Reflected Flow Matching

Tianyu Xie[*1] **Yu Zhu**[*2 3] **Longlin Yu**[*1] **Tong Yang**[4 5] **Ziheng Cheng**[1]
Shiyue Zhang[1] **Xiangyu Zhang**[5] **Cheng Zhang**[1 6]

Abstract

Continuous normalizing flows (CNFs) learn an ordinary differential equation to transform prior samples into data. Flow matching (FM) has recently emerged as a simulation-free approach for training CNFs by regressing a velocity model towards the conditional velocity field. However, on constrained domains, the learned velocity model may lead to undesirable flows that result in highly unnatural samples, e.g., oversaturated images, due to both flow matching error and simulation error. To address this, we add a boundary constraint term to CNFs, which leads to reflected CNFs that keep trajectories within the constrained domains. We propose reflected flow matching (RFM) to train the velocity model in reflected CNFs by matching the conditional velocity fields in a simulation-free manner, similar to the vanilla FM. Moreover, the analytical form of conditional velocity fields in RFM avoids potentially biased approximations, making it superior to existing score-based generative models on constrained domains. We demonstrate that RFM achieves comparable or better results on standard image benchmarks and produces high-quality class-conditioned samples under high guidance weight.

1. Introduction

Deep generative models, which are deep learning models designed to generate new data samples that resemble a given data set, find applications in various domains including image synthesis (Arjovsky et al., 2017; Ho et al., 2020; Dhariwal & Nichol, 2021; Song et al., 2023), text generation (Devlin et al., 2018; Brown et al., 2020; Zhang et al., 2022), and molecular design (Jin et al., 2018; Xu et al., 2021; Hoogeboom et al., 2022). While the recent advances in this domain are mostly driven by diffusion models (Ho et al., 2020; Song et al., 2020b), a powerful alternative is flow-based model (Rezende & Mohamed, 2015), which works by learning a *flow*, i.e., a function that transforms samples from a simple prior distribution into samples distributed similarly to the target data distribution. Flows can also be implicitly defined via ordinary differential equations (ODEs), which lead to a general framework called continuous normalizing flows (CNFs) (Chen et al., 2018). Despite their flexibility, classical maximum likelihood training of CNFs can be inefficient as it requires expensive numerical ODE simulations.

Alternatively, flow matching (FM) (Lipman et al., 2023) presents an efficient and scalable training method for CNFs by regressing a learnable velocity model to the target velocity field (i.e., the drift term in ODE) that generates the data distribution. Inspired by denoising score matching (DSM) (Vincent, 2011), FM decomposes the intractable target velocity field into a mixture of analytical conditional velocity fields, leading to a regression objective that is computationally stable and simulation-free. A typical example of conditional velocity fields is the optimal transport (OT) conditional velocity field, which has a constant speed along the straight line from a prior sample to a data point (i.e., its condition). FM has also been extended to equivariance modeling (Klein et al., 2023), Bayesian inference (Wildberger et al., 2023), and dynamic optimal transport (Tong et al., 2023), etc.

Despite the superiority of FM, the learned velocity model may cause unreasonable flows, due to both the inherent flow matching error (i.e., the difference between the learned velocity model and the true velocity field) and compounded simulation error (i.e., the error introduced by discretizing the ODE during sampling), which can generate highly unnatural samples, especially for complex data distributions on constrained domains. For example, an RGB image is constrained to a domain $[0, 255]^d$ (d is the number of digits), and a boundary violation may result in collapsed and

[*]Equal contribution. This work was done during an internship at Megvii Technology Inc. [1]School of Mathematical Sciences, Peking University, Beijing, China [2]Institute of Automation, Chinese Academy of Sciences, Beijing, China [3]Beijing Academy of Artificial Intelligence, Beijing, China [4]School of Computer Science, Fudan University, Shanghai, China [5]Megvii Technology Inc., Beijing, China [6]Center for Statistical Science, Peking University, Beijing, China. Correspondence to: Cheng Zhang <chengzhang@math.pku.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

singular samples. Recently, there has been a growing interest in generative modeling on constrained domains, driven by its broad range of applications (Fishman et al., 2023; Lou & Ermon, 2023; Liu et al., 2023). Within the framework of score-based generative models (Song et al., 2020b; Ho et al., 2020), reflected diffusion models (RDMs) (Lou & Ermon, 2023) incorporate the constraints through a reflected stochastic differential equation (SDE) that always stays within the domain. The reverse diffusion process is then parameterized with the scores of the perturbed intermediate distributions, which can be trained via DSM. However, the transition kernels in RDMs lack a closed-form expression, necessitating the complicated conditional score approximation that requires computing reflecting points (Jing et al., 2022b) or solving partial differential equations (Bortoli et al., 2022). These challenges introduce bias and make the application of RDMs difficult for general domains, limiting their practicality to simple constrained domains like hypercubes. Although implicit score matching (ISM) avoids the computation of transition kernels (Fishman et al., 2023), it is less stable than DSM and does not scale well to high-dimensional cases.

In this work, we propose reflected flow matching (RFM), which extends the training and sampling procedures of CNFs to constrained domains. To achieve this, we add an additional boundary constraint term to the ODEs in CNFs, which we call reflected CNFs. When initiated from prior distributions supported on the target constrained domain, the reflected CNFs ensure that the trajectories remain within this domain. The velocity fields in the reflected CNFs can be trained by matching the conditional velocity fields in a simulation-free manner, akin to vanilla FM (Lipman et al., 2023). Moreover, these conditional velocity fields can be readily derived from corresponding conditional flows that satisfy the constraints. This means that, unlike RDMs (Lou & Ermon, 2023), FM can be smoothly generalized to constrained domains with no extra computational burden. Moreover, inherited from FM, the RFM framework permits the choice of arbitrary prior distributions on the constrained domain, which may provide extra flexibility for generative modeling. We demonstrate the effectiveness of RFM across various generative tasks, including low-dimensional toy examples as well as unconditional and conditional image generation benchmarks. Our code is available at `https://github.com/tyuxie/RFM`.

2. Background

2.1. Flow Matching

A time-dependent *flow* $\boldsymbol{x}_t = \boldsymbol{\phi}_t(\boldsymbol{x})$, which describes the motion from a base point $\boldsymbol{x}$, can be defined by an ordinary differential equation (ODE)

$$\mathrm{d}\boldsymbol{\phi}_t(\boldsymbol{x}) = \boldsymbol{v}_t(\boldsymbol{\phi}_t(\boldsymbol{x}))\mathrm{d}t, \tag{1a}$$

$$\boldsymbol{\phi}_0(\boldsymbol{x}) = \boldsymbol{x}, \tag{1b}$$

for $t \in [0,1]$, where $\boldsymbol{v}_t(\cdot) \in \mathbb{R}^d$ is called the *velocity field*. Given a density $p_0(\boldsymbol{x})$ at $t=0$, the velocity field induces a *probability path* $p_t(\boldsymbol{x}) : [0,1] \times \mathbb{R}^d \to \mathbb{R}$, where p_t is the probability density function (PDF) of the points from $p_0(\boldsymbol{x})$ transported along $\boldsymbol{v}_t$ from time 0 to time t, and it is characterized by the well-known continuity equation:

$$\frac{\partial}{\partial t} p_t(\boldsymbol{x}) = -\nabla \cdot (p_t(\boldsymbol{x})\boldsymbol{v}_t(\boldsymbol{x})),\ \boldsymbol{x} \in \mathbb{R}^d. \tag{2}$$

Chen et al. (2018) models the velocity field by a neural network $\boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}, t)$ with parameters $\boldsymbol{\theta}$ to generate a learnable model of $\boldsymbol{\phi}_t$, called a continuous normalizing flow (CNF). Given a target probability path $p_t(\boldsymbol{x})$ that connects a simple prior distribution $p_0(\boldsymbol{x})$ and a complicated target distribution $p_1(\boldsymbol{x})$ which is close to the data distribution $p_{\mathrm{data}}(\boldsymbol{x})$, the main idea of flow matching (FM) (Lipman et al., 2023) is to regress the velocity model $\boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}, t)$ to the target velocity field $\boldsymbol{v}_t(\boldsymbol{x})$ by minimizing the FM objective

$$\mathcal{L}_{\mathrm{FM}}(\boldsymbol{\theta}) = \int_0^1 \mathbb{E}_{p_t(\boldsymbol{x})} \|v_{\boldsymbol{\theta}}(\boldsymbol{x}, t) - \boldsymbol{v}_t(\boldsymbol{x})\|^2 \mathrm{d}t. \tag{3}$$

One way to construct such a target probability path is to create per-sample *conditional probability paths* $p_t(\boldsymbol{x}|\boldsymbol{x}_1)$ that satisfy $p_0(\boldsymbol{x}|\boldsymbol{x}_1) = p_0(\boldsymbol{x}), p_1(\boldsymbol{x}|\boldsymbol{x}_1) \approx \delta(\boldsymbol{x}_1)$, and marginalize them over the data distribution as follows

$$p_t(\boldsymbol{x}) = \int_{\mathbb{R}^d} p_t(\boldsymbol{x}|\boldsymbol{x}_1) p_{\mathrm{data}}(\boldsymbol{x}_1) \mathrm{d}\boldsymbol{x}_1. \tag{4}$$

Lipman et al. (2023) then shows that the corresponding velocity field $\boldsymbol{v}_t(\boldsymbol{x})$ takes the following form

$$\boldsymbol{v}_t(\boldsymbol{x}) = \int_{\mathbb{R}^d} \boldsymbol{v}_t(\boldsymbol{x}|\boldsymbol{x}_1) \frac{p_t(\boldsymbol{x}|\boldsymbol{x}_1) p_{\mathrm{data}}(\boldsymbol{x}_1)}{p_t(\boldsymbol{x})} \mathrm{d}\boldsymbol{x}_1, \tag{5}$$

where $\boldsymbol{v}_t(\boldsymbol{x}|\boldsymbol{x}_1)$ is the conditional velocity field that generates $p_t(\boldsymbol{x}|\boldsymbol{x}_1)$. The velocity model $\boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}, t)$, therefore, can be trained by minimizing the conditional flow matching (CFM) objective

$$\mathcal{L}_{\mathrm{CFM}}(\boldsymbol{\theta}) = \int_0^1 \mathbb{E}_{p_t(\boldsymbol{x}|\boldsymbol{x}_1) p_{\mathrm{data}}(\boldsymbol{x}_1)} \|\boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}, t) - \boldsymbol{v}_t(\boldsymbol{x}|\boldsymbol{x}_1)\|^2 \mathrm{d}t. \tag{6}$$

Concretely, Lipman et al. (2023) considers the Gaussian conditional probability paths, which can be induced by a flow conditioned on $\boldsymbol{x}_1$ as

$$\boldsymbol{\phi}_t(\boldsymbol{x}|\boldsymbol{x}_1) = \sigma_t(\boldsymbol{x}_1)\boldsymbol{x} + \boldsymbol{\mu}_t(\boldsymbol{x}_1), \tag{7}$$

where $\boldsymbol{x}$ follows a standard Gaussian distribution. Consequently, the conditional velocity field takes the form

$$\boldsymbol{v}_t(\boldsymbol{x}|\boldsymbol{x}_1) = \frac{\sigma_t'(\boldsymbol{x}_1)}{\sigma_t(\boldsymbol{x}_1)}\left(\boldsymbol{x} - \boldsymbol{\mu}_t(\boldsymbol{x}_1)\right) + \boldsymbol{\mu}_t'(\boldsymbol{x}_1). \quad (8)$$

A typical example is the optimal transport (OT) conditional velocity field, where $\sigma_t(\boldsymbol{x}_1) = 1 - (1 - \sigma_{\min})t$, $\boldsymbol{\mu}_t(\boldsymbol{x}_1) = t\boldsymbol{x}_1$, and $\sigma_{\min}$ is a sufficiently small value.

2.2. Reflected Diffusion Models

For generative modeling on constrained domains, reflected diffusion models (RDMs) (Lou & Ermon, 2023) use reflected stochastic differential equations (SDEs) that perform reflection operations at the boundary to keep samples inside the target domain. Let the support of the data distribution $q_0(\cdot)$ be Ω, which is assumed to be connected and compact with a nonempty interior. Concretely, the forward process in an RDM is described by the following reflected SDE

$$\mathrm{d}\boldsymbol{u}_t = \boldsymbol{f}(\boldsymbol{u}_t, t)\mathrm{d}t + g(t)\mathrm{d}\boldsymbol{B}_t + \mathrm{d}\boldsymbol{L}_t,\ \boldsymbol{u}_0 \sim q_0(\cdot), \quad (9)$$

where $t \in [0, L]$, $\boldsymbol{f}(\boldsymbol{u}_t, t)$ and $g(t)$ are the drift term and diffusion term as in the classical diffusion models (Song et al., 2020b), and $\boldsymbol{L}_t$ is an additional boundary constraint that intuitively forces the particle to stay inside Ω. When $\boldsymbol{u}_t$ hits the boundary $\partial\Omega$, $\boldsymbol{L}_t$ neutralizes the outward-normal velocity. To generate samples from the data distribution, one can simulate the reverse reflected SDE (Cattiaux, 1988; Williams, 1987)

$$\begin{aligned}\mathrm{d}\boldsymbol{u}_t = &\left[\boldsymbol{f}(\boldsymbol{u}_t, t) - \frac{1}{2}(1+\lambda^2)g^2(t)\nabla \log q_t(\boldsymbol{u}_t)\right]\mathrm{d}t \\ &+\lambda g(t)\mathrm{d}\bar{\boldsymbol{B}}_t + \mathrm{d}\boldsymbol{L}_t,\ \boldsymbol{u}_L \sim q_L(\cdot),\end{aligned} \quad (10)$$

where $q_L(\cdot) = \mathrm{uniform}(\Omega)$ under some special conditions, $\bar{\boldsymbol{B}}_t$ is a standard Brownian motion when time flows back from L to 0 and $\nabla \log q_t(\boldsymbol{u}_t)$ is the score function at time t.

