^2(\boldsymbol{x}, t) \in \mathbb{R}$ that predicts the trace of the covariance directly. The velocity v_θ and the variance head σ_θ^2 are trained jointly with the following combined objective:

$$\mathcal{L}(\theta) \coloneqq \mathbb{E}\left[\frac{\left\|\boldsymbol{X}_1 - \boldsymbol{X}_0 - v_\theta(\boldsymbol{X}_t, t)\right\|^2}{\sigma_\theta^2(\boldsymbol{X}_t, t)} + \log \sigma_\theta^2(\boldsymbol{X}_t, t)\right], \tag{4}$$

This objective avoids Monte Carlo estimation of the covariance at each training step: by the law of total expectation, the conditional variance reduces to an unconditional expectation that is directly optimizable. The following lemma certifies that its minimizer recovers the desired trace.

Lemma 3.2 (Joint minimizer)**.** *Assume the function classes for v_θ and σ^2_θ are sufficiently rich to represent any measurable vector- and positive scalar-valued function of $(\boldsymbol{x}, t)$. Then (v_*, σ^2_*) is a joint minimizer of* (4) *if and only if*

$$v_*(\boldsymbol{x}, t) = \mathbb{E}\left[\boldsymbol{X}_1 - \boldsymbol{X}_0 \mid \boldsymbol{X}_t = \boldsymbol{x}\right], \qquad \sigma^2_*(\boldsymbol{x}, t) = \mathrm{Tr}\left(\mathrm{Cov}(\boldsymbol{X}_1 - \boldsymbol{X}_0 \mid \boldsymbol{X}_t = \boldsymbol{x})\right).$$

3.3 Tractable Likelihoods with the Amortized Trace

With the trained velocity v_θ and variance head σ^2_θ, we substitute the divergence in the ICOV formula (1) with our amortized estimator. The complete update for evaluating the log-likelihood of a target $\boldsymbol{x}^*$ becomes a joint ODE:

$$\begin{bmatrix} \boldsymbol{z}_{t_{i+1}} \\ \ell_{t_{i+1}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{z}_{t_i} \\ \ell_{t_i} \end{bmatrix} + (t_{i+1} - t_i) \begin{bmatrix} v_\theta(\boldsymbol{z}_{t_i}, t_i) \\ \frac{1}{1-t_i}\left(t_i \sigma^2_\theta(\boldsymbol{z}_{t_i}, t_i) - d\right) \end{bmatrix}, \tag{5}$$

where the trajectory $\{\boldsymbol{z}_t\}$ is driven by v_θ. This reduces likelihood computation from the $\mathcal{O}(d^{\omega+1})$ bottleneck of generic divergence evaluation to a single $\mathcal{O}(d^\omega)$ forward pass through v_θ and σ^2_θ, enabling scalable per-sample likelihood evaluation on modern high-dimensional flow architectures.

Remarks Related forms of the identity in Proposition 3.1 have appeared in disparate settings: information-theory studies (Guo et al., 2005; Hatsell & Nolte, 1971; Palomar & Verdú, 2006; Dytso et al., 2020), and, within the flow and diffusion literature, a small set of works that invoke conditional-covariance structure for purposes other than likelihood estimation (Meng et al., 2021; Kong et al., 2023; Rissanen et al., 2025; Shoushtari et al., 2025). None of these works, however, treat the conditional covariance as a quantity *learnable during training* like equation (4), nor close it into a per-sample likelihood estimator for the underlying flow; the field has consequently continued to rely on ICOV with the Hutchinson trace estimator for likelihood evaluation. Our recipe closes this loop: amortizing the covariance through the Gaussian-NLL objective (4), together with the boundary parametrization of Section 4.2 that renders the divergence integrable across $t = 1$, yielding a deterministic, forward-only per-sample likelihood estimator at inference. The linear schedule $\boldsymbol{X}_t = (1-t)\boldsymbol{X}_0 + t\boldsymbol{X}_1$ is chosen for notational simplicity; all results in this paper extend directly to any affine stochastic interpolant $\boldsymbol{X}_t = \alpha_t \boldsymbol{X}_1 + \beta_t \boldsymbol{X}_0$ with smooth α_t, β_t.

4 Practical Design of the Coal-DiT

The recipe of Sec. 3.1 is minimal in its mathematical content: train a scalar trace predictor with (4), plug it into (5), and likelihoods follow. Scaling it to a realistic regime, however, exposes a number of subtleties that the theory itself does not address. This section collects the design choices that we found to materially improve the likelihood method in practice, and explains the mathematical reason behind each.

4.1 Dense diagonal supervision instead of a single scalar

Equation (3) only requires the *trace* of the conditional covariance, so in principle a scalar head $\sigma^2_\theta(\boldsymbol{x}, t) \in \mathbb{R}$ suffices. In practice, regressing a single scalar onto the residual $\|\boldsymbol{X}_1 - \boldsymbol{X}_0 - v_\theta\|^2$ yields an extremely noisy signal: every minibatch element contributes only a single target, and any per-coordinate misprediction of the velocity is absorbed into the scalar's bias. Since the trace is a sum, predicting the full diagonal $\sigma^2_{\theta,i}(\boldsymbol{x}, t)$ provides d regression targets per sample at no additional inference cost; we use the scalar prediction $\sigma^2_\theta(\boldsymbol{x}, t) \coloneqq \sum_i \sigma^2_{\theta,i}(\boldsymbol{x}, t)$ in the divergence. We train each entry $\sigma^2_{\theta,i}$ to match the corresponding diagonal of the conditional covariance. The objective then becomes

$$\mathcal{L}(\theta) \coloneqq \mathbb{E}\left[\sum_{i=1}^{d} \frac{\left(\boldsymbol{X}_{1,i} - \boldsymbol{X}_{0,i} - v_{\theta,i}(\boldsymbol{X}_t, t)\right)^2}{\sigma^2_{\theta,i}(\boldsymbol{X}_t, t)} + \log \sigma^2_{\theta,i}(\boldsymbol{X}_t, t)\right], \tag{6}$$

which is the sum of d independent Gaussian NLLs. Lemma 3.2 extends verbatim to each coordinate, so the minimizer satisfies $\sigma^2_{*,i}(\boldsymbol{x}, t) = \mathrm{Var}(\boldsymbol{X}_{1,i} - \boldsymbol{X}_{0,i} \mid \boldsymbol{X}_t = \boldsymbol{x})$, and summing over i recovers exactly the trace needed in (3). The compute cost is nearly identical, and this parametrization further avoids the extra projection that would otherwise collapse the token sequence into a single scalar.

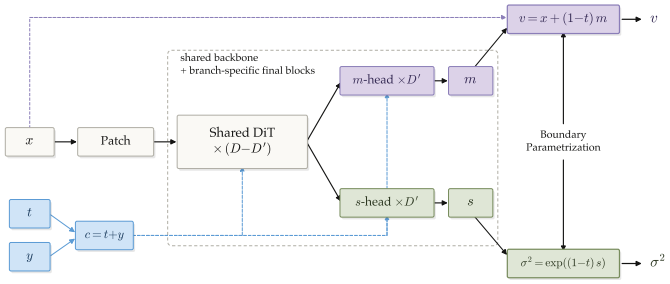


Figure 2: Architecture of the COAL-DiT. A shared DiT backbone of $D - D'$ layers splits in its final D' blocks into parallel velocity and variance heads. Both branches use *boundary parametrization*, $v_\theta = \boldsymbol{x} + (1-t)\,m_\theta$ and $\log \sigma^2_\theta = (1-t)\,s_\theta$, enforcing $v_\theta(\boldsymbol{x}, 1) = \boldsymbol{x}$ and $\sigma^2_\theta(\boldsymbol{x}, 1) = \mathbf{1}$ by construction at the terminal time.

Crucially, the supervision signal is now d times denser, providing substantially more learning signal per step at no additional cost, which we find essential for efficient training on high-dimensional latents.

4.2 BOUNDARY PARAMETRIZATION OF v AND σ^2

Despite the $1/(1-t)$ prefactor, theoretically (3) still admits a finite limit as $t \to 1$ for the optimal Rectified Flow, by virtue of two properties. The terminal conditions

$$\begin{aligned} v_*(\boldsymbol{x}, 1) &= \mathbb{E}[\boldsymbol{X}_1 - \boldsymbol{X}_0 \mid \boldsymbol{X}_1 = \boldsymbol{x}] = \boldsymbol{x} - \mathbb{E}[\boldsymbol{X}_0] = \boldsymbol{x}, \\ \mathrm{Cov}(\boldsymbol{X}_1 - \boldsymbol{X}_0 \mid \boldsymbol{X}_1 = \boldsymbol{x}) &= \mathrm{Cov}(\boldsymbol{X}_0 \mid \boldsymbol{X}_1 = \boldsymbol{x}) = \mathrm{Cov}(\boldsymbol{X}_0) = \mathbf{I}, \end{aligned} \tag{7}$$

follow directly from $\boldsymbol{X}_0 \perp \boldsymbol{X}_1$ and $\boldsymbol{X}_0 \sim \mathcal{N}(0, \mathbf{I})$. At $t = 1$, they drive the numerator $t\,\mathrm{Tr}(\mathrm{Cov}) - d$ of (3) to zero, leaving a $0/0$ indeterminate form. The analytic regularity of the Rectified Flow marginal near $t = 1$ then resolves this limit to a finite value via L'Hôpital's rule (Appendix A.3). A freely parametrized network enjoys no such cancellation. The Rectified Flow objective does not enforce (7), so a learned v_θ and σ^2_θ generically violate these conditions at $t = 1$. Any such residual mismatch is then divided by $(1-t)$ in (3), producing an $\mathcal{O}(1/(1-t))$ blow-up that dominates the terminal-time contribution to $\int_0^1 \nabla \cdot v_t \, \mathrm{d}t$.

