Proceedings of Machine Learning Research 266:1–3, 2025 Conformal and Probabilistic Prediction with Applications

FlowGuard: Guarding Flow Matching via Conformal Sampling

Ziyun Li ZIYLI@KTH.SE
Henrik Boström BOSTROMH@KTH.SE
KTH Royal Institute of Technology, STOCKHOLM, SWEDEN

Editor: Khuong An Nguyen, Zhiyuan Luo, Tuwe Löfström, Lars Carlsson and Henrik Boström

1. Introduction

Iterative denoising models, diffusion (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021) and flow matching (FM) (Lipman et al., 2022) achieve state-of-the-art generative performance across e.g., images, video, audio, and protein design. They learn a time-dependent velocity field that maps Gaussian noise to the data manifold, but sampling remains costly: generating a single high-quality output typically requires tens to hundreds of forward passes, limiting real-time and resource-constrained use.

Multiple research strands aim to reduce the high inference cost of generative models. A prominent approach is to train the velocity field such that trajectories between noise and data approximate straight Euclidean paths, enabling larger ODE steps with lower numerical error. Methods like Rectified Flow (Liu et al., 2022), LOOM-CFM (Park et al., 2023), Optimal Flow Matching (Huang et al., 2023), and Consistency Models (Song et al., 2024) adopt this strategy, achieving competitive image quality with significantly fewer function evaluations. While these techniques improve overall performance and stabilize sampling globally, they overlook sample-level reliability. As a result, individual trajectories can still fail, hallucinating object parts or ignoring conditioning, requiring repeated sampling to obtain a usable output.

We address this limitation with *FlowGuard*, a lightweight conformal mechanism (Vovk et al., 2022; Linusson et al., 2017; Vovk et al., 2005; Angelopoulos and Bates, 2021) that enforces per-sample reliability during inference. *FlowGuard* monitors each trajectory's curvature by computing its maximum velocity jump and rejects those exceeding a conformal threshold calibrated on held-out data. This enables early termination of low-quality generations, improving runtime efficiency without compromising model integrity. Crucially, *FlowGuard* is model-agnostic and can be seamlessly applied to any iterative-based generative model. Empirically, *FlowGuard* reduces FID by 2.3% at 100 number of function evaluations (NFEs) and 2.8% at 30 NFEs with no added model cost. The code is released at `https://github.com/ziyunli-2023/flowguard-clean`.

2. FlowGuard

Conditional Flow Matching (CFM) Sampler. Let $x_1 \sim q$ be a clean data sample and $x_0 \sim \mathcal{N}(0, I)$ its paired noise. K represents the number of function evaluations. CFM defines a simple conditional path $p_t(x \mid x_1)$ (e.g., linear interpolation with noise) with known drift $u_t(x \mid x_1)$, and trains a neural field $v_\theta(x, t)$ to match it via

$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t, x_1, x \sim p_t(\cdot|x_1)} \|v_\theta(x, t) - u_t(x \mid x_1)\|_2^2$. At sampling time, we draw $x_0 \sim \mathcal{N}(0, I)$ and generate a sample by integrating

$$x_{k+1} = x_k + v_\theta(x_k, t_k)\,\Delta t, \quad \text{with } \Delta t = 1/K.$$

Calibration. We draw n data samples $x_1^{(1)}, \ldots, x_1^{(n)} \sim q$ from a held-out calibration set. For each $x_1^{(i)}$, we run the conditional sampler to generate a discrete trajectory $\{x_k^{(i)}\}_{k=0}^K$, where each $x_k^{(i)}$ corresponds to a solver step at time $t_k = k \cdot \Delta t$ with step size $\Delta t = 1/K$. The trajectory begins from $x_0^{(i)} \sim \mathcal{N}(0, I)$ and is integrated forward using Euler steps. Under Euler sampling, the ideal flow-matching velocity grows as $1/(1-t)^2$, we multiply each velocity gap by a normalization factor $(1-t_k)^2/\Delta t$, which cancels this deterministic growth. This keeps the score stable for well-behaved trajectories while highlighting true divergences. The calibrated score for each sample is then computed as:

$$S_i = \max_{k=0,\ldots,K-1} \frac{(1-t_k)^2}{\Delta t} \cdot \left\| v_\theta(x_{k+1}^{(i)}, t_{k+1}) - v_\theta(x_k^{(i)}, t_k) \right\|_2^2. \tag{1}$$