Although denoising score matching (DSM) (Vincent, 2011; Song et al., 2020b) succeeds in estimating the intractable score function $\nabla \log q_t(\boldsymbol{u}_t)$, the required conditional score function $\nabla_{\boldsymbol{u}_t} \log q_t(\boldsymbol{u}_t|\boldsymbol{u}_0)$ is no longer analytically available for the forward process in equation (9). To estimate $\nabla_{\boldsymbol{u}_t} \log q_t(\boldsymbol{u}_t|\boldsymbol{u}_0)$, RDMs rely on approximation methods based on sum of Gaussians (Jing et al., 2022b) or Laplacian eigenfunctions (Bortoli et al., 2022). However, both of them can be domain-dependent and require truncating an infinite series of functions. Besides, although the noisy training data in DSM can be obtained in a simulation-free manner, the required geometric techniques may be restrictive in practice.

2.3. Classifier-free Guidance

Diffusion models can be controlled to generate guided samples from $\tilde{q}_t(\boldsymbol{u}|c) \propto q_t(c|\boldsymbol{u})^w q_t(\boldsymbol{u})$ where c is a condition (e.g., a class label) and w is the guidance weight (Ho & Salimans, 2022). The score function of $\tilde{q}_t(\boldsymbol{u}|c)$ satisfies

$$\nabla \log \tilde{q}_t(\boldsymbol{u}|c) = w\nabla \log q_t(c|\boldsymbol{u}) + \nabla \log q_t(\boldsymbol{u}). \quad (11)$$

With the Bayes formula $q_t(c|\boldsymbol{u}) = \frac{q_t(\boldsymbol{u}|c)q_t(c)}{q_t(\boldsymbol{u})}$ and a conditional score model $\boldsymbol{s}_\phi(\boldsymbol{u}, t, c) \approx \nabla \log q_t(\boldsymbol{u}|c)$, the score function of $\tilde{q}_t(\boldsymbol{u}|c)$ can be estimated by

$$\tilde{\boldsymbol{s}}_\phi(\boldsymbol{u}, t, c) = w\boldsymbol{s}_\phi(\boldsymbol{u}, t, c) + (1-w)\boldsymbol{s}_\phi(\boldsymbol{u}, t, \emptyset), \quad (12)$$

where $\boldsymbol{s}_\phi(\boldsymbol{u}, t, \emptyset) = \boldsymbol{s}_{\boldsymbol{\phi}}(\boldsymbol{u}, t) \approx \nabla \log q_t(\boldsymbol{u})$ denotes the unconditional score with c set to the empty token. Equation (12) also holds for RDMs, and one can directly substitute the score function in equation (10) with $\tilde{\boldsymbol{s}}_\phi(\boldsymbol{u}, t, c)$ to generate guided samples (Lou & Ermon, 2023).

3. Proposed Method

In this section, we present reflected flow matching (RFM), which learns a flow-based generative model for the data distribution $p_{\mathrm{data}}(\cdot)$ supported on a constrained domain $\Omega \subset \mathbb{R}^d$. The data domain Ω is assumed to be connected and compact with a non-empty interior Ω^o and a sufficiently regular boundary $\partial\Omega$.

3.1. Reflected Ordinary Differential Equation

To model flows over the constrained domain Ω, we add an additional term for the boundary constraint to the vanilla ODE (1), inspired by Lou & Ermon (2023). Concretely, let $\boldsymbol{L}_t$ reflect the outward normal direction at $\partial\Omega$. Given a initial point $\boldsymbol{x}$, our *reflected ODE* for describing a time-dependent flow $\boldsymbol{x}_t = \boldsymbol{\phi}_t(\boldsymbol{x})$ takes the form

$$\mathrm{d}\boldsymbol{\phi}_t(\boldsymbol{x}) = \boldsymbol{v}_t(\boldsymbol{\phi}_t(\boldsymbol{x}))\mathrm{d}t + \mathrm{d}\boldsymbol{L}_t, \quad (13a)$$

$$\boldsymbol{\phi}_0(\boldsymbol{x}) = \boldsymbol{x}, \quad (13b)$$

where $\boldsymbol{v}_t(\cdot) : \Omega \to \mathbb{R}^d$ defines velocity field over Ω for each time $t \in [0, 1]$. In equation (13), the velocity term $\boldsymbol{v}_t(\boldsymbol{\phi}_t(\boldsymbol{x}))$ describes the motion of particles in Ω^o just as in the vanilla ODE (1), and the reflection term $\mathrm{d}\boldsymbol{L}_t$ compensates the outward velocity at $\partial\Omega$ by pushing the motion back to the domain Ω. That is, upon the motion's hitting the boundary $\partial\Omega$ at $\boldsymbol{x}_t$, $\mathrm{d}\boldsymbol{L}_t$ neutralizes the outward normal-pointing component. We give the following theorem for the existence and uniqueness of the solution to the reflected ODE (13), which is a corollary of Pilipenko (2014, Theorem 2.5.4) (see Appendix A.1 for details).

Theorem 3.1. *Assume i) the domain Ω satisfies the uniform exterior sphere condition and the uniform cone condition; ii) the velocity field $\boldsymbol{v}_t(\boldsymbol{x})$ is Lipschitz continuous in $\boldsymbol{x} \in \Omega$ (uniformly on t). Then the solution to the reflected ODE (13) exists and is unique on $t \in [0, 1]$.*

Remark 3.2. The reflected ODE (13) recovers the vanilla ODE (1) in flow matching by taking $\Omega = \mathbb{R}^d$. In this case, the reflection term $\mathrm{d}\boldsymbol{L}_t$ disappears because the motion will never hit the boundary.

With the initial point $\boldsymbol{x}$ distributed as a simple prior distribution $p_0(\cdot)$ supported on Ω, the PDF $p_t(\cdot)$ of the flow $\boldsymbol{\phi}_t(\boldsymbol{x})$ can also be used to construct a probability path which connects $p_0(\cdot)$ and a target distribution $p_1(\cdot)$ that closely approximates the data distribution $p_{\text{data}}(\cdot)$. Moreover, due to the reflection term $\mathrm{d}\boldsymbol{L}_t$, the probability path $p_t(\boldsymbol{x})$ is naturally supported on Ω and evolves according to the following continuity equation with the Neumann boundary condition[1] (Schuss, 2015)

$$\frac{\partial}{\partial t}p_t(\boldsymbol{x}) = -\nabla \cdot (\boldsymbol{v}_t(\boldsymbol{x})p_t(\boldsymbol{x})), \boldsymbol{x} \in \Omega^o, \tag{14a}$$

$$p_t(\boldsymbol{x})\boldsymbol{v}_t(\boldsymbol{x}) \cdot \boldsymbol{n}(\boldsymbol{x}) = 0, \boldsymbol{x} \in \partial\Omega, \tag{14b}$$

where $\boldsymbol{n}(\boldsymbol{x})$ is the outward normal vector at $\boldsymbol{x}$ on $\partial\Omega$.

3.2. Reflected Flow Matching

Similarly to Lipman et al. (2023), the basic idea of RFM is to learn a parametrized velocity model $\boldsymbol{v}_\theta(\boldsymbol{x}, t)$, which leads to a deep model of $\boldsymbol{\phi}_t$ called *reflected CNF*, by minimizing the following RFM loss

$$\mathcal{L}_{\text{RFM}}(\boldsymbol{\theta}) = \int_0^1 \mathbb{E}_{p_t(\boldsymbol{x})}\|\boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}, t) - \boldsymbol{v}_t(\boldsymbol{x})\|^2 \mathrm{d}t, \tag{15}$$

given a target probability path $p_t(\boldsymbol{x})$ and a corresponding velocity field $\boldsymbol{v}_t(\boldsymbol{x})$. We can construct such a probability path and velocity field as follows. Let $p_t(\boldsymbol{x}|\boldsymbol{x}_1)$ be a conditional probability path satisfying $p_0(\boldsymbol{x}|\boldsymbol{x}_1) = p_0(\boldsymbol{x}), p_1(\boldsymbol{x}|\boldsymbol{x}_1) \approx \delta(\boldsymbol{x}_1)$. The marginal probability path then takes the form

$$p_t(\boldsymbol{x}) = \int_\Omega p_t(\boldsymbol{x}|\boldsymbol{x}_1)p_{\text{data}}(\boldsymbol{x}_1)\mathrm{d}\boldsymbol{x}_1. \tag{16}$$

Let $\boldsymbol{v}_t(\boldsymbol{x}|\boldsymbol{x}_1)$ be a conditional velocity field that generates the conditional probability path $p_t(\boldsymbol{x}|\boldsymbol{x}_1)$ and defines the marginal velocity field

$$\boldsymbol{v}_t(\boldsymbol{x}) = \int_\Omega \boldsymbol{v}_t(\boldsymbol{x}|\boldsymbol{x}_1)\frac{p_t(\boldsymbol{x}|\boldsymbol{x}_1)p_{\text{data}}(\boldsymbol{x}_1)}{p_t(\boldsymbol{x})}\mathrm{d}\boldsymbol{x}_1. \tag{17}$$

Note that we require $\operatorname{supp}(p_t(\boldsymbol{x}|\boldsymbol{x}_1)) \subset \Omega$ for all t and $\boldsymbol{x}_1$, in contrast to the vanilla FM (Lipman et al., 2023). In fact, the velocity field $\boldsymbol{v}_t(\boldsymbol{x})$ in equation (17) exactly generates the probability path $p_t(\boldsymbol{x})$ in equation (16), which is formalized in Theorem 3.3.

[1]Informally, when a particle hits the boundary, the outward-normal part of its velocity would be neutralized by the reflection term $\mathrm{d}L_t$, which would make the velocity satisfy the Neumann boundary condition.

Theorem 3.3. *Assume* $\operatorname{supp}(p_t(\boldsymbol{x}|\boldsymbol{x}_1)) \subset \Omega$ *for all* t *and* $\boldsymbol{x}_1$. *For any target distribution* $p_1(\boldsymbol{x}_1)$, *if the conditional probability path* $p_t(\boldsymbol{x}|\boldsymbol{x}_1)$ *and conditional velocity field* $\boldsymbol{v}_t(\boldsymbol{x}|\boldsymbol{x}_1)$ *satisfy the continuity equation* (14), *then marginal probability path* $p_t(\boldsymbol{x})$ *and the marginal velocity field* $\boldsymbol{v}_t(\boldsymbol{x})$ *also satisfy the continuity equation* (14).

Proof. The fact that $p_t(\boldsymbol{x})$ and $\boldsymbol{v}_t(\boldsymbol{x})$ satisfy equation (14a) is from Lipman et al. (2023, Theorem 1). To prove $p_t(\boldsymbol{x})$ and $\boldsymbol{v}_t(\boldsymbol{x})$ satisfy the Neumann boundary condition (14b), it suffices to combine

$$p_t(\boldsymbol{x})\boldsymbol{v}_t(\boldsymbol{x}) = \int_\Omega \boldsymbol{v}_t(\boldsymbol{x}|\boldsymbol{x}_1)p_t(\boldsymbol{x}|\boldsymbol{x}_1)p_{\text{data}}(\boldsymbol{x}_1)\mathrm{d}\boldsymbol{x}_1$$

and $p_t(\boldsymbol{x}|\boldsymbol{x}_1)\boldsymbol{v}_t(\boldsymbol{x}|\boldsymbol{x}_1) \cdot \boldsymbol{n}(\boldsymbol{x}) = 0$ for $\boldsymbol{x} \in \partial\Omega$. □

Similarly to the CFM objective in equation (6), the velocity model $\boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}, t)$ in reflected CNFs can be trained by optimizing the following conditional reflected flow matching (CRFM) objective

$$\mathcal{L}_{\text{CRFM}}(\boldsymbol{\theta}) = \int_0^1 \mathbb{E}_{p_t(\boldsymbol{x}|\boldsymbol{x}_1)p_{\text{data}}(\boldsymbol{x}_1)}\|\boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}, t) - \boldsymbol{v}_t(\boldsymbol{x}|\boldsymbol{x}_1)\|^2 \mathrm{d}t, \tag{18}$$

as proved in Theorem 3.4 (see Appendix A.2 for proof).

Theorem 3.4. *Assume that* $p_t(\boldsymbol{x}) > 0$ *for all* $\boldsymbol{x} \in \Omega^o$ *and* $t \in [0, 1]$. *Then* $\nabla\mathcal{L}_{RFM}(\boldsymbol{\theta}) = \nabla\mathcal{L}_{CRFM}(\boldsymbol{\theta})$.

Moreover, we prove the following Wasserstein bound for the reflected CNFs trained with RFM (see Appendix A.3 for proof), similar to Albergo & Vanden-Eijnden (2022, Proposition 3).