To eliminate this blow-up, we bake (7) directly into the architecture, taking inspiration from Hu et al. (2025), so the cancellation holds by construction. The velocity adopts the standard boundary parametrization, and the variance head predicts a *log-variance* (ensuring positivity without clamping) under the matching residual factorization:

$$v_\theta(\boldsymbol{x}, t) = \boldsymbol{x} + (1-t)\,m_\theta(\boldsymbol{x}, t), \qquad \log \sigma^2_\theta(\boldsymbol{x}, t) = (1-t)\,s_\theta(\boldsymbol{x}, t). \tag{8}$$

Both expressions are forced to satisfy the terminal conditions $v_\theta(\boldsymbol{x}, 1) = \boldsymbol{x}$ and $\sigma^2_\theta(\boldsymbol{x}, 1) \equiv \mathbf{1}$ regardless of the residual networks m_θ and s_θ, recovering (7) by construction. With both conditions satisfied regardless of the learned parameters, the L'Hôpital argument of the preceding paragraph carries over to the learned model: its divergence stays finite at $t = 1$ whenever m_θ and s_θ are themselves regular (Appendix A.3). Architecturally, the COAL-DiT shares a backbone of $D - D'$ layers across the velocity and variance branches and splits only in the final D' transformer blocks, so the auxiliary output incurs only a small additional computational cost (see Fig. 2); we find $D' = 2$ sufficient for DiT-B and $D' = 4$ for DiT-XL.

Numerically stable divergence via `expm1`. Under (8), $\sigma^2_{\theta,i} = \exp((1-t)s_{\theta,i})$, so (3) reads $\left(t \sum_i \exp((1-t)s_i) - d\right)/(1-t)$. As $t \to 1$, both numerator and denominator are $\mathcal{O}(1-t)$, and a naive evaluation loses most significant digits to catastrophic cancellation. The library primitive $\mathrm{expm1}(u) \coloneqq \exp(u) - 1$ evaluates this difference accurately for small u. Writing $\exp(u) = 1 + u\,r(u)$ with $r(u) \coloneqq \mathrm{expm1}(u)/u$ and substituting $u = (1-t)s_{\theta,i}$, the $(1-t)$ factor cancels analytically and yields

$$\frac{1}{d} \nabla \cdot v_t(\boldsymbol{x}) = -1 + t \cdot \frac{1}{d} \sum_{i=1}^{d} s_{\theta,i}(\boldsymbol{x}, t) \cdot r((1-t)\, s_{\theta,i}(\boldsymbol{x}, t)). \tag{9}$$

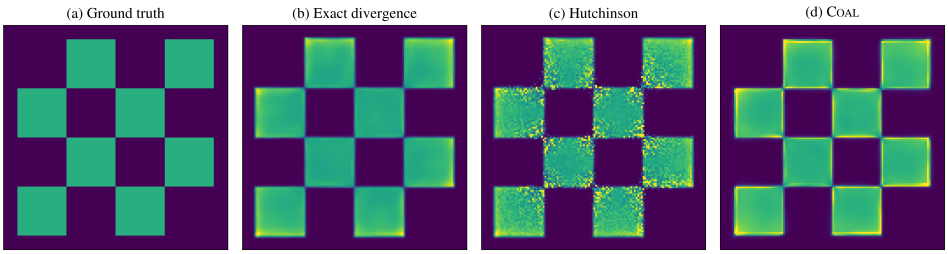


Figure 3: Log-likelihood on the 2D checkerboard at 32 Euler steps: (a) analytic target; (b) exact divergence; (c) Hutchinson with one Rademacher probe per step; (d) COAL. COAL closely tracks exact divergence without Hutchinson's single-probe speckle, indicating a lower-variance likelihood estimator.

Since r itself is $0/0$ at $u = 0$, we use the Taylor expansion $r(u) \approx 1 + u/2 + u^2/6$ for $|u| < 10^{-4}$ and $\texttt{torch.expm1}(u)/u$ otherwise. With the $1/(1-t)$ factor removed in closed form, the integrand is well-conditioned uniformly over $t \in [0, 1]$.

5 EXPERIMENTS

5.1 2D VISUALIZATIONS

As a sanity check, we first illustrate the estimator on a low-dimensional toy where exact ICOV is tractable. We train a Rectified Flow model on the 2D checkerboard (Figure 3a) jointly under (6) and evaluate the log-likelihood with 32 Euler steps under three estimators: exact divergence via iterative backpropagation (tractable at d=2), Hutchinson with one Rademacher probe per step, and our covariance-based estimator with learned σ_θ^2. Panel (b) shows that exact divergence recovers the checkerboard cleanly, indicating that the trained velocity induces the correct push-forward. Hutchinson (Figure 3c) reproduces the coarse structure but is overlaid with pronounced per-point speckle, most visible in low-density off-cells where the single-probe variance dominates the signal. COAL (Figure 3d) is deterministic and visually close to the exact-divergence panel (b), indicating that the amortized estimator recovers the correct divergence.

5.2 LIKELIHOOD EVALUATION ON IMAGENET 256

With correctness established on the toy, we turn to a realistic-scale flow where exact divergence is intractable and the question shifts to practical performance under tight compute budgets. We evaluate COAL on a DiT-XL/1 backbone trained on class-conditional ImageNet 256×256 (Deng et al., 2009). The backbone follows the LightningDiT (Yao et al., 2025) construction, extended with boundary parametrization and a variance head for covariance prediction. The model is a latent Rectified Flow operating in the MAR-VAE (Li et al., 2024) latent space (size $16\times16\times16$, $d = 4096$). We train on the full ImageNet training set and evaluate on the $50{,}000$-image validation split with cached latents. Full training hyperparameters are reported in the appendix.

Reference baseline. Since the ground-truth log-density is intractable at this scale, we construct a per-sample reference by evaluating (1) with Hutchinson on an Euler grid of $N_{\text{ref}} = 1024$ steps (1 Rademacher probe per step). At this resolution both the Euler discretization error and the trajectory-integrated probe variance are an order of magnitude below the inter-method gap, so the 1024-step Hutchinson BPD serves as a practical per-sample proxy for the true log-likelihood (see Appendix B.1).

Protocol. We report per-sample likelihoods in bits-per-dimension (BPD), the per-coordinate, base-2 negative log-likelihood (NLL) standard for image models; smaller is better. Let $\mathrm{BPD}_i^\star$ denote the reference BPD for validation image i, and $\mathrm{BPD}_i^{(m,N)}$ the value produced by method $m \in \{\text{learned}, \text{Hutchinson}\}$ with N integration steps. Over the $|\mathcal{V}| = 50{,}000$ validation images, we report

$$\mathrm{BPD}_i^{(m,N)} = -\frac{1}{d}\log_2 p_\theta^{(m,N)}(\boldsymbol{x}_i), \qquad \mathrm{RMSE}(m,N) = \sqrt{\frac{1}{|\mathcal{V}|}\sum_{i\in\mathcal{V}}\left(\mathrm{BPD}_i^{(m,N)} - \mathrm{BPD}_i^\star\right)^2},$$

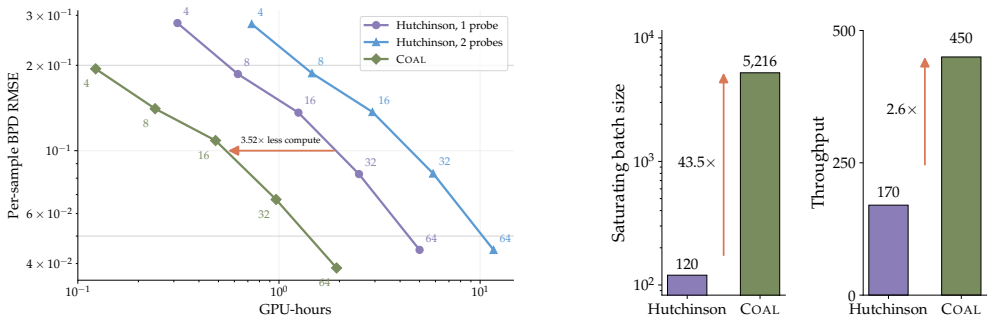


Figure 4: COAL vs. Hutchinson on the DiT-XL latent flow over the 50,000-image ImageNet-256 validation split. *Left:* per-sample BPD RMSE against the 1024-step reference as a function of total evaluation compute (GPU-hours); marker labels are integration steps per sample, the serial integrator depth that cannot be amortized across GPUs. *Right:* memory-saturating batch size on an 80GB H100 (left bars) and maximum single-GPU throughput in ODE steps per second (right bars).

so that the metric directly measures per-sample readout accuracy. We use per-sample RMSE rather than a dataset-averaged BPD because the latter is insensitive to per-sample errors that cancel in the mean. We sweep $N \in \{4, 8, 16, 32, 64\}$ for both the learned head and Hutchinson.

Implementation. Both methods share the backbone and are integrated on a uniform Euler grid from $t = 1$ to $t = 0$ with the same deterministic initial state $z_{t=1}$ given by the encoded latent. The velocity backbone runs under `bf16` autocast while the log-density integral is accumulated in `fp32` to avoid catastrophic cancellation at large N. The learned estimator is deterministic; Hutchinson probe vectors are redrawn per step. Evaluation is distributed across 4 GPUs, with each method at its memory-saturating batch size.

Results. The left panel of Figure 4 reports per-sample BPD RMSE against the 1024-step reference as a function of integration compute (total GPU-hours over the 50,000 validation images) for COAL and for Hutchinson at $n_{\text{ps}} \in \{1, 2\}$ Rademacher probes per step. Marker labels indicate the number of integration steps per sample, the serial integrator depth that cannot be amortized across GPUs. COAL Pareto-dominates both Hutchinson variants over the entire swept range: at matched RMSE per sample, COAL reaches the target with roughly $3.5\times$ less compute than the strongest Hutchinson baseline. The right panel explains part of this compute gap at the hardware level. On an 80GB H100, Hutchinson's probe-dependent autodifferentiation limits the memory-saturating batch size to 120, whereas the deterministic COAL readout reaches 5,216 latents per batch, a $43.5\times$ increase in available batch parallelism. Even ignoring this larger batch, COAL executes 450 latent ODE steps per second on one GPU versus 170 for Hutchinson, giving a $2.6\times$ single-GPU throughput gain.

5.3 ABLATIONS

Ablations use the LightningDiT-B backbone rather than the DiT-XL of Section 5.2, since retraining DiT-XL per cell is too costly. Unless noted, evaluation matches Section 5.2: BPD-RMSE at $N{=}64$ Euler steps against a 256-step Hutchinson reference computed with the *same* velocity backbone, on the full 50,000-image ImageNet validation split.