The conformal threshold τ_α is set as the $(1-\alpha)$ upper order statistic:

$$\tau_\alpha = S_{(\lceil (1-\alpha)(n+1) \rceil)}. \tag{2}$$

Conformal Sampler. For each sampling trajectory, we track the velocity-jump scores $s_k = \frac{(1-t_k)^2}{\Delta t} \cdot \|v_\theta(x_{k+1}, t_{k+1}) - v_\theta(x_k, t_k)\|_2^2$ and define the trajectory score as $S_{\text{traj}} = \max_k s_k$. If $S_{\text{traj}} > \tau_\alpha$, the trajectory is rejected early; otherwise, x_1 is accepted. Because calibration and test trajectories are exchangeable, the split conformal threshold τ_α ensures $\Pr[S_{\text{test}} \leq \tau_\alpha] \geq 1 - \alpha$. *This enables early termination of unreliable generations, improving runtime efficiency without extra model evaluations.*

3. Experiments

Experiment Setup. We evaluate the effectiveness of our method on the CIFAR-10 dataset (32×32 resolution) using a pretrained OTCFM model on CIFAR-10 provided by TorchCFM (Duan et al., 2023). Euler sampler uses both 100 and 30 function evaluations (NFE), covering both high-fidelity and fast-generation regimes. The conformal threshold τ_α is calibrated on $n = 2000$ held-out trajectories.

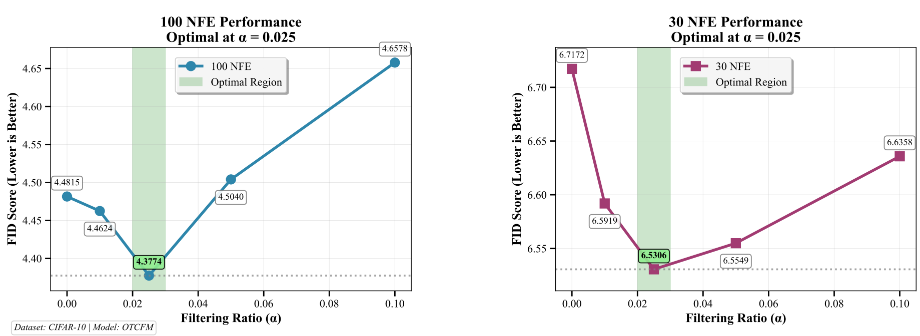

Figure 1: FID vs. filtering ratio α under 100 and 30 Euler steps. Filtering high-jump trajectories improves quality across both settings.

Results. Figure 1 plots Fréchet Inception Distance (FID) against filtering ratio α for 100- and 30-NFEs sampling. Conformal filtering consistently improves quality, reducing FID by 2.3% at 100 steps and 2.8% at 30 steps without any added model cost.

Acknowledgments

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

References

Anastasios N Angelopoulos and Stephen Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*, 2021.

Tony Duan, Atong Song, et al. Torchcfm: Conditional flow matching in pytorch. https://github.com/atong01/conditional-flow-matching, 2023. Accessed: 2025-07-05.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

Xingqian Huang, Zhendong Zhang, Duygu Ceylan, Xinyang Chen, Yang Liu, and Junbo Zhu. Optimal flow matching: Fast and high-quality generative modeling with one forward pass. In *NeurIPS*, 2023.

Henrik Linusson, Ulf Norinder, Henrik Boström, Ulf Johansson, and Tuve Löfström. On the calibration of aggregated conformal predictors. In *Conformal and probabilistic prediction and applications*, pages 154–173. PMLR, 2017.

Yotam Lipman, Federico Alberti, Filippo Specogna, Andrea Tagliasacchi, and Maria Vakalopoulou. Flow matching for generative modeling. In *NeurIPS*, 2022.

Zhijie Liu, Han Zhang, Xuejun Li, Junbo Zhu, and Yang Liu. Rectified flow: A more stable and sample-efficient approach to score-based diffusion models. In *NeurIPS*, 2022.

Jaesung Park, Junbum Jeong, Jascha Sohl-Dickstein, and Yang Song. Learning optimal transport maps via conditional flow matching. In *ICLR*, 2023.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015.

Yang Song, Jascha Sohl-Dickstein, Diederik Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021.

Yang Song, Chenlin Meng, Ruixiang Tan, Jiaming Zhao, Stefano Ermon, and Jascha Sohl-Dickstein. Consistency models. In *ICLR*, 2024.

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, 2005. Foundational work on conformal prediction.

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World (2nd ed.)*. Springer, 2022.