Theorem 3.5 (Wasserstein Bound)**.** *Assume* Ω *is convex and the velocity model* $\boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}, t)$ *is* M*-Lipschitz in* $\boldsymbol{x}$ *(uniformly on* t*). Let* $p_{\boldsymbol{\theta},t}(\boldsymbol{x})$ *be the probability path of the reflected CNF induced by* $\boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}, t)$ *starting from the same prior distribution* $p_0(\boldsymbol{x})$. *Then the squared Wasserstein-2 distance between* $p_1(\boldsymbol{x})$ *and* $p_{\boldsymbol{\theta},1}(\boldsymbol{x})$ *is bounded by*

$$W_2^2(p_1(\boldsymbol{x}), p_{\boldsymbol{\theta},1}(\boldsymbol{x})) \leq e^{1+2M}\mathcal{L}_{RFM}(\boldsymbol{\theta}). \tag{19}$$

3.3. Construction of Conditional Velocity Fields

One remaining thing is the choice of the conditional probability path $p_t(\boldsymbol{x}|\boldsymbol{x}_1)$ and the conditional velocity field $\boldsymbol{v}_t(\boldsymbol{x}|\boldsymbol{x}_1)$ on Ω. Based on reflected ODEs, both of them can be simply derived from a conditional flow $\boldsymbol{\phi}_t(\boldsymbol{x}|\boldsymbol{x}_1)$, which satisfies that $\boldsymbol{\phi}_0(\boldsymbol{x}|\boldsymbol{x}_1) = \boldsymbol{x}$, $\boldsymbol{\phi}_1(\boldsymbol{x}|\boldsymbol{x}_1) \approx \boldsymbol{x}_1$, and $\boldsymbol{\phi}_t(\boldsymbol{x}|\boldsymbol{x}_1) \in \Omega$ for all $\boldsymbol{x}, \boldsymbol{x}_1 \in \Omega$ and $t \in [0, 1]$. In fact, with a data sample $\boldsymbol{x}_1$ from $p_{\text{data}}(\cdot)$,

$$\boldsymbol{v}_t(\boldsymbol{x}|\boldsymbol{x}_1) = \boldsymbol{\phi}_t'(\boldsymbol{\phi}_t^{-1}(\boldsymbol{x}|\boldsymbol{x}_1)|\boldsymbol{x}_1), \tag{20a}$$

$$p_t(\boldsymbol{x}|\boldsymbol{x}_1) = p_0(\boldsymbol{\phi}_t^{-1}(\boldsymbol{x}|\boldsymbol{x}_1))\left|\det\left(\nabla_{\boldsymbol{x}}\boldsymbol{\phi}_t^{-1}(\boldsymbol{x}|\boldsymbol{x}_1)\right)\right|, \tag{20b}$$

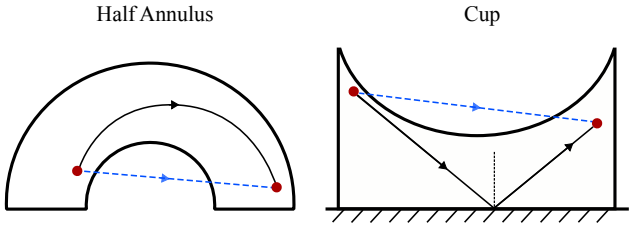

Figure 1. Illustration of the conditional velocity fields on two non-convex domains: half annulus (left) and cup (right). The black solid curve is the designed conditional velocity field within the domain. The OT conditional velocity field is represented by the blue dashed segment which violates the domain constraint.

naturally satisfy the continuity equation (14). Here we provide two concrete examples.

Example 3.6 (Convex Domain). Let Ω be a convex domain. Using the same idea of OT conditional velocity field, we set

$$\boldsymbol{\phi}_t(\boldsymbol{x}|\boldsymbol{x}_1) = (1-(1-\sigma_{\min})t)\boldsymbol{x} + (1-\sigma_{\min})t\boldsymbol{x}_1, \quad (21)$$

where $\sigma_{\min}$ is a sufficiently small number such that the resulting $p_1(\cdot|\boldsymbol{x}_1)$ is concentrated around $\boldsymbol{x}_1$. Note that equation (21) is different from the vanilla OT conditional velocity field in FM in that it has a coefficient $(1-\sigma_{\min})$ on $\boldsymbol{x}_1$ to ensure that $\boldsymbol{\phi}_t(\boldsymbol{x}|\boldsymbol{x}_1) \in \Omega$ due to the convexity of Ω. A typical example of convex domains is the convex polytope defined as

$$\Omega = \{\boldsymbol{u} \,|\, \boldsymbol{A}\boldsymbol{u} < \boldsymbol{b}\} \subset \mathbb{R}^d, \quad (22)$$

where $\boldsymbol{A} \in \mathbb{R}^{m\times d}$, $\boldsymbol{b} \in \mathbb{R}^m$, and m is the number of linear constraints. By taking $\boldsymbol{A} = [\boldsymbol{I}_d, -\boldsymbol{I}_d]'$ and $\boldsymbol{b} = \mathbf{1}_{2d}$ where $\boldsymbol{I}_d$ is the d-dimensional identity matrix and $\mathbf{1}_{2d}$ is an all-ones vector of length $2d$, the corresponding constrained domain becomes a hypercube $\Omega = [-1,1]^d$, which is frequently encountered in image generation tasks.

Example 3.7 (Half Annulus). Consider a two-dimensional half annulus area

$$\Omega = \{(x_1, x_2) \,|\, r^2 \le x_1^2 + x_2^2 \le R^2, x_2 \ge 0\} \subset \mathbb{R}^2 \quad (23)$$

with $0 < r < R$. Let $\boldsymbol{x} = (\|\boldsymbol{x}\|\cos\alpha, \|\boldsymbol{x}\|\sin\alpha)'$ and $\boldsymbol{x}_1 = (\|\boldsymbol{x}_1\|\cos\alpha_1, \|\boldsymbol{x}_1\|\sin\alpha_1)'$ where $\alpha, \alpha_1 \in [0,\pi]$. The conditional flow that (approximately) connects $\boldsymbol{x}$ and $\boldsymbol{x}_1$ can be given by $\boldsymbol{\phi}_t(\boldsymbol{x}|\boldsymbol{x}_1) = (x_t\cos\alpha_t, x_t\sin\alpha_t)'$ where

$$x_t = (1-(1-\sigma_{\min})t)\|\boldsymbol{x}\| + (1-\sigma_{\min})t\|\boldsymbol{x}_1\|, \quad (24a)$$
$$\alpha_t = (1-(1-\sigma_{\min})t)\alpha + (1-\sigma_{\min})t\alpha_1. \quad (24b)$$

It is easy to verify that $\boldsymbol{\phi}_t(\boldsymbol{x}|\boldsymbol{x}_1) \in \Omega$ for all $\boldsymbol{x}$, $\boldsymbol{x}_1$, and t. Intuitively, equation (24) forms an OT conditional velocity field that (approximately) pushes $\boldsymbol{x}$ to $\boldsymbol{x}_1$ in the polar coordinates. See the left plot in Figure 1 for an illustration.

Algorithm 1 Sampling from Reflected CNFs

Input: A velocity model $\boldsymbol{v}_\theta(\boldsymbol{x}, t)$; the prior distribution $p_0(\cdot)$; total number of discretization steps K; an increasing time sequences $\{\gamma_k\}_{k=0}^K$ such that $\gamma_0 = 0$ and $\gamma_K = 1$; an one-step ODE solver SOLVER$(\cdot)$; a reflection function $\mathrm{Refl}_{\partial\Omega}(\cdot)$.
Output: A sample $\boldsymbol{x}_1$ from the reflected CNF.
Sample a point $\boldsymbol{x}_0$ from $p_0(\cdot)$;
for $k = 1$ **to** K **do**
  $\bar{\boldsymbol{x}}_{\gamma_k} \leftarrow \mathrm{SOLVER}\left(\boldsymbol{x}_{\gamma_{k-1}}, \gamma_{k-1}, \gamma_k, \boldsymbol{v}_\theta\right)$;
  $(\Delta, \boldsymbol{\alpha}, \boldsymbol{y}) \leftarrow \left(\|\bar{\boldsymbol{x}}_{\gamma_k} - \boldsymbol{x}_{\gamma_{k-1}}\|, \frac{\bar{\boldsymbol{x}}_{\gamma_k} - \boldsymbol{x}_{\gamma_{k-1}}}{\|\bar{\boldsymbol{x}}_{\gamma_k} - \boldsymbol{x}_{\gamma_{k-1}}\|}, \boldsymbol{x}_{\gamma_{k-1}}\right)$;
  while TRUE **do**
    Calculate the first intersection $\boldsymbol{y}'$ of the ray $\boldsymbol{y} + \boldsymbol{\alpha}s(s \ge 0)$ and the boundary $\partial\Omega$;
    $\Delta' \leftarrow \|\boldsymbol{y}' - \boldsymbol{y}\|$;
    if $\Delta \le \Delta'$ **then**
      Stop the inner loop;
    end if
    $(\Delta, \boldsymbol{\alpha}, \boldsymbol{y}) \leftarrow (\Delta - \Delta', \mathrm{Refl}_{\partial\Omega}(\boldsymbol{\alpha}, \boldsymbol{y}'), \boldsymbol{y}')$;
  end while
  $\boldsymbol{x}_{\gamma_k} \leftarrow \boldsymbol{y} + \boldsymbol{\alpha}\Delta \in \Omega$;
end for

Compared to RDMs (Lou & Ermon, 2023), the superiority of RFM can be summarized in three key aspects: i) one can efficiently sample from $p_t(\boldsymbol{x}|\boldsymbol{x}_1)$ through the analytical conditional flow $\boldsymbol{\phi}_t(\boldsymbol{x}|\boldsymbol{x}_1)$ to obtain the training data in $\mathcal{L}_{\text{CRFM}}(\boldsymbol{\theta})$ without requiring additional geometric techniques; ii) the conditional velocity field $\boldsymbol{v}_t(\boldsymbol{x}|\boldsymbol{x}_1)$ has an analytical form, eliminating the need for potentially complicated approximations; iii) in principle, the prior distribution can take any distribution on arbitrary domain Ω, thereby enhancing the flexibility of reflected CNFs.

3.4. Sampling from Reflected CNFs

To generate a sample from the learned reflected CNFs, we start from the prior distribution and simulate the reflected ODE (13) where $\boldsymbol{v}_t(\boldsymbol{x})$ is replaced by $\boldsymbol{v}_\theta(\boldsymbol{x}, t)$. Assume an increasing sequence $0 = \gamma_0 < \cdots < \gamma_K = 1$ of the interpolation time points for ODE simulation. Here, we only discuss the simulation methods with fixed step sizes, and those with adaptive step sizes can be derived similarly.

First of all, we sample a point $\boldsymbol{x}_0$ from the prior distribution $p_0(\cdot)$ as our initialization as $t = 0$. In the k-th step, given the current position $\boldsymbol{x}_{\gamma_{k-1}}$, we need to calculate the next position $\boldsymbol{x}_{\gamma_k}$. For ODEs without spatial constraints, this one-step update can be achieved by many widely-used algorithms, e.g., Euler method, Runge-Kutta method, etc. These algorithms can generally be represented by

$$\bar{\boldsymbol{x}}_{\gamma_k} = \mathrm{SOLVER}\left(\boldsymbol{x}_{\gamma_{k-1}}, \gamma_{k-1}, \gamma_k, \boldsymbol{v}_\theta\right), \quad (25)$$

where the ODE solver $\mathrm{SOLVER}(\cdot)$ takes the current position $\boldsymbol{x}_{\gamma_{k-1}}$, the starting and ending time γ_{k-1}, γ_k, and a velocity field $\boldsymbol{v}_\theta$ as inputs. However, if the ODE solver does not account for the boundary constraints, the next position $\bar{\boldsymbol{x}}_{\gamma_k}$ may extend beyond the constrained domain Ω.

Inspired by the geometry implication of the reflection term $\mathrm{d}\boldsymbol{L}_t$, we handle the case $\bar{\boldsymbol{x}}_{\gamma_k} \notin \Omega$ by applying reflections iteratively to the segment from $\boldsymbol{x}_{\gamma_{k-1}}$ to $\bar{\boldsymbol{x}}_{\gamma_k}$. Concretely, let $\Delta = \|\bar{\boldsymbol{x}}_{\gamma_k} - \boldsymbol{x}_{\gamma_{k-1}}\|$ and $\boldsymbol{\alpha} = \frac{\bar{\boldsymbol{x}}_{\gamma_k} - \boldsymbol{x}_{\gamma_{k-1}}}{\|\bar{\boldsymbol{x}}_{\gamma_k} - \boldsymbol{x}_{\gamma_{k-1}}\|} \in \mathbb{R}^d$ be the length and direction of the one-step update, and $\boldsymbol{y} = \boldsymbol{x}_{\gamma_{k-1}}$ is the current position. We first calculate the first intersection $\boldsymbol{y}'$ of the ray $\boldsymbol{y} + \boldsymbol{\alpha} s (s \geq 0)$ and the boundary $\partial\Omega$. If $\|\boldsymbol{y}' - \boldsymbol{y}\| =: \Delta' < \Delta$, we continue the iteration and update the triplet $(\Delta, \boldsymbol{\alpha}, \boldsymbol{y})$ by

$$(\Delta, \boldsymbol{\alpha}, \boldsymbol{y}) \leftarrow (\Delta - \Delta', \mathrm{Refl}_{\partial\Omega}(\boldsymbol{\alpha}, \boldsymbol{y}'), \boldsymbol{y}'), \tag{26}$$

where $\mathrm{Refl}_{\partial\Omega}(\boldsymbol{\alpha}, \boldsymbol{y}')$ gives the reflected velocity of $\boldsymbol{\alpha}$ at $\boldsymbol{y}'$ on $\partial\Omega$; otherwise, we stop the iteration and calculate

$$\boldsymbol{x}_{\gamma_k} = \boldsymbol{y} + \boldsymbol{\alpha}\Delta \in \Omega \tag{27}$$

as the next position. The whole procedure for sampling from reflected CNFs is summarized in Algorithm 1.

Note that for certain special domains Ω, it is possible to directly compute the next position instead of applying reflections iteratively. For example, for $\Omega = [-1, 1]^d$, we can directly compute $\boldsymbol{x}_{\gamma_k} = 1 - |(\bar{\boldsymbol{x}}_{\gamma_k} + 1) \bmod 4 - 2|$. This observation is crucial to efficient sampling in image generation tasks.

Flow Guidance Inspired by the classifier-free guidance for score-based conditional generation (Section 2.3), we consider a similar scheme for guided generation based on the reflected ODE (13). With a c-conditioned velocity field $\boldsymbol{v}_t(\boldsymbol{x}|c)$ which pushes the prior distribution $p_0(\boldsymbol{x}|c) = p_0(\boldsymbol{x})$ to $p_1(\boldsymbol{x}|c)$, we defines the *guided velocity field* as

$$\tilde{\boldsymbol{v}}_t(\boldsymbol{x}|c) = w\boldsymbol{v}_t(\boldsymbol{x}|c) + (1 - w)\boldsymbol{v}_t(\boldsymbol{x}|\emptyset), \tag{28}$$

where w is the guidance weight and $\boldsymbol{v}_t(\boldsymbol{x}|\emptyset)$ is the unconditioned velocity with c set to the empty token. A similar idea is also considered in Dao et al. (2023); Zheng et al. (2023). According to the reflected ODE, the *guided probability path* $\tilde{p}_t(\boldsymbol{x}|c)$ is then defined implicitly by the solution to the following continuity equation with the Neumann boundary condition

$$\frac{\partial}{\partial t}\tilde{p}_t(\boldsymbol{x}|c) = -\nabla_{\boldsymbol{x}} \cdot (\tilde{\boldsymbol{v}}_t(\boldsymbol{x}|c)\tilde{p}_t(\boldsymbol{x}|c)), \boldsymbol{x} \in \Omega^o, \tag{29a}$$

$$\tilde{p}_t(\boldsymbol{x}|c)\tilde{\boldsymbol{v}}_t(\boldsymbol{x}|c) \cdot \boldsymbol{n}(\boldsymbol{x}) = 0, \boldsymbol{x} \in \partial\Omega, \tag{29b}$$

where $\boldsymbol{n}(\boldsymbol{x})$ is the outward normal direction at $\boldsymbol{x} \in \partial\Omega$. In particular, one has $\tilde{p}_t(\boldsymbol{x}|c) = p_t(\boldsymbol{x})$ or $\tilde{p}_t(\boldsymbol{x}|c) = p_t(\boldsymbol{x}|c)$ if $w = 0$ or $w = 1$ respectively.