A1: Scalar vs. dense-diagonal variance head. This ablation isolates the supervision-density claim of Section 4.1: a scalar trace target supplies less learning signal per minibatch element than the per-coordinate diagonal, and at $d = 4096$ this gap slows the variance branch's convergence. To eliminate any coupling between the variance loss and the velocity backbone, we first train a velocity model under the MSE (2) alone to convergence. From this shared checkpoint we branch a variance head off the $(D{-}D')$-th transformer block with $D'{=}2$, append two additional transformer layers, and *freeze the velocity backbone in its entirety* for the remainder of training, so any downstream BPD-RMSE gap is attributable to the variance branch alone. Two variants are trained in parallel under identical optimizer and schedule. The *scalar* head $\sigma^2_\theta(\boldsymbol{x}, t) \in \mathbb{R}$ (two transformer layers, linear projection, token mean-pool) is trained under (4) with the parametrization $\log \sigma^2_\theta(\boldsymbol{x}, t) = \log d + (1 - t)\, s_\theta(\boldsymbol{x}, t)$, so that $\sigma^2_\theta(\boldsymbol{x}, 1) = d$ matches the optimal trace at the boundary. The *dense-diagonal* head $\sigma^2_\theta(\boldsymbol{x}, t) \in \mathbb{R}^d$

Table 1: **Variance-head ablations on DiT-B.** Both subtables follow the protocol of Section 5.3; lower is better on every metric.

D'	Branch (M)	Total (M)	BPD-RMSE ↓
0	1.18	131.89	0.0325
1	11.82	142.52	0.0283
2	22.45	153.15	**0.0220**
3	33.08	163.79	0.0250
4	43.71	174.42	0.0285

(a) Variance-branch depth D' with the dense-diagonal head, each cell jointly trained from scratch; D'=2 is the DiT-B operating point.

Cell	v-bnd	σ^2-bnd	NLL ↓	BPD-RMSE ↓	FID-50k ↓
A	ON	ON	**0.2912**	**0.0220**	**9.87**
B	ON	OFF	0.2948	0.0456	10.75
C	OFF	ON	0.3019	0.0331	11.95
D	OFF	OFF	0.3043	0.0351	12.28

(b) Boundary parametrization 2×2 of v and σ^2; each cell jointly trains its own velocity and variance head.

(two transformer layers, linear projection) is trained under (6) with the per-coordinate parametrization of Section 4.2. Each branch runs for 50,000 steps with the reference cached once on the frozen velocity; BPD-RMSE is logged every 1,000 steps. Figure 5 plots both training curves: the dense-diagonal head reaches the scalar head's 50k-step BPD-RMSE in about 18.9k steps, a 2.6× step reduction. It continues to improve beyond, attaining 0.0311 BPD-RMSE at 50k steps versus 0.0369 for the scalar head, a 15.6% lower error. The per-coordinate target both accelerates convergence and lowers the attainable error, supporting the dense-diagonal design of Section 4.1.

A2: Variance-branch depth D'. This ablation asks how much added capacity the variance branch requires on top of the shared trunk. Larger D' buys more capacity, but every additional layer enlarges the per-step inference cost linearly; we want the smallest D' at which BPD-RMSE saturates. Each cell is trained jointly from scratch under (6), with identical optimizer and schedule across cells: a dense-diagonal variance head with σ^2-boundary parametrization branches off the $(D-D')$-th block, with $D' \in \{0, 1, 2, 3, 4\}$ added transformer layers, where D'=0 degenerates to a per-token linear projection directly off the shared trunk's final block. Because each cell has its own velocity, we recompute the Hutchinson reference per cell on that cell's velocity, so each row's BPD-RMSE is measured against the likelihood the row actually trained. Table 1a reports BPD-RMSE alongside the variance branch's parameter count and the total model size. BPD-RMSE drops monotonically from D'=0 to D'=2 and regresses mildly for $D' \in \{3, 4\}$, reflecting the slower convergence of larger heads under the matched 20k-step budget; D'=2 is the sweet spot, adding only ∼17% parameter overhead (22.45M of 131.89M).

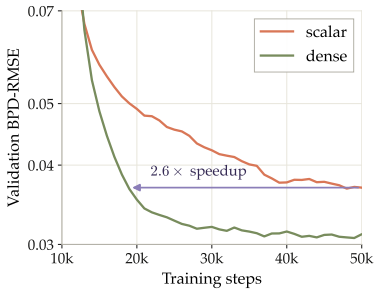


Figure 5: **A1: scalar vs. dense-diagonal variance head.** Validation BPD-RMSE on a frozen velocity; dense-diagonal converges faster, to lower error.

A3: Boundary parametrization of v and σ^2. This ablation tests the design choices of Section 4.2: that baking the terminal-time conditions $v_\theta(\boldsymbol{x}, 1) = \boldsymbol{x}$ and $\sigma^2_\theta(\boldsymbol{x}, 1) = \mathbf{1}$ into the architecture, rather than letting the model learn them from data, improves the resulting likelihood estimator. We arrange the two parametrizations in a 2×2: the v-boundary axis toggles $v_\theta(\boldsymbol{x}, t) = \boldsymbol{x} + (1-t)\, m_\theta(\boldsymbol{x}, t)$ versus a freely parametrized $v_\theta = m_\theta$, and the σ^2-boundary axis toggles $\log \sigma^2_\theta(\boldsymbol{x}, t) = (1-t)\, s_\theta(\boldsymbol{x}, t)$ versus the unconstrained $\log \sigma^2_\theta = s_\theta$. We train each of the four variants jointly from scratch under (6) on DiT-B at matched compute. Because each row has its own velocity, we recompute the Hutchinson reference per row on that row's velocity, so each row's BPD-RMSE is measured against the likelihood the row actually trained. We also report FID-50k at fixed sampling depth, measuring each parametrization's effect on generation quality. Table 1b reports the four cells. The full boundary-parametrized model is best on every metric, while the all-OFF cell is worst on NLL and FID-50k; the v-boundary drives the larger FID gain (2.08 along the v-axis, C→A, vs. 0.88 along σ^2, B→A), since it directly shapes the sampling velocity. We therefore recommend both parametrizations as defaults.

6 Related Work

Likelihood estimation for continuous flow models. The standard estimator integrates the velocity divergence along the trajectory via the instantaneous change-of-variables (ICOV) formula (1) (Chen

et al., 2018; Grathwohl et al., 2019). Exact divergence costs $\mathcal{O}(d^3)$ per step on Transformer flows. The Hutchinson stochastic trace estimator (Hutchinson, 1989; Grathwohl et al., 2019; Meyer et al., 2021) reduces this to K Rademacher vector-Jacobian products, but still requires K backward passes per step at non-vanishing variance. A second line constrains the architecture so log-determinants are tractable by design: coupling-based (Dinh et al., 2017; Kingma & Dhariwal, 2018; Ho et al., 2019), residual (Chen et al., 2019), and autoregressive-transformer (Zhai et al., 2025) normalizing flows (see Papamakarios et al. (2021) for a survey); Verlet Flows (Erives et al., 2024) for exact likelihoods on the discretized map; and HollowFlow (Gloy & Olsson, 2025), which enforces a block-diagonal Jacobian via hollow message passing. Huang et al. (2025) add a divergence-alignment term to the Rectified Flow loss that tightens the total-variation bound, and Herold & Zaid (2024) survey density estimation for flow-based generators. A third line targets *flow-map* models: FALCON (Rehman et al., 2025a) regularizes few-step flows toward invertibility, still paying an $\mathcal{O}(d^3)$ determinant per step, while F2D2 (Ai et al., 2025) jointly distills the sampling map and a likelihood channel under Hutchinson supervision. Each route either pays extra backward passes for stochastic Jacobian probes, constrains the architecture, or still incurs $\mathcal{O}(d^3)$ computation costs per step; COAL is, to our knowledge, the first deterministic, Jacobian-free, architecture-agnostic per-sample likelihood estimator for general Rectified Flows.

Variational diffusion likelihoods and learned-variance regression. Diffusion models admit a hierarchical variational lower bound on log-likelihood, originally for nonequilibrium-thermodynamic generative processes (Sohl-Dickstein et al., 2015; Ho et al., 2020) and extended to continuous time as the probability-flow ODE log-density (Song et al., 2021b). Kingma et al. (2021) formulate the continuous-time variational diffusion ELBO with competitive bits-per-dimension on image benchmarks, and Nichol & Dhariwal (2021) introduce a learned per-step variance into the denoising distribution that substantially improves NLL over DDPM. Independently, training a network to output a mean and per-coordinate variance under Gaussian negative log-likelihood can be traced back to Nix & Weigend (1994) and was formalized as aleatoric uncertainty estimation by Kendall & Gal (2017), with Lakshminarayanan et al. (2017) adopting the same per-coordinate Gaussian NLL as the predictive component of deep ensembles.

Conditional-covariance identities for score and divergence. A handful of information-theoretic identities relate Gaussian-channel statistics to conditional covariances in forms far removed from flow-based likelihood: I-MMSE (Guo et al., 2005) links the derivative of mutual information in a Gaussian channel to the MMSE, with parallel results on the geometry of the likelihood ratio (Hatsell & Nolte, 1971), mutual-information gradients in vector Gaussian channels (Palomar & Verdú, 2006), and derivative identities for the Gaussian-noise conditional-mean estimator (Dytso et al., 2020). In generative modeling, adjacent structure surfaces only in scattered, purpose-specific uses aimed at goals other than likelihood: Meng et al. (2021) iterate Tweedie's formula for higher-order *score* derivatives; Kong et al. (2023) use an I-MMSE-style decomposition to rewrite the diffusion ELBO; Rissanen et al. (2025) transfer denoiser covariance across noise levels for posterior-variance estimates; and EigenScore (Shoushtari et al., 2025) inspects the eigenspectrum of a diffusion denoiser's posterior covariance for OOD detection. These essentially exhaust the related covariance-structure appearances in flow and diffusion, and in none is the conditional covariance made *learnable* during training or closed into a per-sample likelihood for the underlying flow. COAL is the first to amortize this covariance jointly with the velocity during training and convert it into a per-sample likelihood estimator at inference.

7 CONCLUSIONS

We introduced COAL, a practical estimator of sample likelihood for Rectified Flow models. By expressing the divergence of the flow velocity as a closed-form function of the conditional covariance trace and amortizing that trace into a lightweight variance head, the estimator replaces the generic divergence computation inside ICOV with a deterministic forward pass, removing input-space backpropagation from likelihood evaluation entirely. Across experimental settings, from a 2D checkerboard to a DiT-XL latent flow on ImageNet-256, we demonstrate close agreement with the Hutchinson reference at a fraction of the compute, establishing the estimator as a simple and scalable tool for likelihood evaluation in large flow models. Scaling to substantially larger backbones and to other modalities such as video lies beyond the compute budget of this study and is a natural direction for future work.