The following theorem implies that in the vanilla FM setting, the velocity field is connected to that in the ODE counterpart of the reversed SDE (i.e., $\lambda = 0$ in equation (10)), establishing an equivalence between equation (12) and (28).

Theorem 3.8. *Let $\Omega = \mathbb{R}^d$ and the prior distribution $p_0(\boldsymbol{x})$ be a standard Gaussian distribution. Assume the OT conditional velocity field for FM, i.e., $\boldsymbol{\phi}_t(\boldsymbol{x}|\boldsymbol{x}_1) = t\boldsymbol{x}_1 + (1 - (1 - \sigma_{\min})t)\boldsymbol{x}$. Then*

$$\boldsymbol{v}_t(\boldsymbol{x}) = \frac{1}{t}\boldsymbol{x} + \frac{1 - (1 - \sigma_{\min})t}{t}\nabla \log p_t(\boldsymbol{x}). \tag{30}$$

The proof of Theorem 3.8 can be found in Appendix A.4.

4. Experiments

4.1. Low-dimensional Toy Examples

We first test the effectiveness of RFM for modeling probability distributions on low-dimensional constrained domains, including hypercube, simplex, half annulus, and cup.

Hypercube and Simplex We consider the d-dimensional hypercube $\Omega = [-1, 1]^d$ and the d-dimensional simplex

$$\Omega = \left\{ (x_1, \ldots, x_d)' \middle| x_1 \geq 0, \ldots, x_d \geq 0, \sum_{i=1}^d x_i \leq 1 \right\}.$$

Both domains are convex and satisfy the condition in Example 3.6. We consider two cases here: $d = 2$ and $d = 10$. For RFM, we use the OT conditional velocity field in equation (21) with $\sigma_{\min} = 10^{-5}$.

Half Annulus We consider a two-dimensional half annulus area which is defined in Example 3.7 and choose the conditional velocity field in equation (24) with $\sigma_{\min} = 10^{-5}$ for RFM.

Cup We also consider a non-convex two-dimensional cup area defined as

$$\Omega = \{(x_1, x_2)' | -1 \leq x_1 \leq 1, x_2 \geq 0, x_1^2 + (x_2 - 3)^2 \leq 2\}.$$

Let $\boldsymbol{x} = (x_1, x_2)$ and $\boldsymbol{x}_1 = (x_{1,1}, x_{1,2})$ be the prior sample and data sample. We define the conditional flow as

$$\boldsymbol{\phi}_t(\boldsymbol{x}|\boldsymbol{x}_1) = ((1 - t)x_1 + tx_{1,1}, |(1 - t)x_2 - tx_{1,2}|)',$$

which is a broken line connecting $\boldsymbol{x}$ and $\boldsymbol{x}_1$ with one reflection at the x-axis (Figure 1, right plot).

For all these domains above, we set the prior distribution $p_0(\cdot)$ to the standard Gaussian distribution for CNFs (trained with FM) and the uniform distribution over the constrained domain Ω for reflected CNFs (trained with RFM). As an additional baseline, the CNFs (trained with FM) where $p_0(\cdot)$

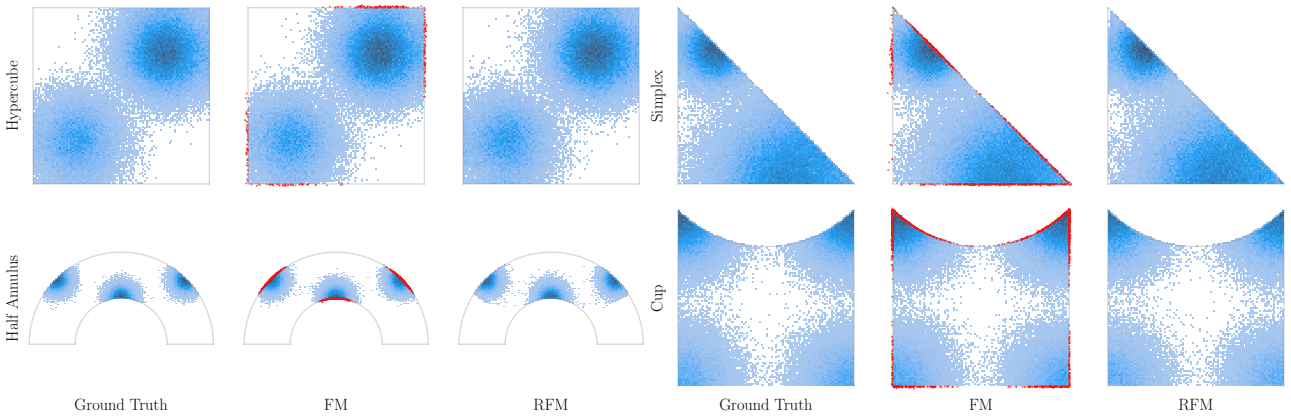

Figure 2. The histplots of samples obtained by different methods compared to the ground truth on the two-dimensional hypercube, simplex, and cup data set. Samples out of the constrained domain are plotted with red dots. The total sample size is 100,000.

Table 1. KL divergences to the ground truth obtained by different methods on low-dimensional generation tasks. For each method, we calculate the KL divergence using the Python ITE module (Szabó, 2014) with a sample size of 50,000. The results are averaged over 10 independent runs with standard deviation in the brackets.

Method	Hypercube		Simplex		Half Annulus	Cup
	$d = 2$	$d = 10$	$d = 2$	$d = 10$	$d = 2$	$d = 2$
RDM	0.0110(0.0003)	0.4798(0.0006)	0.0055(0.0001)	0.7613(0.0036)	N/A	N/A
FM	0.0044(0.0010)	**0.0224(0.0010)**	0.0062(0.0011)	0.0522(0.0022)	0.0083(0.0008)	0.0122(0.0012)
FM*	**0.0021(0.0010)**	0.0257(0.0020)	0.0045(0.0009)	0.0460(0.0022)	0.0085(0.0011)	0.0118(0.0020)
RFM	**0.0021(0.0011)**	0.0243(0.0011)	**0.0027(0.0011)**	**0.0452(0.0023)**	**0.0038(0.0009)**	**0.0058(0.0010)**

Table 2. Constraint violation ratio (‰) of different methods on low-dimensional generation tasks. We collect 500,000 samples to calculate the constraint violation ratio.

Method	Hypercube		Simplex		Half Annulus	Cup
	$d = 2$	$d = 10$	$d = 2$	$d = 10$	$d = 2$	$d = 2$
FM	3.1	26.9	9.0	86.2	10.4	13.9
FM*	0.2	8.6	5.9	33.7	10.2	10.2
RFM	0.0	0.0	0.0	0.0	0.0	0.0

is a uniform distribution are also considered and denoted by FM*. We choose the target distribution $p_{\text{data}}(\cdot)$ as the mixture of Gaussians which is truncated by the domain Ω (Fishman et al., 2023). On all domains, the OT conditional velocity field with $\sigma_{\min} = 10^{-5}$ is chosen for FM. The results are collected after 200,000 iterations with a batch size of 512. We use the popular fixed-step third-order Heun algorithm (Chen et al., 2018) and set the number of function evaluations (NFE) as 300 to sample from the CNFs and the reflected CNFs. We also consider the RDM baseline implemented by Fishman et al. (2023) and generate the samples using the Euler-Maruyama algorithm (Kloeden & Platen, 1992) with 300 NFE. See more details of implementation in Appendix B.1.

Table 1 reports the approximation accuracies measured by the Kullback–Leibler (KL) divergences to the ground truth obtained by different methods. We find that in all cases, RFM performs on par or better than FM and FM* and consistently outperforms RDM. Interestingly, the results of RDM get worse in the 10-dimensional cases, partially due to the requirement for more discretization steps. One can also see from Table 2 that a large proportion of samples violating the constraints are generated by FM and FM*, while our RFM enjoys zero constraint violation ratio by design. This demonstrates the effectiveness of the reflection operation in CNFs. Interestingly, the boundary violation ratio of CNFs can be somewhat reduced by employing a uniform prior distribution, as evidenced by the FM* results.

Figure 2 shows the histplots of samples obtained by different methods. We see that both CNFs and reflected CNFs (trained by FM and RFM respectively) can generate samples similar to the data distributions. Moreover, samples from reflected CNFs all stay within the domains while those from CNFs can violate the constraints (the red dots). It is worth noting that the conditional velocity fields of RFM on the nonconvex half annulus and cup areas work well, although

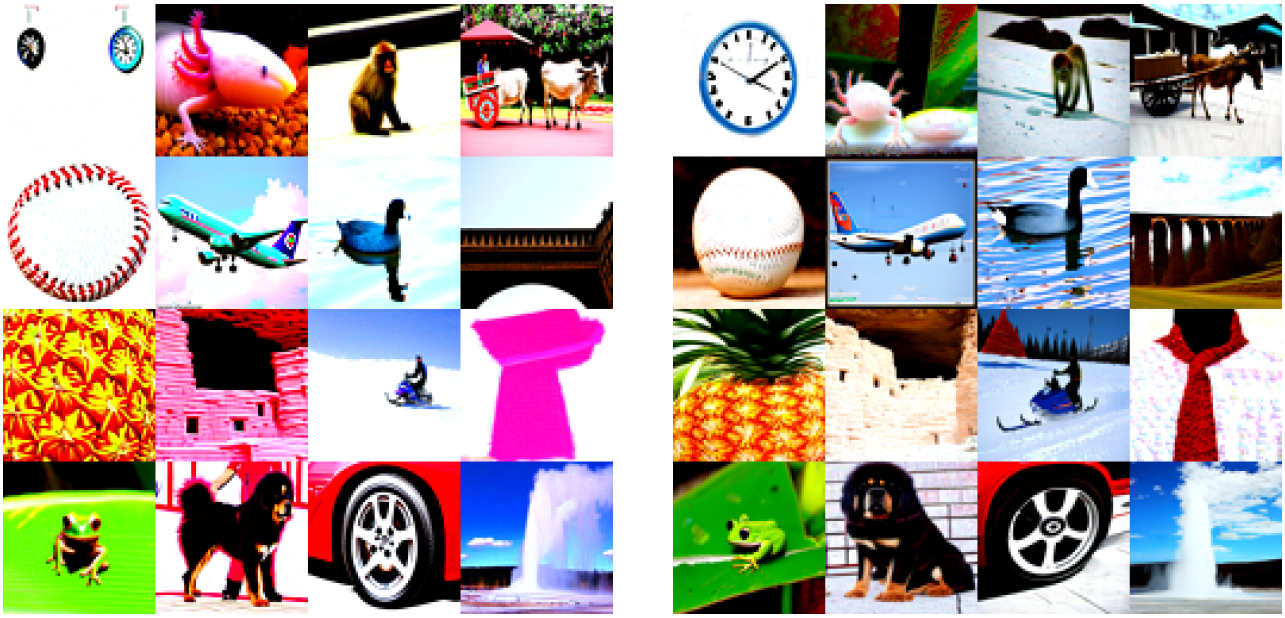

Figure 3. Class-conditioned guided samples from the CNFs trained with FM (left, NFE $= 769$) and the reflected CNFs trained with RFM (right, NFE $= 723$) with a high guidance weight $w = 15$ on ImageNet (64×64). The samples from CNFs are clipped to $[0, 255]$ after the ODE simulation.

Table 3. Sample quality measured by FID score (↓) obtained by different methods on CIFAR-10 (32×32). Methods for constrained generation are marked in the shaded area. The result with $*$ is produced by ourselves using the official checkpoint in Lou & Ermon (2023) and the predictor-corrector (PC) algorithm. The other results are from their original papers.

Method	FID	NFE
NCSN++ (Song et al., 2020b)	2.20	2000
DDPM++ (Song et al., 2020b)	2.41	2000
Subspace NCSN++ (Jing et al., 2022a)	2.17	2000
DDIM (Song et al., 2020a)	4.16	100
FM (Lipman et al., 2023)	6.35	142
RDM (Lou & Ermon, 2023)	2.72	2000
RDM (Lou & Ermon, 2023)	44.40*	200
RFM (ours)	4.76	139

they are not optimal transports. More results about alternative ODE solvers and learned velocity fields can be found in Appendix C.1.

4.2. Image Generation Benchmarks

We then explore the performances of RFM for the unconditional image generation task on CIFAR-10 (32×32) and the conditional generation task on ImageNet (64×64). For both data sets, the data are rescaled to a hypercube $\Omega = [-1, 1]^{3\times m\times m}$ (m is the image size), which is a convex domain (Example 3.6). For CNFs trained with FM, we set the prior distribution to be a standard Gaussian distribution and use the OT conditional velocity field. For reflected CNFs trained with RFM, we set the prior distribution to be a truncated standard Gaussian distribution over Ω and use the OT conditional velocity field in equation (21). The architectures of the velocity model $\boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}, t)$ for CIFAR-10 or the class-conditioned velocity model $\boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}, t, c)$ for ImageNet employed by RFM and FM are the same as the UNet (Ronneberger et al., 2015) of the score function model in Dhariwal & Nichol (2021). On CIFAR-10, the velocity model of RFM is optimized with Adam (Kingma & Ba, 2015) and a constant learning rate of 0.0002 after a warm-up phase of 5000 training steps; on ImageNet, the velocity model of RFM is optimized with AdamW (Loshchilov & Hutter, 2018) and a constant learning rate of 0.0001 after a warm-up phase of 5000 training steps. The total number of training steps is 800,000 on CIFAR-10 and 540,000 on ImageNet. The batch size is set to 128 on CIFAR-10 and 2048 on ImageNet. We also train CNFs with FM on ImageNet by ourselves using the same setting as RFM. We use the adaptive-step Dormand–Prince method (Dormand & Prince, 1980) with absolute and relative tolerances of 10^{-5} to sample from the CNFs and the reflected CNFs. See more implementation details in Appendix B.2 and B.3.