AI USE STATEMENT

We used generative AI coding assistants to implement parts of the experimental pipeline and engineering infrastructure; all such code was reviewed, tested, and validated by the author against the reference Hutchinson estimator before any result was reported. We also used generative AI tools to edit the manuscript for readability. We did not use generative AI tools to formulate the mathematical claims, to write the proofs, to generate data, or to interpret the experimental results. We take full responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Xinyue Ai, Yutong He, Albert Gu, Ruslan Salakhutdinov, J Zico Kolter, Nicholas Matthew Boffi, and Max Simchowitz. Joint distillation for fast likelihood evaluation and sampling in flow-based models. *arXiv preprint arXiv:2512.02636*, 2025.

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

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K. Duvenaud. Neural ordinary differential equations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/69386f6bb1dfed68692a24c8686939b9-Paper.pdf.

Ricky T. Q. Chen, Jens Behrmann, David K. Duvenaud, and Joern-Henrik Jacobsen. Residual flows for invertible generative modeling. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/5d0d5594d24f0f955548f0fc0ff83d10-Paper.pdf.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using Real NVP. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=HkpbnH9lx.

Alex Dytso, H. Vincent Poor, and Shlomo Shamai Shitz. A general derivative identity for the conditional mean estimator in Gaussian noise and some applications. In *2020 IEEE International Symposium on Information Theory (ISIT)*, pp. 1183–1188. IEEE, 2020. doi: 10.1109/ISIT44484.2020.9174071. URL https://doi.org/10.1109/ISIT44484.2020.9174071.

Ezra Erives, Bowen Jing, and Tommi Jaakkola. Verlet flows: exact-likelihood integrators for flow-based generative models. *arXiv preprint arXiv:2405.02805*, 2024.

Johann Flemming Gloy and Simon Olsson. HollowFlow: Efficient sample likelihood evaluation using hollow message passing. In *Advances in Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=KYlIC6sLhw.

Will Grathwohl, Ricky T. Q. Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. FFJORD: Free-form continuous dynamics for scalable reversible generative models. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJxgknCcK7.

Dongning Guo, Shlomo Shamai, and Sergio Verdú. Mutual information and minimum mean-square error in Gaussian channels. *IEEE Transactions on Information Theory*, 51(4):1261–1282, 2005. doi: 10.1109/TIT.2005.844072. URL https://doi.org/10.1109/TIT.2005.844072.

C. Hatsell and L. Nolte. Some geometric properties of the likelihood ratio (corresp.). *IEEE Transactions on Information Theory*, 17(5):616–618, 1971. doi: 10.1109/TIT.1971.1054672. URL https://doi.org/10.1109/TIT.1971.1054672.

Maternus Herold and Faried Abu Zaid. Elaborating on the value of flow matching for density estimation. In *ICLR Blogposts 2024*, 2024. URL https://iclr-blogposts.github.io/2024/blog/elaborating-on-the-value-of-flow-matching-for-density-estimation/.

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

Jonathan Ho, Xi Chen, Aravind Srinivas, Yan Duan, and Pieter Abbeel. Flow++: Improving flow-based generative models with variational dequantization and architecture design. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2722–2730. PMLR, 2019. URL https://proceedings.mlr.press/v97/ho19a.html.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

Xixi Hu, Runlong Liao, Keyang Xu, Bo Liu, Yeqing Li, Eugene Ie, Hongliang Fei, and Qiang Liu. Improving rectified flow with boundary conditions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18177–18186, October 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Hu_Improving_Rectified_Flow_with_Boundary_Conditions_ICCV_2025_paper.html.

Yuhao Huang, Taos Transue, Shih-Hsin Wang, William M Feldman, Hong Zhang, and Bao Wang. Improving flow matching by aligning flow divergence. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025.

M. F. Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics - Simulation and Computation*, 18(3):1059–1076, 1989. doi: 10.1080/03610918908812806. URL https://doi.org/10.1080/03610918908812806.

Alex Kendall and Yarin Gal. What uncertainties do we need in Bayesian deep learning for computer vision? In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/2650d6089a6d640c5e85b2b88265dc2b-Paper.pdf.

Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 21696–21707. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/b578f2a52a0229873fefc2a4b06377fa-Paper.pdf.

Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/d139db6a236200b21cc7f752979132d0-Paper.pdf.

Xianghao Kong, Rob Brekelmans, and Greg Ver Steeg. Information-theoretic diffusion. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=UvmDCdSPDOW.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/9ef2ed4b7fd2c810847ffa5fa85bce38-Paper.pdf.

Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 56424–56445. Curran Associates, Inc., 2024. doi: 10.52202/079017-1797. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/66e226469f20625aaebddbe47f0ca997-Paper-Conference.pdf.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

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. URL https://openreview.net/forum?id=XVjTT1nw5z.

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 *Computer Vision – ECCV 2024*, Lecture Notes in Computer Science, pp. 23–40. Springer, 2024. doi: 10.1007/978-3-031-72980-5_2. URL https://doi.org/10.1007/978-3-031-72980-5_2.

Chenlin Meng, Yang Song, Wenzhe Li, and Stefano Ermon. Estimating high order gradients of the data distribution by denoising. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 25359–25369. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/d582ac40970f9885836a61d7b2c662e4-Paper.pdf.

Raphael A. Meyer, Cameron Musco, Christopher Musco, and David P. Woodruff. Hutch++: Optimal stochastic trace estimation. In *Symposium on Simplicity in Algorithms (SOSA)*, pp. 142–155. Society for Industrial and Applied Mathematics, 2021. doi: 10.1137/1.9781611976496.16. URL https://doi.org/10.1137/1.9781611976496.16.

Thomas Müller, Brian McWilliams, Fabrice Rousselle, Markus Gross, and Jan Novák. Neural importance sampling. *ACM Transactions on Graphics (ToG)*, 38(5):1–19, 2019.

Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Do deep generative models know what they don't know? In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=H1xwNhCcYm.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8162–8171. PMLR, 2021. URL https://proceedings.mlr.press/v139/nichol21a.html.

David A. Nix and Andreas S. Weigend. Estimating the mean and variance of the target probability distribution. In *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, pp. 55–60 vol.1. IEEE, 1994. doi: 10.1109/ICNN.1994.374138. URL https://doi.org/10.1109/ICNN.1994.374138.

Daniel P. Palomar and Sergio Verdú. Gradient of mutual information in linear vector Gaussian channels. *IEEE Transactions on Information Theory*, 52(1):141–154, 2006. doi: 10.1109/TIT.2005.860424. URL https://doi.org/10.1109/TIT.2005.860424.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021. URL https://jmlr.org/papers/v22/19-1028.html.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023. doi: 10.1109/ICCV51070.2023.00387. URL https://doi.org/10.1109/ICCV51070.2023.00387.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pp. 53728–53741, 2023. doi: 10.52202/075280-2338. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/a85b405ed65c6477a4fe8302b5e06ce7-Abstract-Conference.html.

Danyal Rehman, Tara Akhound-Sadegh, Artem Gazizov, Yoshua Bengio, and Alexander Tong. Falcon: Few-step accurate likelihoods for continuous flows. *arXiv preprint arXiv:2512.09914*, 2025a.

Danyal Rehman, Oscar Davis, Jiarui Lu, Jian Tang, Michael Bronstein, Yoshua Bengio, Alexander Tong, and Avishek Joey Bose. Efficient regression-based training of normalizing flows for boltzmann generators. *arXiv preprint arXiv:2506.01158*, 2025b.

Severi Rissanen, Markus Heinonen, and Arno Solin. Free hunch: Denoiser covariance estimation for diffusion models without extra costs. In *International Conference on Learning Representations*, 2025.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Shirin Shoushtari, Yi Wang, Xiao Shi, M Salman Asif, and Ulugbek S Kamilov. Eigenscore: Ood detection using covariance in diffusion models. *arXiv preprint arXiv:2510.07206*, 2025.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2256–2265. PMLR, 2015. URL https://proceedings.mlr.press/v37/sohl-dickstein15.html.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=St1giarCHLP.

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*, 2021b. URL https://openreview.net/forum?id=PxTIG12RRHS.

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 (CVPR)*, pp. 15703–15712, June 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Yao_Reconstruction_vs._Generation_Taming_Optimization_Dilemma_in_Latent_Diffusion_Models_CVPR_2025_paper.html.

Shuangfei Zhai, Ruixiang Zhang, Preetum Nakkiran, David Berthelot, Jiatao Gu, Huangjie Zheng, Tianrong Chen, Miguel Ángel Bautista, Navdeep Jaitly, and Joshua M. Susskind. Normalizing flows are capable generative models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 74348–74369. PMLR, 2025. URL https://proceedings.mlr.press/v267/zhai25d.html.

A PROOFS

This appendix collects the formal results stated in the main text, together with the supporting boundary-regularity lemma invoked in Sec. 4.2. Each result is restated in full so the appendix is self-contained; proofs follow.