Unconditional Image Generation In Table 3, we draw 50,000 samples and report the Fréchet inception distance (FID) score (Seitzer, 2020) as well as the NFE obtained by different methods. Among the methods for constrained

Table 4. Digit-level constraint violation ratio of FM and RFM on the ImageNet (64×64) conditional generation task. (Reflected) CNFs are simulated by the fixed-step third-order Heun algorithm with 600 NFE.

Flow guidance weight w	7.5	15	25
FM	51.28%	61.12%	64.51%
RFM	0.0%	0.0%	0.0%

generation, we see that although RDM with the predictor-corrector (PC) sampler works well under a large NFE, it has a large performance drop when the NFE is small. In contrast, RFM can obtain high-quality samples on constrained domains with small NFEs, inheriting the benefit of straighter velocity fields of FM. Besides, we want to point out that the FID score produced by FM can be sensitive to the experimental setting (e.g., FID=3.66 reproduced by Tong et al. (2023)). In comparison with FM, the benefit of RFM mainly comes from a zero boundary violation rate and more realistic image samples (see the next paragraph), and the FID score is not very sensitive to the boundary violation issue. Image samples generated from reflected CNFs trained with RFM can be found in Figure 7 of Appendix C.2.

Conditional Image Generation Figure 3 shows the class-conditioned samples generated by the CNFs trained with FM and the reflected CNFs trained with RFM under a high guidance weight using the flow guidance introduced in Section 3.4. We see that the vanilla CNFs tend to generate oversaturated and unnatural images due to the boundary violation. In contrast, reflected CNFs trained with RFM effectively improve the sample quality by producing more faithful images with a comparable NFE. Table 4 reports the constraint violation ratios of different methods with varying guidance weights. We see that the constraint violation ratio of FM is large for this high-dimensional task and can increase as the flow guidance weight gets larger.

5. Conclusion

In this work, we presented reflected flow matching (RFM), an extension of flow matching for constrained domains through reflected ODEs/CNFs. By leveraging conditional probability paths that adhere to the constraints, we show that the velocity fields in the reflected ODEs/CNFs can be effectively trained by matching the analytical conditional velocity fields in a simulation-free manner, akin to FM. We demonstrate the efficacy of RFM across various choices of conditional velocity fields on low-dimensional convex and nonconvex domains. On standard image benchmarks, we show that RFM performs on par or better than existing baselines with a small NFE and produces more faithful samples under high guidance weight. Further applications of RFM to constrained generative modeling of high-resolution images, videos, cloud points, human motions, etc., will be interesting future directions.

Limitations To apply RFM, we have the following requirements for the constrained domain: i) the domain has a regular shape so that we can design a conditional flow that is confined to it. All convex domains satisfy this requirement. ii) the boundary has an analytical form to compute the intersection point and the reflection direction if necessary. iii) the conditional flow is easy to simulate and differentiate for velocity computation. We leave exploring the design of conditional probability paths on more general domains to future work.

Impact Statement

This paper presents work whose goal is to advance the field of machine learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

Acknowledgements

This work was supported by National Natural Science Foundation of China (grant no. 12201014 and grant no. 12292983). The research of Cheng Zhang was support in part by National Engineering Laboratory for Big Data Analysis and Applications, the Key Laboratory of Mathematics and Its Applications (LMAM) and the Key Laboratory of Mathematical Economics and Quantitative Finance (LMEQF) of Peking University. The authors appreciate the anonymous ICML reviewers for their constructive feedback.

References

Albergo, M. S. and Vanden-Eijnden, E. Building normalizing flows with stochastic interpolants. In *The Eleventh International Conference on Learning Representations*, 2022.

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein gan. *ArXiv*, abs/1701.07875, 2017. URL https://api.semanticscholar.org/CorpusID:13943041.

Bortoli, V. D., Mathieu, E., Hutchinson, M. J., Thornton, J., Teh, Y. W., and Doucet, A. Riemannian score-based generative modelling. In *Neural Information Processing Systems*, 2022.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Cattiaux, P. Time reversal of diffusion processes with a boundary condition. *Stochastic processes and their Applications*, 28(2):275–292, 1988.

Chen, R. T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. Neural ordinary differential equations. *Advances in Neural Information Processing Systems*, 2018.

Dao, Q., Phung, H., Nguyen, B., and Tran, A. Flow matching in latent space. *arXiv preprint arXiv:2307.08698*, 2023.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Dormand, J. R. and Prince, P. J. A family of embedded runge-kutta formulae. *Journal of computational and applied mathematics*, 6(1):19–26, 1980.

Fishman, N., Klarner, L., Bortoli, V. D., Mathieu, E., and Hutchinson, M. J. Diffusion models for constrained domains. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.

Hendrycks, D. and Gimpel, K. Gaussian error linear units (GELUs). *arXiv preprint arXiv:1606.08415*, 2016.

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

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Hoogeboom, E., Satorras, V. G., Vignac, C., and Welling, M. Equivariant diffusion for molecule generation in 3d. In *International conference on machine learning*, pp. 8867–8887. PMLR, 2022.

Jin, W., Barzilay, R., and Jaakkola, T. Junction tree variational autoencoder for molecular graph generation. In *International conference on machine learning*, pp. 2323–2332. PMLR, 2018.

Jing, B., Corso, G., Berlinghieri, R., and Jaakkola, T. Subspace diffusion generative models. In *European Conference on Computer Vision*, pp. 274–289. Springer, 2022a.

Jing, B., Corso, G., Chang, J., Barzilay, R., and Jaakkola, T. S. Torsional diffusion for molecular conformer generation. In *Advances in Neural Information Processing Systems*, 2022b.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Klein, L., Krämer, A., and Noe, F. Equivariant flow matching. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Kloeden, P. and Platen, E. *Numerical solution of stochastic differential equations*. Springer, 1992.

Lamperski, A. Projected stochastic gradient langevin algorithms for constrained sampling and non-convex learning. In *Conference on Learning Theory*, pp. 2891–2937. PMLR, 2021.

Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Liu, G.-H., Chen, T., Theodorou, E., and Tao, M. Mirror diffusion models for constrained and watermarked generation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=XPWEtXzlLy.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

Lou, A. and Ermon, S. Reflected diffusion models. *arXiv preprint arXiv:2304.04740*, 2023.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In *Neural Information Processing Systems*, 2019.

Pilipenko, A. *An introduction to stochastic differential equations with reflection*, volume 1. Universitätsverlag Potsdam, 2014.

Rezende, D. and Mohamed, S. Variational inference with normalizing flows. In *Proceedings of The 32nd International Conference on Machine Learning*, pp. 1530–1538, 2015.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computerassisted intervention*. Springer, 2015.

Schuss, Z. *Brownian dynamics at boundaries and interfaces*. Springer, 2015.

Seitzer, M. pytorch-fid: FID Score for PyTorch. https://github.com/mseitzer/pytorch-fid, August 2020. Version 0.3.0.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020a.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020b.

Song, Y., Dhariwal, P., Chen, M., and Sutskever, I. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

Szabó, Z. Information theoretical estimators toolbox. *The Journal of Machine Learning Research*, 15(1):283–287, 2014.

Tong, A., Malkin, N., Huguet, G., Zhang, Y., Rector-Brooks, J., Fatras, K., Wolf, G., and Bengio, Y. Improving and generalizing flow-based generative models with minibatch optimal transport. In *ICML Workshop on New Frontiers in Learning, Control, and Dynamical Systems*, 2023.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Vincent, P. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011. doi: 10.1162/NECO_a_00142.

Wildberger, J. B., Dax, M., Buchholz, S., Green, S. R., Macke, J. H., and Schölkopf, B. Flow matching for scalable simulation-based inference. In *ICML 2023 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2023.

Williams, R. On time-reversal of reflected brownian motions. In *Seminar on Stochastic Processes, 1987*, pp. 265–276. Springer, 1987.

Xu, M., Yu, L., Song, Y., Shi, C., Ermon, S., and Tang, J. Geodiff: A geometric diffusion model for molecular conformation generation. In *International Conference on Learning Representations*, 2021.

Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

Zheng, Q., Le, M., Shaul, N., Lipman, Y., Grover, A., and Chen, R. T. Q. Guided flows for generative modeling and decision making. *ArXiv*, abs/2311.13443, 2023.

A. Theoretical Results

A.1. Details of Theorem 3.1

Theorem 3.1. Assume
(i) the domain Ω satisfies the *uniform exterior sphere condition*, i.e.,

$$\exists r_0 > 0, \forall \boldsymbol{x} \in \partial\Omega, \text{we have } \left\{\boldsymbol{n} \middle| \|\boldsymbol{n}\| = 1, B(\boldsymbol{x} - r_0\boldsymbol{n}, r_0) \cap \Omega = \emptyset\right\} \neq \emptyset,$$

where $B(\boldsymbol{x}, r)$ is a open ball with center $\boldsymbol{x}$ and radius r;
(ii) the domain Ω satisfies the *uniform cone condition*, i.e.,

$$\exists \delta > 0, \alpha \in [0, 1), \forall \boldsymbol{x} \in \partial\Omega, \text{we have } \left\{\boldsymbol{m} \middle| \|\boldsymbol{m}\| = 1, \forall \boldsymbol{y} \in B(\boldsymbol{x}, \delta) \cap \partial\Omega, C(\boldsymbol{y}, \boldsymbol{m}, \alpha) \cup B(\boldsymbol{x}, \delta) \subset \Omega\right\} \neq \emptyset,$$

where $C(\boldsymbol{y}, \boldsymbol{m}, \alpha) = \left\{\boldsymbol{z} \middle| (\boldsymbol{z} - \boldsymbol{y}) \cdot \boldsymbol{m} \geq \|\boldsymbol{z} - \boldsymbol{y}\|\alpha\right\}$.
(iii) the velocity field $\boldsymbol{v}_t(\boldsymbol{x})$ is Lipschitz continuous in $\boldsymbol{x} \in \Omega$ (uniformly on t).
Then the solution to the reflected ODE (13) exists and is unique on $t \in [0, 1]$.

Sketch of Proof The reflected ODE we considered can be viewed as a special case of reflected SDE where the diffusion coefficient is zero. Therefore, Theorem 3.1 is indeed a corollary of Pilipenko (2014, Theorem 2.5.4) with no diffusion term (i.e., diffusion coefficients $b_k(\boldsymbol{x}_t)$, $1 \leq k \leq m$, are zeros).

For a more rigorous derivation, note that Pilipenko (2014, Theorem 2.5.2) says that under the assumption (i) (ii) of the domain D in our Theorem 3.1, there exists a unique solution to the Skorokhod problem with normal reflection as long as the drift function $a(\boldsymbol{x}_t)$ is continuous. To prove the existence and uniqueness of the solution to the reflected ODE, one can first use Euler discretization to find a sequence of approximate solutions whose existence and uniqueness are guaranteed by Pilipenko (2014, Theorem 2.5.2). We can then prove that this sequence of approximate solutions converges to the solution of the reflected ODE problem as the stepsize goes to zero. This proof simply follows the proof provided by Pilipenko (2014) for reflected SDEs.

A.2. Proof of Theorem 3.4

First of all, we have

$$\|\boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}, t) - \boldsymbol{v}_t(\boldsymbol{x})\|^2 = \|\boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}, t)\|^2 - 2\boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}, t) \cdot \boldsymbol{v}_t(\boldsymbol{x}) + \|\boldsymbol{v}_t(\boldsymbol{x})\|^2 \tag{31a}$$

$$\|\boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}, t) - \boldsymbol{v}_t(\boldsymbol{x}|\boldsymbol{x}_1)\|^2 = \|\boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}, t)\|^2 - 2\boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}, t) \cdot \boldsymbol{v}_t(\boldsymbol{x}|\boldsymbol{x}_1) + \|\boldsymbol{v}_t(\boldsymbol{x}|\boldsymbol{x}_1)\|^2 \tag{31b}$$

To prove that $\nabla\mathcal{L}_{\text{RFM}}(\boldsymbol{\theta}) = \nabla\mathcal{L}_{\text{CRFM}}(\boldsymbol{\theta})$, it can be easily seen that

$$\mathbb{E}_{p_{\text{data}}(\boldsymbol{x}_1)p_t(\boldsymbol{x}|\boldsymbol{x}_1)}\|\boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}, t)\|^2 = \mathbb{E}_{p_t(\boldsymbol{x})}\|\boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}, t)\|^2 \tag{32}$$

by the definition of $p_t(\boldsymbol{x})$ in equation (16). The remaining thing to show is

$$\mathbb{E}_{p_{\text{data}}(\boldsymbol{x}_1)p_t(\boldsymbol{x}|\boldsymbol{x}_1)}\boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}, t) \cdot \boldsymbol{v}_t(\boldsymbol{x}|\boldsymbol{x}_1) = \mathbb{E}_{p_t(\boldsymbol{x})}\boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}, t) \cdot \boldsymbol{v}_t(\boldsymbol{x}). \tag{33}$$

By the definition of $\boldsymbol{v}_t(\boldsymbol{x})$ in equation (17), we have

$$\begin{aligned}
\mathbb{E}_{p_t(\boldsymbol{x})}\boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}, t) \cdot \boldsymbol{v}_t(\boldsymbol{x}) &= \int_\Omega p_t(\boldsymbol{x})\boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}, t) \cdot \boldsymbol{v}_t(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \\
&= \int_\Omega p_t(\boldsymbol{x})\boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}, t) \cdot \int_\Omega \boldsymbol{v}_t(\boldsymbol{x}|\boldsymbol{x}_1)\frac{p_t(\boldsymbol{x}|\boldsymbol{x}_1)p_{\text{data}}(\boldsymbol{x}_1)}{p_t(\boldsymbol{x})}\mathrm{d}\boldsymbol{x}_1\mathrm{d}\boldsymbol{x} \\
&= \int_\Omega\int_\Omega \boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}, t) \cdot \boldsymbol{v}_t(\boldsymbol{x}|\boldsymbol{x}_1)p_t(\boldsymbol{x}|\boldsymbol{x}_1)p_{\text{data}}(\boldsymbol{x}_1)\mathrm{d}\boldsymbol{x}_1\mathrm{d}\boldsymbol{x} \\
&= \mathbb{E}_{p_{\text{data}}(\boldsymbol{x}_1)p_t(\boldsymbol{x}|\boldsymbol{x}_1)}\boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}, t) \cdot \boldsymbol{v}_t(\boldsymbol{x}|\boldsymbol{x}_1).
\end{aligned}$$

This finishes the proof.

A.3. Proof of Theorem 3.5

We first give an alternative expression (Pilipenko, 2014) of the reflected ODE

$$\mathrm{d}\boldsymbol{\phi}_t(\boldsymbol{x}) = \boldsymbol{v}_t(\boldsymbol{\phi}_t(\boldsymbol{x}))\mathrm{d}t + \boldsymbol{n}(\boldsymbol{\phi}_t(\boldsymbol{x}))\mathrm{d}l_t, \tag{34a}$$

$$\boldsymbol{\phi}_0(\boldsymbol{x}) = \boldsymbol{x}, \tag{34b}$$

where the initial point $\boldsymbol{x} \sim p_0(\boldsymbol{x})$, $\boldsymbol{n}(\boldsymbol{x})$ is the inward unit normal vector at $\boldsymbol{x}$ on $\partial\Omega$, and l_t is non-decreasing in t with $l_0 = 0$. Moreover, l_t satisfies $\int_0^t \mathbb{I}(\boldsymbol{\phi}_s(\boldsymbol{x}) \notin \partial\Omega)\mathrm{d}l_s = 0$ for $t > 0$, i.e., $\mathrm{d}l_t$ vanishes in the interior and will push the velocity along $\boldsymbol{n}(\boldsymbol{x})$ when the trajectory hits the boundary.

Let $\boldsymbol{\phi}_{\boldsymbol{\theta},t}(\boldsymbol{x})$ (with a reflection term $\bar{l}_t$) be the solution to the reflected ODE (34) with velocity field $\boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}, t)$. By the definition of Wasserstein-2 distance, we have

$$W_2^2\left(p_1(\boldsymbol{x}), p_{\boldsymbol{\theta},1}(\boldsymbol{x})\right) \leq \int_\Omega \|\boldsymbol{\phi}_1(\boldsymbol{x}) - \boldsymbol{\phi}_{\boldsymbol{\theta},1}(\boldsymbol{x})\|^2 p_0(\boldsymbol{x})\mathrm{d}\boldsymbol{x} := \hat{W}_2^2\left(p_t(\boldsymbol{x}), p_{\boldsymbol{\theta},t}(\boldsymbol{x})\right)\big|_{t=1}. \tag{35}$$

Take the derivative of $\hat{W}_2^2\left(p_t(\boldsymbol{x}), p_{\boldsymbol{\theta},t}(\boldsymbol{x})\right)$ w.r.t. t gives

$$\mathrm{d}\hat{W}_2^2\left(p_t(\boldsymbol{x}), p_{\boldsymbol{\theta},t}(\boldsymbol{x})\right) = 2\int_\Omega \left(\boldsymbol{\phi}_t(\boldsymbol{x}) - \boldsymbol{\phi}_{\boldsymbol{\theta},t}(\boldsymbol{x})\right)\left(\mathrm{d}\boldsymbol{\phi}_t(\boldsymbol{x}) - \mathrm{d}\boldsymbol{\phi}_{\boldsymbol{\theta},t}(\boldsymbol{x})\right) p_0(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \tag{36}$$

Note that

$$\begin{aligned}
&\left(\boldsymbol{\phi}_t(\boldsymbol{x}) - \boldsymbol{\phi}_{\boldsymbol{\theta},t}(\boldsymbol{x})\right)\left(\mathrm{d}\boldsymbol{\phi}_t(\boldsymbol{x}) - \mathrm{d}\boldsymbol{\phi}_{\boldsymbol{\theta},t}(\boldsymbol{x})\right)\\
=&\left(\boldsymbol{\phi}_t(\boldsymbol{x}) - \boldsymbol{\phi}_{\boldsymbol{\theta},t}(\boldsymbol{x})\right)\left[\left(\boldsymbol{v}_t(\boldsymbol{\phi}_t(\boldsymbol{x})) - \boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{\phi}_{\boldsymbol{\theta},t}(\boldsymbol{x}), t)\right)\mathrm{d}t + \boldsymbol{n}(\boldsymbol{\phi}_t(\boldsymbol{x}))\mathrm{d}l_t - \boldsymbol{n}(\boldsymbol{\phi}_{\boldsymbol{\theta},t}(\boldsymbol{x}))\mathrm{d}\bar{l}_t\right]\\
=&\left(\boldsymbol{\phi}_t(\boldsymbol{x}) - \boldsymbol{\phi}_{\boldsymbol{\theta},t}(\boldsymbol{x})\right)\left(\boldsymbol{v}_t(\boldsymbol{\phi}_t(\boldsymbol{x})) - \boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{\phi}_{\boldsymbol{\theta},t}(\boldsymbol{x}), t)\right)\mathrm{d}t\\
&\qquad + \left(\boldsymbol{\phi}_t(\boldsymbol{x}) - \boldsymbol{\phi}_{\boldsymbol{\theta},t}(\boldsymbol{x})\right)\boldsymbol{n}(\boldsymbol{\phi}_t(\boldsymbol{x}))\mathrm{d}l_t - \left(\boldsymbol{\phi}_t(\boldsymbol{x}) - \boldsymbol{\phi}_{\boldsymbol{\theta},t}(\boldsymbol{x})\right)\boldsymbol{n}(\boldsymbol{\phi}_{\boldsymbol{\theta},t}(\boldsymbol{x}))\mathrm{d}\bar{l}_t,
\end{aligned}$$

The first term can be bounded by

$$\begin{aligned}
&\left(\boldsymbol{\phi}_t(\boldsymbol{x}) - \boldsymbol{\phi}_{\boldsymbol{\theta},t}(\boldsymbol{x})\right)\left(\boldsymbol{v}_t(\boldsymbol{\phi}_t(\boldsymbol{x})) - \boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{\phi}_{\boldsymbol{\theta},t}(\boldsymbol{x}), t)\right)\\
=&\left(\boldsymbol{\phi}_t(\boldsymbol{x}) - \boldsymbol{\phi}_{\boldsymbol{\theta},t}(\boldsymbol{x})\right)\left(\boldsymbol{v}_t(\boldsymbol{\phi}_t(\boldsymbol{x})) - \boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{\phi}_t(\boldsymbol{x}), t)\right) + \left(\boldsymbol{\phi}_t(\boldsymbol{x}) - \boldsymbol{\phi}_{\boldsymbol{\theta},t}(\boldsymbol{x})\right)\left(\boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{\phi}_t(\boldsymbol{x}), t) - \boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{\phi}_{\boldsymbol{\theta},t}(\boldsymbol{x}), t)\right)\\
\leq&\frac{1}{2}\|\boldsymbol{\phi}_t(\boldsymbol{x}) - \boldsymbol{\phi}_{\boldsymbol{\theta},t}(\boldsymbol{x})\|^2 + \frac{1}{2}\|\boldsymbol{v}_t(\boldsymbol{\phi}_t(\boldsymbol{x})) - \boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{\phi}_t(\boldsymbol{x}), t)\|^2 + M\|\boldsymbol{\phi}_t(\boldsymbol{x}) - \boldsymbol{\phi}_{\boldsymbol{\theta},t}(\boldsymbol{x})\|^2
\end{aligned}$$

where in the last inequality we use the mean inequality and the fact that $\boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}, t)$ is M-Lipschitz in $\boldsymbol{x}$. To handle the second term, note that $dl_t > 0$ only if $\boldsymbol{\phi}_t(\boldsymbol{x}) \in \partial\Omega$. When $\boldsymbol{\phi}_t(\boldsymbol{x}) \in \partial\Omega$, we know $\left(\boldsymbol{\phi}_t(\boldsymbol{x}) - \boldsymbol{\phi}_{\boldsymbol{\theta},t}(\boldsymbol{x})\right)\boldsymbol{n}(\boldsymbol{\phi}_t(\boldsymbol{x})) \leq 0$ due to the convexity of Ω. Therefore, the second term satisfies

$$\left(\boldsymbol{\phi}_t(\boldsymbol{x}) - \boldsymbol{\phi}_{\boldsymbol{\theta},t}(\boldsymbol{x})\right)\boldsymbol{n}(\boldsymbol{\phi}_t(\boldsymbol{x}))\mathrm{d}l_t \leq 0.$$

Using the same argument, we know that the third term satisfies

$$-\left(\boldsymbol{\phi}_t(\boldsymbol{x}) - \boldsymbol{\phi}_{\boldsymbol{\theta},t}(\boldsymbol{x})\right)\boldsymbol{n}(\boldsymbol{\phi}_{\boldsymbol{\theta},t}(\boldsymbol{x}))\mathrm{d}\bar{l}_t \leq 0$$

See Lamperski (2021) for a more rigorous argument. Therefore, we have the following bound for $\mathrm{d}\hat{W}_2^2\left(p_t(\boldsymbol{x}), p_{\boldsymbol{\theta},t}(\boldsymbol{x})\right)/\mathrm{d}t$

$$\begin{aligned}
&\mathrm{d}\hat{W}_2^2\left(p_t(\boldsymbol{x}), p_{\boldsymbol{\theta},t}(\boldsymbol{x})\right)/\mathrm{d}t\\
\leq&\int_\Omega \|\boldsymbol{\phi}_t(\boldsymbol{x}) - \boldsymbol{\phi}_{\boldsymbol{\theta},t}(\boldsymbol{x})\|^2 p_0(\boldsymbol{x})\mathrm{d}\boldsymbol{x} + \int_\Omega \|\boldsymbol{v}_t(\boldsymbol{\phi}_t(\boldsymbol{x})) - \boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{\phi}_t(\boldsymbol{x}), t)\|^2 p_0(\boldsymbol{x})\mathrm{d}\boldsymbol{x} + 2M\int_\Omega \|\boldsymbol{\phi}_t(\boldsymbol{x}) - \boldsymbol{\phi}_{\boldsymbol{\theta},t}(\boldsymbol{x})\|^2 p_0(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\\
=&(1+2M)\hat{W}_2^2\left(p_t(\boldsymbol{x}), p_{\boldsymbol{\theta},t}(\boldsymbol{x})\right) + \int_\Omega \|\boldsymbol{v}_t(\boldsymbol{\phi}_t(\boldsymbol{x})) - \boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{\phi}_t(\boldsymbol{x}), t)\|^2 p_0(\boldsymbol{x})\mathrm{d}\boldsymbol{x}.
\end{aligned}$$

By Gronwall's inequality and $\hat{W}_2^2\left(p_t(\boldsymbol{x}), p_{\boldsymbol{\theta},t}(\boldsymbol{x})\right)\big|_{t=0} = 0$, we have

$$\hat{W}_2^2\left(p_t(\boldsymbol{x}), p_{\boldsymbol{\theta},t}(\boldsymbol{x})\right) \leq e^{1+2M}\int_0^1\int_\Omega \|\boldsymbol{v}_t(\boldsymbol{\phi}_t(\boldsymbol{x})) - \boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{\phi}_t(\boldsymbol{x}), t)\|^2 p_0(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\mathrm{d}t = e^{1+2M}\mathcal{L}_{\text{RFM}}(\boldsymbol{\theta}). \tag{37}$$

By combining equation (35) and (37), we finally conclude

$$W_2^2\left(p_1(\boldsymbol{x}), p_{\boldsymbol{\theta},1}(\boldsymbol{x})\right) \leq e^{1+2M}\mathcal{L}_{\text{RFM}}(\boldsymbol{\theta}).$$

This finishes the proof.

A.4. Proof of Theorem 3.8

Let $\Omega = \mathbb{R}^d$ and the prior distribution $p_0(\cdot)$ be the standard Gaussian distribution. Under the OT conditional flow, the conditional velocity field is $\boldsymbol{v}_t(\boldsymbol{x}|\boldsymbol{x}_1) = \frac{\boldsymbol{x}_1-(1-\sigma_{\min})\boldsymbol{x}}{1-(1-\sigma_{\min})t}$, and the conditional probability path is $p_t(\boldsymbol{x}|\boldsymbol{x}_1) = \mathcal{N}(t\boldsymbol{x}_1, (1-(1-\sigma_{\min})t)^2\boldsymbol{I})$. Define

$$\bar{p}_t(\boldsymbol{x}_1|\boldsymbol{x}) = \frac{p_t(\boldsymbol{x}|\boldsymbol{x}_1)p_{\text{data}}(\boldsymbol{x}_1)}{p_t(\boldsymbol{x})}.$$

By the definition of $\boldsymbol{v}_t(\boldsymbol{x})$, we have

$$\begin{aligned}
\boldsymbol{v}_t(\boldsymbol{x}) &= \mathbb{E}_{\bar{p}_t(\boldsymbol{x}_1|\boldsymbol{x})}\boldsymbol{v}_t(\boldsymbol{x}|\boldsymbol{x}_1) \\
&= \mathbb{E}_{\bar{p}_t(\boldsymbol{x}_1|\boldsymbol{x})}\frac{\boldsymbol{x}_1-(1-\sigma_{\min})\boldsymbol{x}}{1-(1-\sigma_{\min})t} \\
&= \frac{\boldsymbol{x}}{t} - \frac{1-(1-\sigma_{\min})t}{t}\mathbb{E}_{\bar{p}_t(\boldsymbol{x}_1|\boldsymbol{x})}\frac{\boldsymbol{x}-t\boldsymbol{x}_1}{(1-(1-\sigma_{\min})t)^2} \\
&= \frac{\boldsymbol{x}}{t} + \frac{1-(1-\sigma_{\min})t}{t}\mathbb{E}_{\bar{p}_t(\boldsymbol{x}_1|\boldsymbol{x})}\nabla\log p_t(\boldsymbol{x}|\boldsymbol{x}_1) \\
&= \frac{\boldsymbol{x}}{t} + \frac{1-(1-\sigma_{\min})t}{t}\nabla\log p_t(\boldsymbol{x})
\end{aligned}$$

where we use the fact

$$\mathbb{E}_{\bar{p}_t(\boldsymbol{x}_1|\boldsymbol{x})}\nabla_{\boldsymbol{x}}\log p_t(\boldsymbol{x}|\boldsymbol{x}_1) = \int \frac{p_t(\boldsymbol{x}|\boldsymbol{x}_1)p_{\text{data}}(\boldsymbol{x}_1)}{p_t(\boldsymbol{x})}\frac{\nabla_{\boldsymbol{x}}p_t(\boldsymbol{x}|\boldsymbol{x}_1)}{p_t(\boldsymbol{x}|\boldsymbol{x}_1)}\mathrm{d}\boldsymbol{x}_1 = \frac{1}{p_t(\boldsymbol{x})}\nabla_{\boldsymbol{x}}\int p_t(\boldsymbol{x}|\boldsymbol{x}_1)p_{\text{data}}(\boldsymbol{x}_1)\mathrm{d}\boldsymbol{x}_1 = \nabla_{\boldsymbol{x}}\log p_t(\boldsymbol{x}).$$

This finishes the proof.