A.1 DIVERGENCE AS CONDITIONAL COVARIANCE

Proposition A.1 (Restatement of Proposition 3.1)**.** *Consider the Rectified Flow forward process $\boldsymbol{X}_t = (1-t)\boldsymbol{X}_0 + t\boldsymbol{X}_1$ with $\boldsymbol{X}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\boldsymbol{X}_1 \sim \pi_1$, and $\boldsymbol{X}_0 \perp \boldsymbol{X}_1$, and assume the marginal density π_t is strictly positive and twice continuously differentiable on $\mathbb{R}^d$ for every $t \in [0,1)$. Fix a point $\boldsymbol{x}$ with $\pi_1(\boldsymbol{x}) > 0$, and suppose $a(\boldsymbol{x}) \coloneqq \nabla \log \pi_1(\boldsymbol{x})$ exists and the Gaussian-localized remainder*

$$R_{\varepsilon,\boldsymbol{x}}(\mathbf{u}) \coloneqq \frac{\pi_1((\boldsymbol{x}+\varepsilon\mathbf{u})/(1-\varepsilon))}{\pi_1(\boldsymbol{x})} - 1 - \varepsilon\, a(\boldsymbol{x})^\top(\boldsymbol{x}+\mathbf{u})$$

satisfies, as $\varepsilon \downarrow 0$,

$$\int_{\mathbb{R}^d} (1+\|\mathbf{u}\|^2)\, |R_{\varepsilon,\boldsymbol{x}}(\mathbf{u})|\, \rho(\mathbf{u})\, \mathrm{d}\mathbf{u} = O_{\boldsymbol{x}}(\varepsilon^2),$$

where ρ is the standard Gaussian density on $\mathbb{R}^d$. For $t \in [0,1]$, let $v_t^*(\boldsymbol{x}) \coloneqq \mathbb{E}\left[\boldsymbol{X}_1 - \boldsymbol{X}_0 \mid \boldsymbol{X}_t = \boldsymbol{x}\right]$ and

$$\tau_t(\boldsymbol{x}) \coloneqq \mathrm{Tr}(\mathrm{Cov}(\boldsymbol{X}_1 - \boldsymbol{X}_0 \mid \boldsymbol{X}_t = \boldsymbol{x}))\,.$$

Define the extended trace expression

$$\mathcal{D}_t(\boldsymbol{x}) \coloneqq \begin{cases} \dfrac{t\tau_t(\boldsymbol{x}) - d}{1-t}, & 0 \le t < 1, \\ d, & t = 1. \end{cases}$$

Then, for every $t \in [0,1]$,

$$\nabla \cdot v_t^*(\boldsymbol{x}) = \mathcal{D}_t(\boldsymbol{x}). \tag{10}$$

Here $v_1^*(\boldsymbol{x}) = \boldsymbol{x}$, $\tau_1(\boldsymbol{x}) = d$, and $\mathcal{D}_1(\boldsymbol{x}) = d$ is the continuous endpoint value of the expression for $t < 1$.

Proof. Fix $t \in (0,1)$ and let

$$m_t(\boldsymbol{x}) \coloneqq \mathbb{E}[\boldsymbol{X}_1 \mid \boldsymbol{X}_t = \boldsymbol{x}]\,, \qquad C_t(\boldsymbol{x}) \coloneqq \mathrm{Cov}(\boldsymbol{X}_1 \mid \boldsymbol{X}_t = \boldsymbol{x})\,.$$

Since, conditionally on $\boldsymbol{X}_1 = \boldsymbol{x}_1$, the random variable $\boldsymbol{X}_t$ is Gaussian with mean $t\boldsymbol{x}_1$ and covariance $(1-t)^2\mathbf{I}$, the marginal density can be written as

$$\pi_t(\boldsymbol{x}) = \int_{\mathbb{R}^d} \frac{1}{(1-t)^d}\, \rho\!\left(\frac{\boldsymbol{x} - t\boldsymbol{x}_1}{1-t}\right) \pi_1(\boldsymbol{x}_1)\, \mathrm{d}\boldsymbol{x}_1,$$

Differentiating this Gaussian convolution gives the Tweedie–Miyasawa identity

$$\nabla \log \pi_t(\boldsymbol{x}) = \frac{t\, m_t(\boldsymbol{x}) - \boldsymbol{x}}{(1-t)^2}, \qquad \text{equivalently} \qquad m_t(\boldsymbol{x}) = \frac{\boldsymbol{x} + (1-t)^2\, \nabla \log \pi_t(\boldsymbol{x})}{t}.$$

Differentiating the posterior mean once more yields

$$\nabla m_t(\boldsymbol{x}) = \frac{t}{(1-t)^2}\, C_t(\boldsymbol{x}). \tag{11}$$

Indeed, the posterior measure of $\boldsymbol{X}_1$ given $\boldsymbol{X}_t = \boldsymbol{x}$ is proportional to $\rho\!\left(\frac{\boldsymbol{x}-t\boldsymbol{x}_1}{1-t}\right) \pi_1(\boldsymbol{x}_1)\, \mathrm{d}\boldsymbol{x}_1$. Therefore, for each coordinate pair (i,j),

$$\begin{aligned} \partial_{x_i} m_{t,j}(\boldsymbol{x}) &= \mathrm{Cov}\left(\boldsymbol{X}_{1,j},\, \partial_{x_i} \log \rho\!\left(\frac{\boldsymbol{x} - t\boldsymbol{X}_1}{1-t}\right) \,\middle|\, \boldsymbol{X}_t = \boldsymbol{x}\right) \\ &= \mathrm{Cov}\left(\boldsymbol{X}_{1,j},\, \frac{t\boldsymbol{X}_{1,i} - x_i}{(1-t)^2} \,\middle|\, \boldsymbol{X}_t = \boldsymbol{x}\right) \\ &= \frac{t}{(1-t)^2}\, \mathrm{Cov}(\boldsymbol{X}_{1,j}, \boldsymbol{X}_{1,i} \mid \boldsymbol{X}_t = \boldsymbol{x})\,, \end{aligned}$$

which proves (11).

Now condition on $\boldsymbol{X}_t = \boldsymbol{x}$. Since

$$\boldsymbol{X}_0 = \frac{\boldsymbol{x} - t\boldsymbol{X}_1}{1-t}, \qquad \boldsymbol{X}_1 - \boldsymbol{X}_0 = \frac{\boldsymbol{X}_1 - \boldsymbol{x}}{1-t},$$

we have

$$v_t^*(\boldsymbol{x}) = \frac{m_t(\boldsymbol{x}) - \boldsymbol{x}}{1-t}, \qquad \mathrm{Cov}(\boldsymbol{X}_1 - \boldsymbol{X}_0 \mid \boldsymbol{X}_t = \boldsymbol{x}) = \frac{1}{(1-t)^2} C_t(\boldsymbol{x}).$$

Taking the divergence of v_t^* at fixed t and using (11),

$$\begin{aligned}
\nabla \cdot v_t^*(\boldsymbol{x}) &= \frac{1}{1-t}\left(\mathrm{Tr}(\nabla m_t(\boldsymbol{x})) - d\right) \\
&= \frac{1}{1-t}\left(\frac{t}{(1-t)^2}\,\mathrm{Tr}(C_t(\boldsymbol{x})) - d\right) \\
&= \frac{t\tau_t(\boldsymbol{x}) - d}{1-t},
\end{aligned}$$

which proves the claim for $t \in (0,1)$. At $t = 0$, $\boldsymbol{X}_t = \boldsymbol{X}_0$ and $\boldsymbol{X}_0 \perp \boldsymbol{X}_1$, so $v_0^*(\boldsymbol{x}) = \mathbb{E}[\boldsymbol{X}_1] - \boldsymbol{x}$ and hence $\nabla \cdot v_0^*(\boldsymbol{x}) = -d$, matching the right-hand side of (10).

It remains to identify the endpoint. At $t = 1$, $\boldsymbol{X}_t = \boldsymbol{X}_1$, and $\boldsymbol{X}_0 \perp \boldsymbol{X}_1$ with $\mathbb{E}[\boldsymbol{X}_0] = \mathbf{0}$, so

$$v_1^*(\boldsymbol{x}) = \mathbb{E}[\boldsymbol{X}_1 - \boldsymbol{X}_0 \mid \boldsymbol{X}_1 = \boldsymbol{x}] = \boldsymbol{x}. \tag{12}$$

Thus $\nabla \cdot v_1^*(\boldsymbol{x}) = d$. Also $\mathrm{Cov}(\boldsymbol{X}_1 - \boldsymbol{X}_0 \mid \boldsymbol{X}_1 = \boldsymbol{x}) = \mathrm{Cov}(\boldsymbol{x} - \boldsymbol{X}_0) = \mathbf{I}$, so $\tau_1(\boldsymbol{x}) = d$ and the right-hand side of (10) is a $0/0$ indeterminate form as $t \uparrow 1$.

To identify the continuous extension of $\tau_t(\boldsymbol{x})$ at $t = 1$, set $\varepsilon := 1 - t$ and, conditional on $\boldsymbol{X}_t = \boldsymbol{x}$, define $\mathbf{U}_\varepsilon := (t\boldsymbol{X}_1 - \boldsymbol{x})/\varepsilon$. For $0 < \varepsilon < 1$, $\mathbf{U}_\varepsilon$ has conditional density

$$q_{\varepsilon,\boldsymbol{x}}(\mathbf{u}) = \frac{\pi_1\left(\dfrac{\boldsymbol{x} + \varepsilon\mathbf{u}}{1-\varepsilon}\right)\rho(\mathbf{u})}{\displaystyle\int_{\mathbb{R}^d} \pi_1\left(\frac{\boldsymbol{x} + \varepsilon\mathbf{w}}{1-\varepsilon}\right)\rho(\mathbf{w})\,\mathrm{d}\mathbf{w}},$$

where ρ is the standard Gaussian density on $\mathbb{R}^d$. Moreover,

$$\boldsymbol{X}_1 = \frac{\boldsymbol{x} + \varepsilon\mathbf{U}_\varepsilon}{t}, \qquad \boldsymbol{X}_0 = -\mathbf{U}_\varepsilon, \qquad \boldsymbol{X}_1 - \boldsymbol{X}_0 = \frac{\boldsymbol{x} + \mathbf{U}_\varepsilon}{t}.$$

Consequently,

$$\tau_t(\boldsymbol{x}) = \frac{1}{t^2}\,\mathrm{Tr}\big(\mathrm{Cov}_{q_{\varepsilon,\boldsymbol{x}}}(\mathbf{U}_\varepsilon)\big)\,. \tag{13}$$

By the stated Gaussian-localized remainder condition,

$$\frac{\pi_1\left(\dfrac{\boldsymbol{x} + \varepsilon\mathbf{u}}{1-\varepsilon}\right)}{\pi_1(\boldsymbol{x})} = 1 + \varepsilon\, a(\boldsymbol{x})^\top(\boldsymbol{x} + \mathbf{u}) + R_{\varepsilon,\boldsymbol{x}}(\mathbf{u}),$$

with $\int (1 + \|\mathbf{u}\|^2)|R_{\varepsilon,\boldsymbol{x}}(\mathbf{u})|\rho(\mathbf{u})\,\mathrm{d}\mathbf{u} = O_{\boldsymbol{x}}(\varepsilon^2)$. This bound and the standard Gaussian identities $\mathbb{E}[\mathbf{U}] = 0$, $\mathbb{E}[\mathbf{U}\mathbf{U}^\top] = \mathbf{I}$, and $\mathbb{E}[\mathbf{U}\mathbf{U}^\top(a^\top\mathbf{U})] = 0$ give

$$\begin{aligned}
\mathbb{E}_{q_{\varepsilon,\boldsymbol{x}}}[\mathbf{U}_\varepsilon] &= \varepsilon a(\boldsymbol{x}) + O_{\boldsymbol{x}}(\varepsilon^2), \\
\mathbb{E}_{q_{\varepsilon,\boldsymbol{x}}}[\mathbf{U}_\varepsilon\mathbf{U}_\varepsilon^\top] &= \mathbf{I} + O_{\boldsymbol{x}}(\varepsilon^2).
\end{aligned}$$