B. Implementation Details

B.1. Low-dimensional Toy Examples

For all data sets, we use 6-layer MLPs with 512 channels for the parametrization of the velocity model $\boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}, t)$. We use the sinusoidal positional embedding (Vaswani et al., 2017) with 512 channels for the time step t and adds the time embeddings to the input of each activition function in MLPs. We add a residual block after each linear layer in MLPs. All the activition function is set to be the Gaussian error linear units (GELU) (Hendrycks & Gimpel, 2016). All models are implemented in PyTorch (Paszke et al., 2019) and optimized with the Adam (Kingma & Ba, 2015) optimizer ($\beta_1 = 0.9, \beta_2 = 0.999$). The learning rate is set to be 0.0003 at the beginning, with a decay rate of 0.75 per 10,000 iterations. The results are collected after 200,000 iterations.

B.2. Unconditional Image Generation

The data are rescaled to a hypercube $\Omega = [-1, 1]^{3\times32\times32}$. For FM, we set the prior distribution to be a standard Gaussian distribution and use the OT conditional velocity field. For RFM, we set the prior distribution to be a truncated standard Gaussian distribution over Ω and use the OT conditional velocity field in equation (21). The architecture of the velocity model $\boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}, t)$ in RFM and FM is the same as the UNet (Ronneberger et al., 2015) of score function model in Dhariwal & Nichol (2021) (see Table 5 for details). We use full 32-bit precision for training on CIFAR-10. Following Lipman et al. (2023), the velocity model in RFM is optimized with Adam (Kingma & Ba, 2015) optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay $= 0.0$, and $\epsilon = 10^{-8}$) and a constant learning rate of 0.0002 after a warm-up phase of 5000 training steps. In the warm-up phase, the learning rate is linearly increased from 10^{-8} to the maximum learning rate 0.0002. The results are collected after 800,000 iterations with a batch size of 128. We use the Dormand–Prince method (Dormand & Prince, 1980) with absolute and relative tolerances of 10^{-5} to simulate the reflected CNFs. It costs 1.5 day on 8 Nvidia 2080 Ti GPUs to train reflected CNFs with RFM on CIFAR-10.

B.3. Conditional Image Generation

Similarly to the unconditional image generation tasks, we rescale the data to a hypercube $\Omega = [-1, 1]^{3\times64\times64}$. The prior distribution is set to be the standard Gaussian distribution for FM and the truncated standard Gaussian distribution for RFM.

Table 5. Hyper-parameters used for training each model.

Datasets	CIFAR-10	ImageNet64
Channels	128	192
Depth	2	3
Channels multiple	1,2,2,2	1,2,3,4
Heads	4	4
Heads Channels	64	64
Attention resolution	16	32,16,8
Dropout	0.1	0.1
Effective Batch size	128	2048
Iterations	800k	540k
Learning Rate	2e-4	1e-4
EMA rate	0.9999	0.9999
Learning Rate Scheduler	Constant	Constant
Warmup Steps	5k	5k

Both FM and RFM use the OT conditional velocity field. The class-conditioned velocity field model $\boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}, t, c)$ in RFM and FM is the same as the UNet in the class-conditioned score function model in Dhariwal & Nichol (2021) (see Table 5). We use 16-bit mixed precision for training on ImageNet64. We train FM and RFM using AdamW optimizer (Loshchilov & Hutter, 2018) with a constant learning rate of 0.0001 after a warm-up phase of 5000 steps. In the warm-up phase, the learning rate is linearly increased from 10^{-8} to the maximum learning rate 0.0001. The results are collected after 540,000 iterations with a batch size of 2048. For both FM and RFM, we use the Dormand–Prince method (Dormand & Prince, 1980) with absolute and relative tolerances of 10^{-5} to simulate the (reflected) CNFs. It costs 14 days on 32 Nvidia A100 GPUs to train FM and RFM on ImageNet (64×64).

C. Addtional Experimental Results

C.1. Low-dimensional Toy Examples

For the four 2D constrained domains: hypercube, simplex, half annulus, and cup, we plot the velocity field in reflected CNFs learned by RFM in Figure 5 and the velocity field in CNFs learned by FM in Figure 6. We also report the KL divergences to the ground truth (Table 6) and the constraint violation ratio (Table 7) of samples generated using the Dormand–Prince method (Dormand & Prince, 1980) with absolute and relative tolerances of 10^{-5}. According to Table 8, we see that compared to FM, RFM uses similar NFE on hypercube and simplex, but uses more NFE on half annulus and cup due to the non-optimal transport conditional velocity field.

We further investigate the effect of discretization steps on the constraint violation ratio (Table 9). Table 9 shows that the constraint violation ratio can increase when NFE gets larger. We attribute this to the more time the solver spends exploring around the boundary. When the NFE is small, the samples generated by FM collapse to the interior of the domain, which causes a low constraint violation rate but poor approximation accuracy.

Table 6. KL divergences to the ground truth obtained by different methods on low-dimensional generation tasks. For each method, we calculate the KL divergence using the Python ITE module (Szabó, 2014) with a sample size 50,000. We generate the samples from CNFs and reflected CNFs using the Dormand–Prince method (Dormand & Prince, 1980) with absolute and relative tolerances of 10^{-5}. The results are averaged over 10 independent runs with standard deviation in the brackets.

Method	Hypercube		Simplex		Half Annulus	Cup
	$d=2$	$d=10$	$d=2$	$d=10$	$d=2$	$d=2$
RDM	0.0110(0.0003)	0.4798(0.0006)	0.0055(0.0001)	0.7613(0.0036)	N/A	N/A
FM	0.0049(0.0008)	**0.0247(0.0028)**	0.0093(0.0018)	0.0563(0.0012)	0.0091(0.0008)	0.0132(0.0018)
RFM	**0.0020(0.0013)**	0.0248(0.0027)	**0.0030(0.0012)**	**0.0460(0.0013)**	**0.0032(0.0014)**	**0.0069(0.0019)**

Table 7. Constraint violation ratio (‰) of different methods on low-dimensional generation tasks. We generate the samples from CNFs and reflected CNFs using the Dormand–Prince method (Dormand & Prince, 1980) with absolute and relative tolerances of 10^{-5}.

Method	Hypercube		Simplex		Half Annulus	Cup
	$d = 2$	$d = 10$	$d = 2$	$d = 10$	$d = 2$	$d = 2$
FM	3.2	28.6	12.1	92.8	10.4	15.1
RFM	0.0	0.0	0.0	0.0	0.0	0.0

Table 8. NFE used to samples from different models on low-dimensional generation tasks using the Dormand–Prince method (Dormand & Prince, 1980) with absolute and relative tolerances of 10^{-5}.

Method	Hypercube		Simplex		Half Annulus	Cup
	$d = 2$	$d = 10$	$d = 2$	$d = 10$	$d = 2$	$d = 2$
FM	103	92	111	151	110	119
RFM	117	112	116	99	206	275

Table 9. Constraint violation ratio (‰) and KL divergence with varying NFE on the generation task on Simplex ($d = 2$). Solver: third-order Heun algorithm.

NFE	15	60	300
Constraint violation ratio	0.5	1.7	9.0
KL divergence	0.1805(0.0024)	0.0084(0.0013)	0.0062(0.0011)

C.2. Unconditional Image Generation

We investigate that the effect of the ODE solvers on the FID score in Table 10, where the DOPRI5 achieves the best result. Samples generated by the reflected CNFs trained with RFM on CIFAR-10 are shown in Figure 7. We also test the uniform distribution over the constrained domain as the prior distribution for RFM in Figure 8.

Table 10. FID score obtained by RFM with different ODE solvers (Chen et al., 2018) on CIFAR10.

Solver	Dopri5 (NFE=139)	RK4 (NFE=100)	RK4 (NFE=200)	Heun3 (NFE=150)	Heun3 (NFE=300)
FID	**4.76**	5.37	4.85	4.92	4.81

C.3. Conditional Image Generation

Table 11 reports the FID scores on ImageNet (64×64) conditional generation benchmark. Generally, the FID score of class-conditioned images under high guidance weight can be poor and the generated images (Figure 3) are more informative. We provide a sample quality comparison between FM, RFM, and RDM in Figure 9.

Table 11. FID scores on ImageNet (64×64) conditional generation benchmark (flow guidance weight $w = 15$). Solver: Dopri5 for FM/RFM, RK45 for RDM(ODE), Euler-Maruyama for RDM(SDE).

Method	FM	RDM (ODE)	RDM (SDE)	RFM
NFE	769	871	1000	723
FID	31.83	66.74	35.70	**26.67**

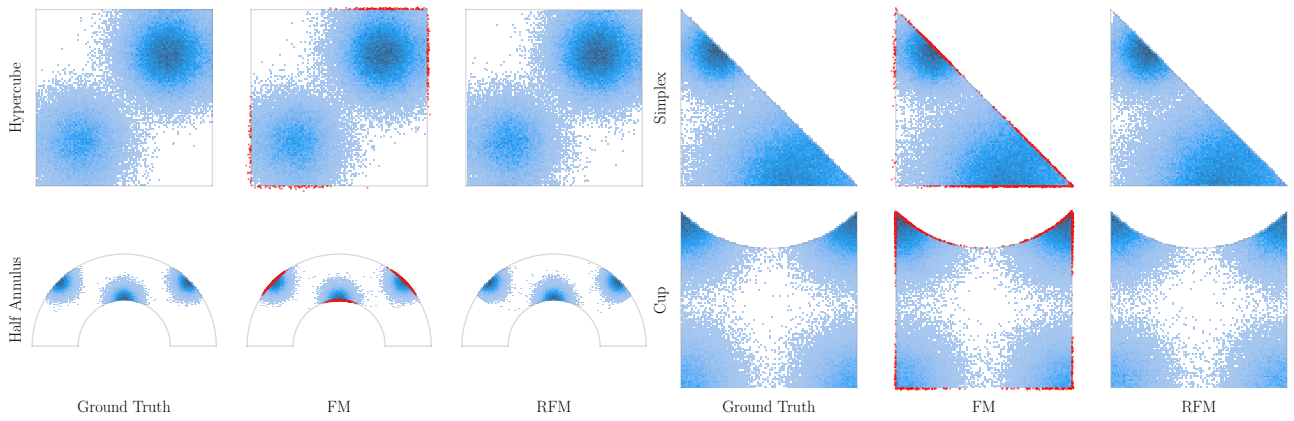

Figure 4. The histplots of samples obtained by different methods compared to the ground truth on the two-dimensional hypercube, simplex, and cup data set. We generate the samples from CNFs and reflected CNFs using the Dormand–Prince method (Dormand & Prince, 1980) with absolute and relative tolerances of 10^{-5}. Samples out of the constrained domain are plotted with red dots. The total sample size is 100,000.

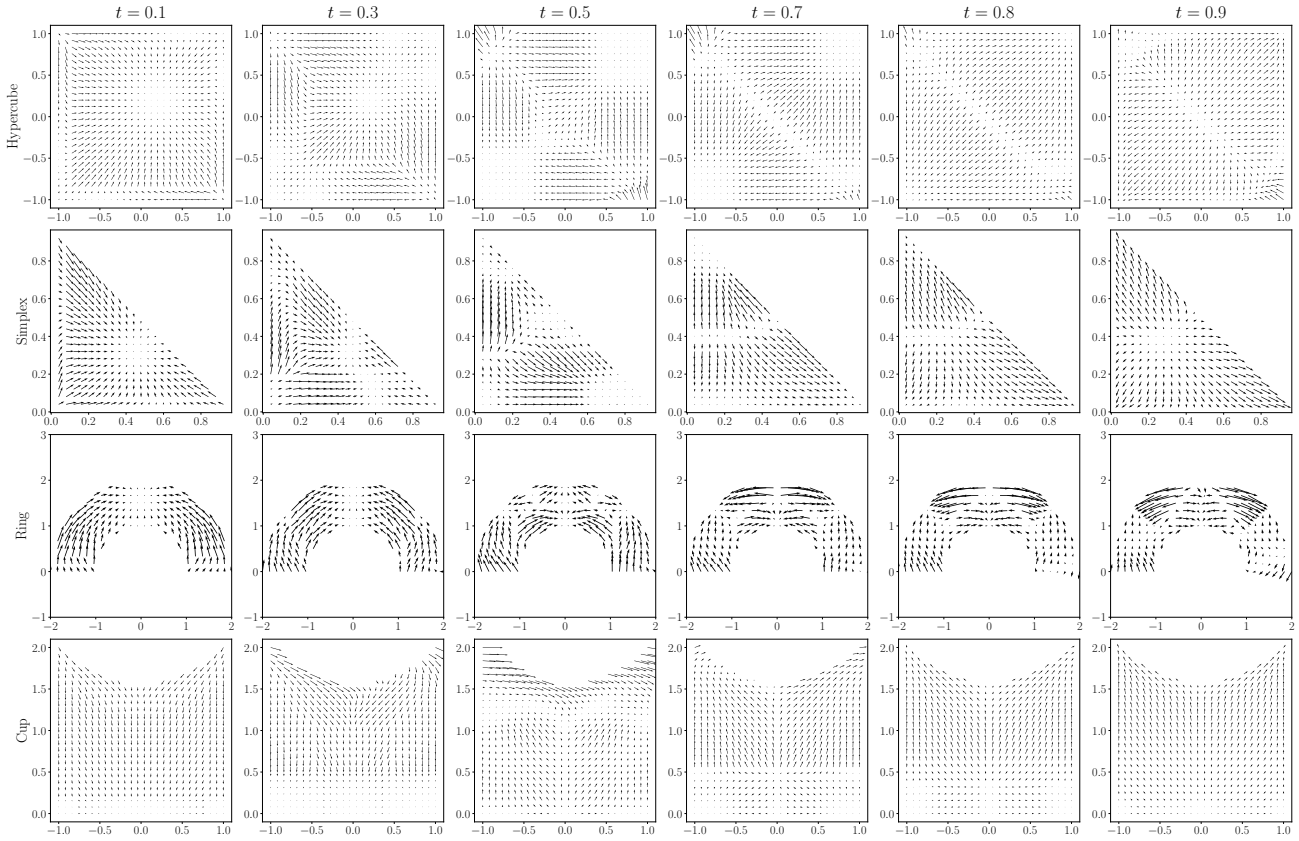

Figure 5. Velocity field in reflected CNFs learned by RFM for different ts on two-dimensional generation tasks.