Thus

$$\mathrm{Cov}_{q_{\varepsilon,\boldsymbol{x}}}(\mathbf{U}_\varepsilon) = \mathbf{I} + O_{\boldsymbol{x}}(\varepsilon^2), \qquad \mathrm{Tr}\big(\mathrm{Cov}_{q_{\varepsilon,\boldsymbol{x}}}(\mathbf{U}_\varepsilon)\big) = d + O_{\boldsymbol{x}}(\varepsilon^2). \tag{14}$$

Combining (13) and (14),

$$\tau_t(\boldsymbol{x}) = \frac{d}{t^2} + O_{\boldsymbol{x}}\big((1-t)^2\big)\,, \qquad t \uparrow 1.$$

Therefore the quotient defining $\mathcal{D}_t$ for $t < 1$ satisfies

$$\frac{t\tau_t(\boldsymbol{x}) - d}{1-t} = \frac{d/t - d}{1-t} + O_{\boldsymbol{x}}(1-t)$$
$$= \frac{d}{t} + O_{\boldsymbol{x}}(1-t),$$

and hence $\lim_{t\uparrow 1}(t\tau_t(\boldsymbol{x}) - d)/(1-t) = d$. Together with (12), this proves that the divergence identity extends continuously to $t = 1$, with endpoint value

$$\nabla \cdot v_1^*(\boldsymbol{x}) = d.$$

□

A.2 Optimality of the Joint Gaussian-NLL Objective

Lemma A.2 (Restatement of Lemma 3.2)**.** *Assume the function classes for v_θ and σ_θ^2 are sufficiently rich to represent any measurable vector- and positive scalar-valued function of $(\boldsymbol{x}, t)$. Assume also that the conditional trace below is positive almost surely, or interpret the result in the closure that permits zero variance. Then (v_*, σ_*^2) is a joint minimizer of the objective*

$$\mathcal{L}(\theta) \;\coloneqq\; \mathbb{E}\left[\frac{\|\boldsymbol{X}_1 - \boldsymbol{X}_0 - v_\theta(\boldsymbol{X}_t, t)\|^2}{\sigma_\theta^2(\boldsymbol{X}_t, t)} + \log \sigma_\theta^2(\boldsymbol{X}_t, t)\right] \tag{15}$$

if and only if

$$v_*(\boldsymbol{x}, t) \;=\; \mathbb{E}[\boldsymbol{X}_1 - \boldsymbol{X}_0 \mid \boldsymbol{X}_t = \boldsymbol{x}]\,, \qquad \sigma_*^2(\boldsymbol{x}, t) \;=\; \mathrm{Tr}(\mathrm{Cov}(\boldsymbol{X}_1 - \boldsymbol{X}_0 \mid \boldsymbol{X}_t = \boldsymbol{x}))\,.$$

Proof. Let $\mathbf{Y} \coloneqq \boldsymbol{X}_1 - \boldsymbol{X}_0$. By the law of total expectation, the objective decomposes pointwise over the conditioning variable $(\boldsymbol{X}_t, t)$. Fix a pair $(\boldsymbol{x}, t)$ and define

$$m(\boldsymbol{x}, t) \coloneqq \mathbb{E}[\mathbf{Y} \mid \boldsymbol{X}_t = \boldsymbol{x}]\,, \qquad \tau(\boldsymbol{x}, t) \coloneqq \mathrm{Tr}(\mathrm{Cov}(\mathbf{Y} \mid \boldsymbol{X}_t = \boldsymbol{x}))\,.$$

For arbitrary $\mathbf{a} \in \mathbb{R}^d$ and $s > 0$,

$$\mathbb{E}\Big[\|\mathbf{Y} - \mathbf{a}\|^2 \mid \boldsymbol{X}_t = \boldsymbol{x}\Big] = \mathbb{E}\Big[\|\mathbf{Y} - m(\boldsymbol{x}, t) + m(\boldsymbol{x}, t) - \mathbf{a}\|^2 \mid \boldsymbol{X}_t = \boldsymbol{x}\Big]$$
$$= \tau(\boldsymbol{x}, t) + \|m(\boldsymbol{x}, t) - \mathbf{a}\|^2\,,$$

where the cross term vanishes by the definition of the conditional mean. Hence the conditional contribution to (15) is

$$\ell_{\boldsymbol{x},t}(\mathbf{a}, s) = \frac{\tau(\boldsymbol{x}, t) + \|m(\boldsymbol{x}, t) - \mathbf{a}\|^2}{s} + \log s.$$

For fixed s, this expression is uniquely minimized over $\mathbf{a}$ by $\mathbf{a} = m(\boldsymbol{x}, t)$. Substituting this value leaves the one-dimensional problem

$$\inf_{s>0}\left\{\frac{\tau(\boldsymbol{x}, t)}{s} + \log s\right\}.$$

When $\tau(\boldsymbol{x}, t) > 0$, the derivative is

$$-\frac{\tau(\boldsymbol{x}, t)}{s^2} + \frac{1}{s} = \frac{s - \tau(\boldsymbol{x}, t)}{s^2},$$

and the objective diverges as $s \downarrow 0$ and as $s \to \infty$; therefore the unique minimizer is $s = \tau(\boldsymbol{x}, t)$. Consequently, the unique pointwise minimizer is

$$\mathbf{a} = m(\boldsymbol{x}, t), \qquad s = \tau(\boldsymbol{x}, t).$$

Integrating these pointwise optima over the law of $(\boldsymbol{X}_t, t)$ gives a global minimizer of (15). Conversely, if a candidate differs from these pointwise values on a set of positive measure, replacing it by the pointwise optimizer on that set strictly decreases the conditional loss and hence strictly decreases $\mathcal{L}$, contradicting global optimality. Thus, up to null-set modifications,

$$v_*(\boldsymbol{x}, t) = \mathbb{E}[\boldsymbol{X}_1 - \boldsymbol{X}_0 \mid \boldsymbol{X}_t = \boldsymbol{x}]\,, \qquad \sigma_*^2(\boldsymbol{x}, t) = \mathrm{Tr}(\mathrm{Cov}(\boldsymbol{X}_1 - \boldsymbol{X}_0 \mid \boldsymbol{X}_t = \boldsymbol{x}))\,.$$

If $\tau(\boldsymbol{x}, t) = 0$ on a null set, the conclusion is unchanged. If it occurs on a set of positive measure while σ_θ^2 is constrained to be strictly positive, the Gaussian-NLL objective has an infimum approached as $\sigma_\theta^2 \downarrow 0$ on that set rather than an attained positive minimizer; the stated minimizer result is therefore understood under the standard nondegeneracy condition $\tau(\boldsymbol{x}, t) > 0$ almost surely, or after taking the closure that allows zero variance. □

Corollary A.3 (Per-coordinate diagonal extension)**.** *Under the assumptions of Lemma A.2, let* $\sigma_\theta^2 : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d_{>0}$ *be a vector-valued head trained under*

$$\mathcal{L}(\theta) \;\coloneqq\; \mathbb{E}\left[\sum_{i=1}^{d} \frac{\left(\boldsymbol{X}_{1,i} - \boldsymbol{X}_{0,i} - v_{\theta,i}(\boldsymbol{X}_t, t)\right)^2}{\sigma_{\theta,i}^2(\boldsymbol{X}_t, t)} + \log \sigma_{\theta,i}^2(\boldsymbol{X}_t, t)\right]. \tag{16}$$

Then the joint minimizer satisfies, for every coordinate $i \in \{1, \dots, d\}$,

$$v_{*,i}(\boldsymbol{x}, t) \;=\; \mathbb{E}[\boldsymbol{X}_{1,i} - \boldsymbol{X}_{0,i} \mid \boldsymbol{X}_t = \boldsymbol{x}], \qquad \sigma_{*,i}^2(\boldsymbol{x}, t) \;=\; \mathrm{Var}(\boldsymbol{X}_{1,i} - \boldsymbol{X}_{0,i} \mid \boldsymbol{X}_t = \boldsymbol{x}),$$

and consequently $\sum_{i=1}^d \sigma_{*,i}^2(\boldsymbol{x}, t) = \mathrm{Tr}(\mathrm{Cov}(\boldsymbol{X}_1 - \boldsymbol{X}_0 \mid \boldsymbol{X}_t = \boldsymbol{x}))$.