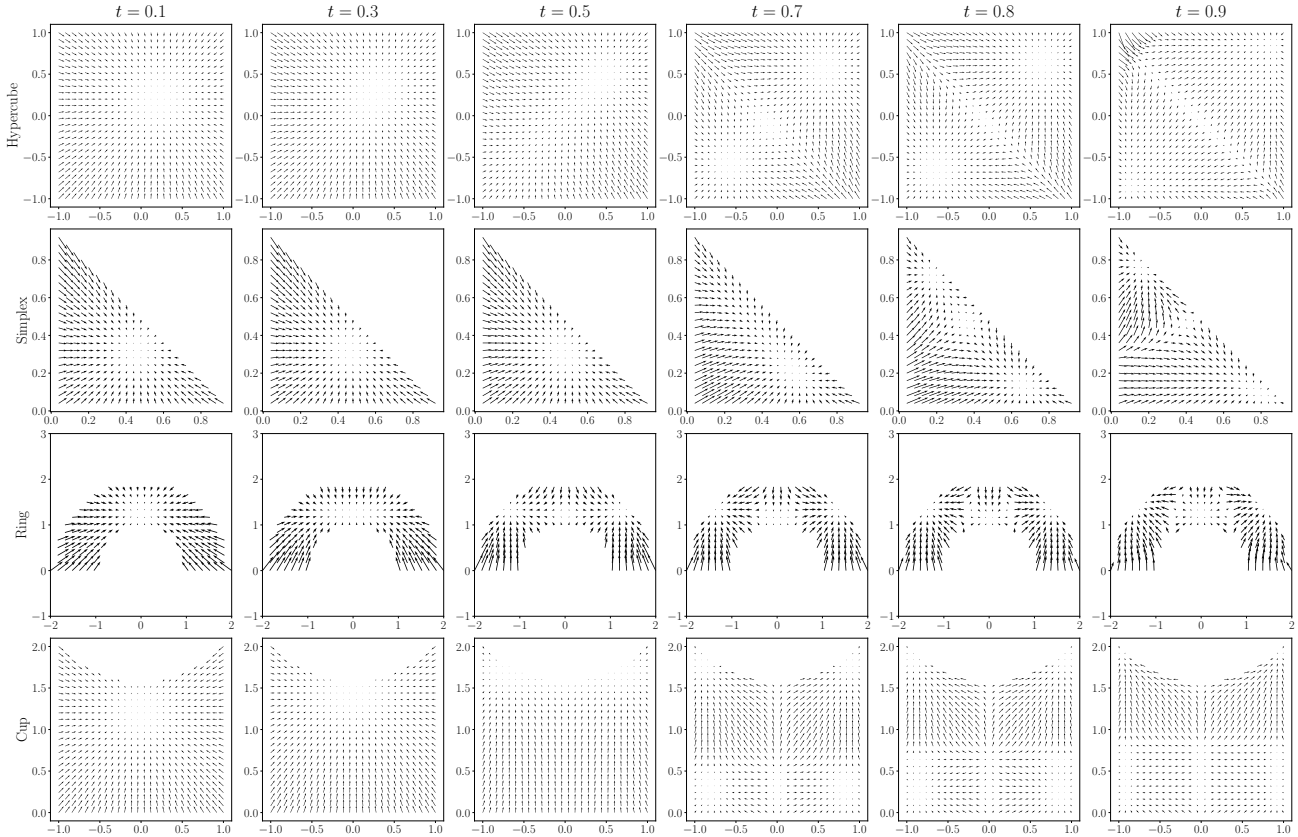

Figure 6. Velocity field in CNFs learned by FM for different ts on two-dimensional generation tasks. We only plot those velocity fields in the constrained domain.

Figure 7. Samples generated by the reflected CNFs trained with RFM on CIFAR-10 (NFE $= 139$). The prior distribution is set to the truncated standard Gaussian distribution over $[-1, 1]^{3\times32\times32}$. The ODE solver is the Dormand–Prince method (Dormand & Prince, 1980) with absolute and relative tolerances of 10^{-5}.

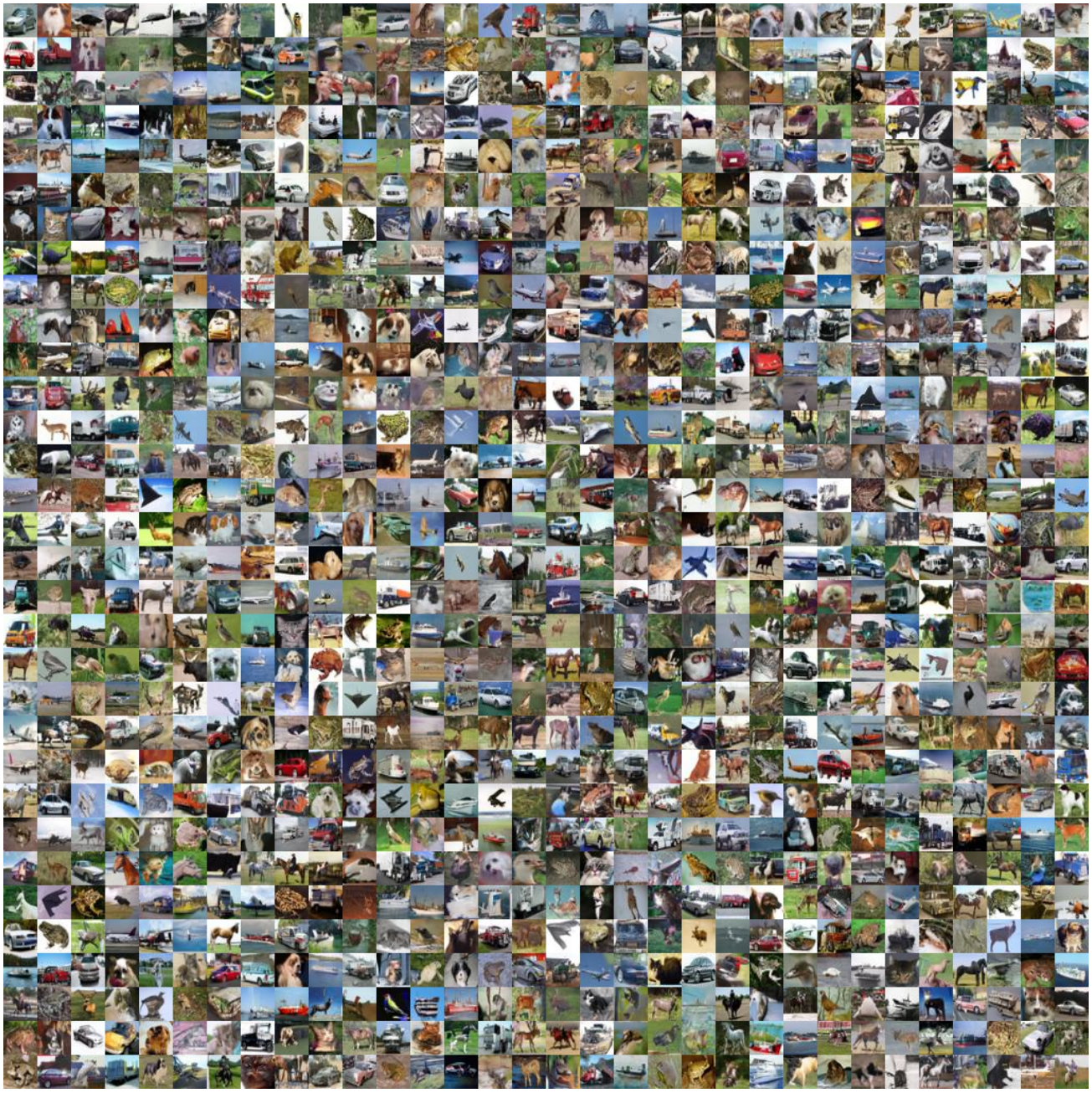

Figure 8. Samples generated by the reflected CNFs trained with RFM on CIFAR-10 (NFE $= 118$). The prior distribution is set to the uniform distribution over $[0, 1]^{3\times32\times32}$. The ODE solver is the Dormand–Prince method (Dormand & Prince, 1980) with absolute and relative tolerances of 10^{-5}.

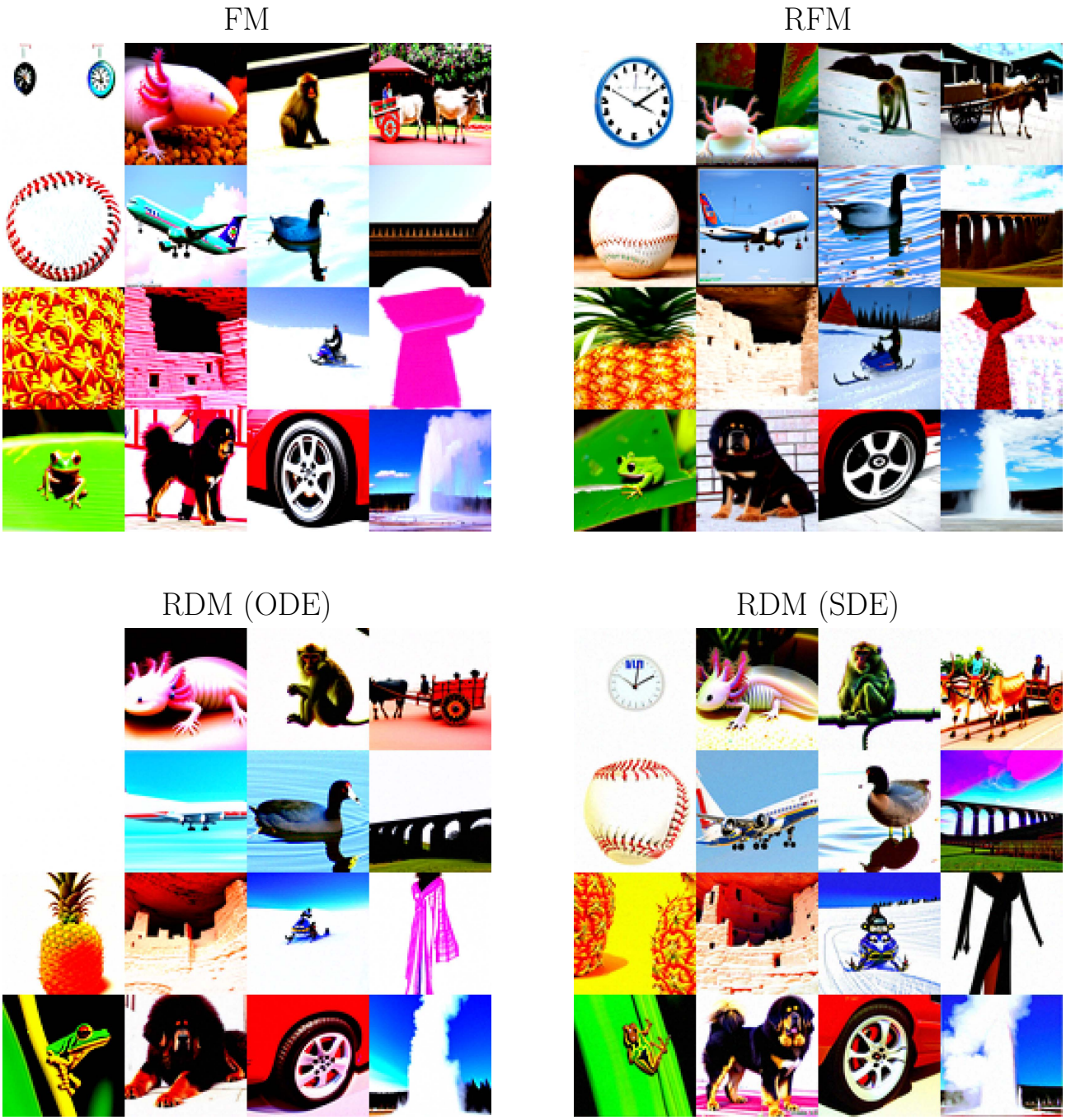

Figure 9. Class-conditioned guided samples with a high guidance weight $w = 15$ on ImageNet (64×64) from different methods, including i) the CNFs trained with FM (upper left, NFE $= 769$, Dormand-Prince method); ii) the reflected CNFs trained with RFM (upper right, NFE $= 723$, Dormand-Prince method); iii) backward ODE in RDM (lower left, NFE $= 871$, Runge–Kutta–Fehlberg method without reflection); iv) backward SDE in RDM (lower right, NFE $= 800$, Euler-Maruyama method with reflection). The samples of i) and iii) are clipped to $[0, 255]$ after the ODE simulation.