Proof. Again set $\mathbf{Y} \coloneqq \boldsymbol{X}_1 - \boldsymbol{X}_0$. Conditioning on $(\boldsymbol{X}_t, t) = (\boldsymbol{x}, t)$, the loss in (16) becomes

$$\sum_{i=1}^{d} \left\{ \frac{\mathbb{E}\big[(\mathbf{Y}_i - a_i)^2 \mid \boldsymbol{X}_t = \boldsymbol{x}\big]}{s_i} + \log s_i \right\}, \qquad s_i > 0.$$

For each coordinate, write

$$m_i(\boldsymbol{x}, t) \coloneqq \mathbb{E}[\mathbf{Y}_i \mid \boldsymbol{X}_t = \boldsymbol{x}], \qquad \tau_i(\boldsymbol{x}, t) \coloneqq \mathrm{Var}(\mathbf{Y}_i \mid \boldsymbol{X}_t = \boldsymbol{x}).$$

Then

$$\mathbb{E}\big[(\mathbf{Y}_i - a_i)^2 \mid \boldsymbol{X}_t = \boldsymbol{x}\big] = \tau_i(\boldsymbol{x}, t) + (m_i(\boldsymbol{x}, t) - a_i)^2.$$

The ith summand therefore has exactly the scalar form treated in Lemma A.2; it is minimized by

$$a_i = m_i(\boldsymbol{x}, t), \qquad s_i = \tau_i(\boldsymbol{x}, t),$$

with the same nondegeneracy convention for zero conditional variance. Since the conditional objective is a sum of these decoupled coordinatewise objectives, the joint minimizer satisfies

$$v_{*,i}(\boldsymbol{x}, t) = \mathbb{E}[\boldsymbol{X}_{1,i} - \boldsymbol{X}_{0,i} \mid \boldsymbol{X}_t = \boldsymbol{x}], \qquad \sigma_{*,i}^2(\boldsymbol{x}, t) = \mathrm{Var}(\boldsymbol{X}_{1,i} - \boldsymbol{X}_{0,i} \mid \boldsymbol{X}_t = \boldsymbol{x}).$$

Finally, the trace of a covariance matrix is the sum of its diagonal entries, so

$$\sum_{i=1}^{d} \sigma_{*,i}^2(\boldsymbol{x}, t) = \sum_{i=1}^{d} \mathrm{Var}(\boldsymbol{X}_{1,i} - \boldsymbol{X}_{0,i} \mid \boldsymbol{X}_t = \boldsymbol{x}) = \mathrm{Tr}(\mathrm{Cov}(\boldsymbol{X}_1 - \boldsymbol{X}_0 \mid \boldsymbol{X}_t = \boldsymbol{x})).$$

□

A.3 Boundary Regularity of the Divergence at $t = 1$

Proposition A.4 (Finite limit of the optimal divergence at $t = 1$)**.** *Under the assumptions of Proposition A.1, suppose in addition that the map* $t \mapsto \mathrm{Tr}(\mathrm{Cov}(\boldsymbol{X}_1 - \boldsymbol{X}_0 \mid \boldsymbol{X}_t = \boldsymbol{x}))$ *is continuously differentiable in a neighborhood of* $t = 1$ *for* π_1*-almost every* $\boldsymbol{x}$*. Then the boundary values*

$$v_1^*(\boldsymbol{x}) \;=\; \boldsymbol{x}, \qquad \mathrm{Cov}(\boldsymbol{X}_1 - \boldsymbol{X}_0 \mid \boldsymbol{X}_1 = \boldsymbol{x}) \;=\; \mathbf{I}$$

hold, and the limit $\lim_{t \to 1^-} \nabla \cdot v_t^*(\boldsymbol{x})$ *exists and is finite, with value obtained by L'Hôpital's rule applied to* (10)*.*

Proof. At $t = 1$ we have $\boldsymbol{X}_t = \boldsymbol{X}_1$. Since $\boldsymbol{X}_0 \perp \boldsymbol{X}_1$ and $\mathbb{E}[\boldsymbol{X}_0] = \mathbf{0}$,

$$v_1^*(\boldsymbol{x}) = \mathbb{E}[\boldsymbol{X}_1 - \boldsymbol{X}_0 \mid \boldsymbol{X}_1 = \boldsymbol{x}] = \boldsymbol{x} - \mathbb{E}[\boldsymbol{X}_0] = \boldsymbol{x}.$$

Moreover, conditioning on $\boldsymbol{X}_1 = \boldsymbol{x}$ makes $\boldsymbol{X}_1$ deterministic and leaves the distribution of $\boldsymbol{X}_0$ unchanged. Hence

$$\mathrm{Cov}(\boldsymbol{X}_1 - \boldsymbol{X}_0 \mid \boldsymbol{X}_1 = \boldsymbol{x}) = \mathrm{Cov}(\boldsymbol{x} - \boldsymbol{X}_0 \mid \boldsymbol{X}_1 = \boldsymbol{x}) = \mathrm{Cov}(\boldsymbol{X}_0) = \mathbf{I}.$$

Define $\tau(t, \boldsymbol{x}) \coloneqq \mathrm{Tr}(\mathrm{Cov}(\boldsymbol{X}_1 - \boldsymbol{X}_0 \mid \boldsymbol{X}_t = \boldsymbol{x}))$. The preceding covariance identity gives $\tau(1, \boldsymbol{x}) = d$. Therefore the numerator and denominator in (10) both vanish as $t \to 1^-$:

$$t\tau(t, \boldsymbol{x}) - d \longrightarrow d - d = 0, \qquad 1 - t \longrightarrow 0.$$

By the assumed continuous differentiability of $t \mapsto \tau(t, \boldsymbol{x})$ near $t = 1$, L'Hôpital's rule applies to the one-sided limit and yields

$$\begin{aligned}\lim_{t\to 1^-} \nabla \cdot v_t^*(\boldsymbol{x}) &= \lim_{t\to 1^-} \frac{t\tau(t, \boldsymbol{x}) - d}{1 - t} \\ &= \lim_{t\to 1^-} \frac{\tau(t, \boldsymbol{x}) + t\,\partial_t \tau(t, \boldsymbol{x})}{-1} \\ &= -d - \partial_t \tau(1, \boldsymbol{x}),\end{aligned}$$

which is finite for π_1-almost every $\boldsymbol{x}$ by the regularity assumption. □

Proposition A.5 (Finite limit under boundary parametrization)**.** *Let the velocity and variance heads be parametrized as*

$$v_\theta(\boldsymbol{x}, t) \;=\; \boldsymbol{x} + (1 - t)\, m_\theta(\boldsymbol{x}, t), \qquad \log \sigma_\theta^2(\boldsymbol{x}, t) \;=\; (1 - t)\, s_\theta(\boldsymbol{x}, t),$$

with $m_\theta : \mathbb{R}^d \times [0, 1] \to \mathbb{R}^d$ and $s_\theta : \mathbb{R}^d \times [0, 1] \to \mathbb{R}^d$ continuously differentiable in a neighborhood of $t = 1$. Then the divergence integrand of (5)*,*

$$\Delta_\theta(\boldsymbol{x}, t) \;\coloneqq\; \frac{1}{1 - t}\left(t \sum_{i=1}^{d} \sigma_{\theta,i}^2(\boldsymbol{x}, t) - d \right),$$

extends continuously to $t = 1$, with $\lim_{t\to 1^-} \Delta_\theta(\boldsymbol{x}, t) = -d + \sum_{i=1}^d s_{\theta,i}(\boldsymbol{x}, 1)$.

Proof. The velocity parametrization immediately gives $v_\theta(\boldsymbol{x}, 1) = \boldsymbol{x}$. For the variance head, $\sigma_{\theta,i}^2(\boldsymbol{x}, t) = \exp((1 - t)s_{\theta,i}(\boldsymbol{x}, t))$, so $\sigma_{\theta,i}^2(\boldsymbol{x}, 1) = 1$ for every coordinate. Thus the numerator of $\Delta_\theta(\boldsymbol{x}, t)$ vanishes at $t = 1$.

Set $u_i(t) \coloneqq (1 - t)s_{\theta,i}(\boldsymbol{x}, t)$ and define $r(u) \coloneqq (e^u - 1)/u$ for $u \neq 0$ with $r(0) \coloneqq 1$. The function r is continuous at 0, and $e^{u_i(t)} = 1 + u_i(t)r(u_i(t))$. Therefore, for $t < 1$,

$$\begin{aligned}\Delta_\theta(\boldsymbol{x}, t) &= \frac{t \sum_{i=1}^d [1 + (1 - t)s_{\theta,i}(\boldsymbol{x}, t)r(u_i(t))] - d}{1 - t} \\ &= \frac{d(t - 1)}{1 - t} + t \sum_{i=1}^{d} s_{\theta,i}(\boldsymbol{x}, t)r(u_i(t)) \\ &= -d + t \sum_{i=1}^{d} s_{\theta,i}(\boldsymbol{x}, t)r(u_i(t)).\end{aligned}$$

Since each $s_{\theta,i}$ is continuous at $t = 1$, we have $u_i(t) \to 0$ and $r(u_i(t)) \to 1$ as $t \to 1^-$. Hence $\lim_{t\to 1^-} \Delta_\theta(\boldsymbol{x}, t) = -d + \sum_{i=1}^d s_{\theta,i}(\boldsymbol{x}, 1)$, and defining $\Delta_\theta(\boldsymbol{x}, 1)$ to equal this value gives a continuous extension to $t = 1$. Equivalently, the normalized divergence has the continuous extension $\lim_{t\to 1^-} \frac{1}{d}\Delta_\theta(\boldsymbol{x}, t) = -1 + \frac{1}{d}\sum_{i=1}^d s_{\theta,i}(\boldsymbol{x}, 1)$, which is the limit of the stable form in (9). □

B Additional Experimental Details

B.1 Convergence of the 1024-Step Hutchinson Reference

Section 5.2 uses a 1024-step, single-probe Hutchinson estimate as a per-sample pseudo-exact reference. To substantiate the claim that this proxy is sufficiently converged to support the comparisons in Figure 4, we sweep the same Hutchinson estimator across $N \in \{4, 8, 16, 32, 64, 128, 256, 512\}$ on a 1,000-image subset of the ImageNet-256 validation split (sampled uniformly without replacement, seed 0) and measure per-sample BPD-RMSE against the N=1024 value evaluated on the same subset.

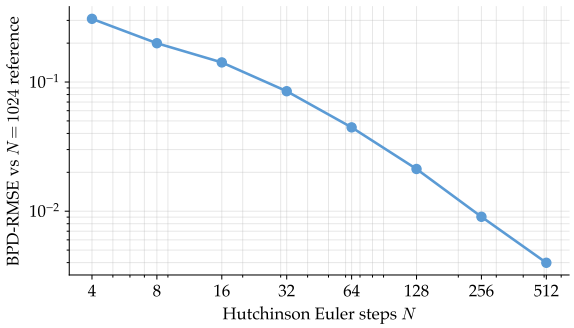


Figure 6: Self-convergence of the Hutchinson reference: per-sample BPD-RMSE between the N-step single-probe Hutchinson estimator and the $N{=}1024$ value on a 1,000-image validation subset. The doubling-N slope is consistent with $\mathcal{O}(1/N)$ Euler error.

Figure 6 plots the result. The RMSE halves with each doubling of N (the slope expected from first-order Euler discretization on a smooth integrand), and the last measured point $\mathrm{RMSE}(512\,\mathrm{vs.}\,1024) = 3.99 \times 10^{-3}$ BPD per dimension is approximately 9% of $\mathrm{RMSE}(64\,\mathrm{vs.}\,1024) = 4.46 \times 10^{-2}$. Extrapolating the same $\mathcal{O}(1/N)$ rate one further doubling places the residual error of the 1024-step reference itself near $\sim 2 \times 10^{-3}$ BPD per dimension, more than an order of magnitude below the inter-method gaps that Figure 4 resolves. We therefore treat the 1024-step Hutchinson estimate as a sufficiently accurate per-sample reference for the comparisons reported in the main paper.

B.2 Bias–Variance Decomposition of the BPD-RMSE in Figure 4

Figure 4 reports per-sample BPD-RMSE against the 1024-step reference. To complement the aggregated RMSE, we decompose the squared error per sample into bias and variance components. Let $\mathrm{BPD}_i^{(m,N)}$ denote the absolute bits-per-dimension produced by method m at N Euler steps for image i, and $\mathrm{BPD}_i^{\star}$ the value under the 1024-step Hutchinson reference. Define the per-sample error $e_i^{(m,N)} := \mathrm{BPD}_i^{(m,N)} - \mathrm{BPD}_i^{\star}$. Then

$$\mathrm{Bias}^{(m,N)} = \frac{1}{n}\sum_i e_i^{(m,N)}, \qquad \mathrm{Std}^{(m,N)} = \sqrt{\frac{1}{n}\sum_i \left(e_i^{(m,N)} - \mathrm{Bias}^{(m,N)}\right)^2},$$

and $\mathrm{Bias}^2 + \mathrm{Std}^2 = \mathrm{MSE}$, which exactly recovers the RMSE plotted in Figure 4. Table 2 reports both components on the 50,000-image ImageNet-256 validation split.

Method	$N{=}16$	$N{=}32$	$N{=}64$
Hutchinson, $n_{\mathrm{ps}}{=}1$	$-0.085\,/\,0.106$	$-0.052\,/\,0.064$	$-0.028\,/\,0.035$
Hutchinson, $n_{\mathrm{ps}}{=}2$	$-0.087\,/\,0.106$	$-0.053\,/\,0.064$	$-0.029\,/\,0.034$
Coal	$-0.101\,/\,0.040$	$-0.059\,/\,0.033$	$-0.031\,/\,0.023$

Table 2: Bias and standard deviation of per-sample BPD error against the 1024-step Hutchinson reference, reported as Bias / Std in BPD per dimension.

For calibration, the per-image BPD assigned by the 1024-step Hutchinson reference itself has mean 0.187 and standard deviation 0.467 across the 50,000 validation images, so the natural image-to-image spread of the BPD signal is ~ 0.47 BPD per dimension. The per-method error standard deviations in Table 2 (largest ~ 0.106) are therefore all below the content-driven variance, and Coal's ~ 0.04 error std is even smaller relative to it.

The standard-deviation term differs sharply: Coal attains substantially lower per-sample error std than Hutchinson across all N. The RMSE advantage in Figure 4 is therefore a variance-reduction effect, not a bias effect; consequently, downstream uses that are sensitive to per-sample noise (out-

of-distribution detection, importance sampling) benefit from COAL more than the matched-RMSE comparison alone suggests.

B.3 DATASET AND LATENT PREPROCESSING

We train and evaluate on the ImageNet-1k (Deng et al., 2009) train and validation splits (1,281,167 and 50,000 images respectively). Pixels are normalized to $[-1, 1]$ via mean = std = 0.5. Training applies a random horizontal flip; validation uses the deterministic crop with no augmentation.

We operate in the latent space of the MAR-VAE (Li et al., 2024), whose encoder maps a 256×256 image to a $16\times16\times16$ latent ($d = 4{,}096$). Latents are rescaled by a fixed factor of $1/0.2325$ to bring them to approximately unit variance and are precomputed once and cached as `NPZ` shards for both splits. Class conditioning uses the 1000-way ImageNet label.

B.4 VARIANCE-HEAD ARCHITECTURE

COAL implements the likelihood model as a BoundaryLikelihoodDiT with a shared transformer prefix and two symmetric prediction branches. The MAR-VAE latent of shape $16\times16\times16$ enters the model as a sequence of 256 tokens via a patch embedding with patch size 1 over the 16×16 spatial grid: each spatial position's 16 latent channels are linearly projected into the hidden dimension, and the resulting tokens are the shared-prefix input. After the shared prefix, the velocity branch and variance branch each apply D' additional DiT blocks with the same width, attention-head count, MLP ratio, time conditioning, and class conditioning as the shared blocks. Each branch then uses a per-token spatial readout that maps the hidden representation back to the 16 latent channels and unpatchifies the token sequence to a $16\times16\times16$ tensor. The velocity branch emits the raw residual m_θ, which is converted to $v_\theta(\boldsymbol{x}, t) = \boldsymbol{x} + (1-t)m_\theta(\boldsymbol{x}, t)$, enforcing $v_\theta(\boldsymbol{x}, 1) = \boldsymbol{x}$. The variance branch emits the raw log-variance residual s_θ, not a variance directly; the dense-diagonal variance used in the Gaussian NLL is

$$\sigma^2_{\theta,i}(\boldsymbol{x}, t) = \exp((1-t)s_{\theta,i}(\boldsymbol{x}, t)),$$

so $\sigma^2_\theta(\boldsymbol{x}, 1) = \mathbf{1}$ holds by construction. At likelihood-evaluation time, the implementation uses the raw s_θ output in the equivalent `expm1`-stabilized divergence formula.

DiT-XL configuration (main results). The configured total depth is 28 DiT blocks at hidden size 1,152 with 16 attention heads. With D'=4, the instantiated model uses 24 shared blocks followed by two separate 4-block branches, one for m_θ and one for s_θ. The ImageNet-256 run uses patch size 1, MAR latent dimension 16, 1000 class labels with classifier-free dropout (Ho & Salimans, 2022) 0.1, rotary positional embeddings, RMSNorm, SwiGLU MLPs, and QK normalization. Total parameter count is $\sim$773M.

DiT-B configuration (ablations). The configured total depth is 12 DiT blocks at hidden size 768 with 12 attention heads. The default ablation configuration uses D'=2, giving 10 shared blocks followed by two separate 2-block branches. The variance branch contributes 22.45M parameters on top of a 131.89M base model, a $\sim 17\%$ overhead. Both backbones use the same raw-s_θ dense-diagonal log-variance output and the same boundary parametrization.

B.5 TRAINING HYPERPARAMETERS

The main DiT-XL run is trained with AdamW (β_1=0.9, β_2=0.95, ϵ=10^{-8}, weight decay 0.01) at peak learning rate 2×10^{-4}, with 20,000 steps of linear warmup followed by a constant schedule (minimum learning rate 10^{-6}). The global batch size is 2048, achieved with a gradient accumulation factor of 2, and the total training length is 800,000 optimizer steps. We track an exponential moving average of the parameters with decay 0.9999 and apply gradient clipping at ℓ_2-norm 3.0. Mixed-precision training uses `bf16` autocast with TF32 enabled and `torch.compile` (inductor backend). The training objective is the per-coordinate Gaussian-NLL of (6), with the time-importance weight $1/(1-t)$ clamped to $[1, 256]$ to stabilize the boundary contribution.

The DiT-B ablations (A1/A2/A3) inherit the same optimizer family, weight decay, gradient clipping, EMA decay, and `bf16` precision, scaled to the smaller backbone; per-ablation training durations and the frozen-vs.-joint training protocol are stated in Section 5.3.

B.6 Compute Resources

All training and evaluation runs use NVIDIA H100 GPUs (80 GB). The main DiT-XL training run consumes $\sim 2{,}000$ H100-hours in total. The 1024 steps Hutchinson reference takes ~ 100 H100-hours. Likelihood-evaluation compute on the $50{,}000$-image ImageNet-256 validation split is reported in the left panel of Figure 4 as total GPU-hours per method/step configuration. FID generation produces $50{,}000$ class-balanced samples at 64 Euler steps per ablation cell; sampling and reference-FID computation are themselves a small fraction of the per-cell training cost. The DiT-B ablation runs are substantially smaller-scale than the main run.

B.7 Likelihood Evaluation Protocol

For completeness we restate the evaluation protocol used in Section 5.2 and reused throughout the ablations. Both COAL and the Hutchinson baseline integrate the divergence on a uniform Euler grid from $t{=}1$ to $t{=}0$ with the same deterministic initial state $z_{t=1}$ given by the encoded latent. The velocity backbone runs under `bf16` autocast, while the log-density integral is accumulated in `fp32` to avoid catastrophic cancellation at large N. COAL is fully deterministic; the Hutchinson estimator redraws fresh Rademacher probe vectors per Euler step. Boundary-time stability at $t{=}1$ is handled architecturally rather than by clamping: the $(1-t)\,s_\theta$ reparametrization of $\log \sigma_\theta^2$ together with the $\mathrm{expm1}$ form of the divergence integrand cancels the apparent singularity exactly. Evaluation is distributed over 4 GPUs at each method's memory-saturating batch size.

B.8 FID-50k Evaluation Protocol

FID-50k in Table 1b is computed against precomputed ImageNet-256 reference statistics. For each ablation cell we generate $50{,}000$ samples balanced across the 1000 ImageNet classes (50 per class). Sampling uses the same Euler integrator as the velocity training, at 64 steps per sample, with classifier-free guidance (Ho & Salimans, 2022) scale 1.5 (the cond/uncond pair runs jointly when CFG is on). The latent-noise seed is fixed across all cells so that any FID difference reflects the trained model rather than sampler variance. Generated latents are decoded through the MAR-VAE used during training (latent scale 0.2325) to RGB at 256×256, then scored with `torch_fidelity`.

B.9 Licenses for External Assets

The two external assets that materially shape our experimental setup are the ImageNet dataset and the MAR-VAE latent autoencoder; we summarize each license below.

ImageNet (ILSVRC-2012). ImageNet is distributed under a custom Terms of Access maintained by Princeton University and Stanford University.[1] The agreement grants use of the database to the registered researcher only for *non-commercial research and educational purposes*; for-profit researchers are bound by the same terms via their employer. The agreement disclaims warranties and places liability for downstream use on the researcher. Our use is non-commercial academic research, consistent with these terms.

MAR-VAE. We use the latent autoencoder released with the MAR codebase (Li et al., 2024). The reference implementation and released checkpoints are distributed under the *MIT License*.[2]

[1] `https://image-net.org/accessagreement`

[2] `https://github.com/LTH14/mar/blob/main/LICENSE`