EXPLICIT FLOW MATCHING: ON THE THEORY OF FLOW MATCHING ALGORITHMS WITH APPLICATIONS

Gleb Ryzhakov
CAIT, Skolkovo Institute of Science and Technology
Bolshoy Boulevard, 30, p.1,
Moscow 121205, Russia
G.Ryzhakov@skol.tech

Svetlana Pavlova
CAIT, Skolkovo Institute of Science and Technology
Bolshoy Boulevard, 30, p.1, Moscow 121205, Russia
Svetlana.Pavlova@skol.tech

Egor Sevriugov
CAIT, Skolkovo Institute of Science and Technology
Bolshoy Boulevard, 30, p.1,
Moscow 121205, Russia
Egor.Sevriugov@skol.tech

Ivan Oseledets
AIRI, p. 19, Nizhny Susalny per.
5, Moscow, 105064, Russia
and
CAIT, Skolkovo Institute of Science and Technology
Bolshoy Boulevard, 30, p.1, Moscow 121205, Russia
I.Oseledets@skol.tech

ABSTRACT

This paper proposes a novel method, Explicit Flow Matching (ExFM), for training and analyzing flow-based generative models. ExFM leverages a theoretically grounded loss function, ExFM loss (a tractable form of Flow Matching (FM) loss), to demonstrably reduce variance during training, leading to faster convergence and more stable learning. Based on theoretical analysis of these formulas, we derived exact expressions for the vector field (and score in stochastic cases) for model examples (in particular, for separating multiple exponents), and in some simple cases, exact solutions for trajectories. In addition, we also investigated simple cases of diffusion generative models by adding a stochastic term and obtained an explicit form of the expression for score. While the paper emphasizes the theoretical underpinnings of ExFM, it also showcases its effectiveness through numerical experiments on various datasets, including high-dimensional ones. Compared to traditional FM methods, ExFM achieves superior performance in terms of both learning speed and final outcomes.

1 INTRODUCTION

In recent years, there has been a remarkable surge in Deep Learning, wherein the advancements have transitioned from purely neural networks to tackling differential equations. Notably, Diffusion Models Sohl-Dickstein et al. (2015) have emerged as key players in this field. These models transform a simple initial distribution, usually a standard Gaussian distribution, into a target distribution via a solution of Stochastic Differentiable Equation (SDE) Albergo et al. (2023) or Ordinary Differentiable Equation (ODE)Albergo & Vanden-Eijnden (2023) with right-hand side representing a trained neural network. The Conditional Flow Matching (CFM) Lipman et al. (2023) technique, which we focus on in our research, is a promising approach for constructing probability distributions using conditional probability paths, which is notably a robust and stable alternative for training Diffusion Models. The development of the CFM-based approach includes various techniques and heuristics Chen & Lipman (2023); Jolicoeur-Martineau et al. (2023); Pooladian et al. (2023) aimed at improving convergence or quality of learning or inference.For example, in the works Tong et al. (2024a;b); Liu et al. (2023) it was proposed to straighten the trajectories between points by different methods, which led to serious modifications of the learning process. We refer the reader for, example, to the paper Tong et al. (2024b) where different FM-based approaches are summarised, and to the paper Lipman et al. (2023) for the connection between Diffusion Models and CFM.

In our work, we introduce an approach which we called Explicit Flow Matching (ExFM), to consider the Flow Matching framework theoretically by modifying the loss and writing the explicit value of the vector field. Strictly speaking, the presented loss is a tractable form of the FM loss, see Eq. (5) of Lipman et al. (2023). Based on this methods we can improve the convergence of the method in practical examples reducing the variance of the loss, but the main focus of our paper is on theoretical derivations.

Our method allows us to write an expression for the vector field in closed form for quite simple cases (Gaussian distributions), however, we note that Diffusion Models framework in the case of a Gaussian Mixture of two Gaussian as a target distribution is still under investigation, see recent publications Shah et al. (2023); Li et al. (2023).

Our main contributions are:

1. A tractable form of the FM loss is presented, which reaches a minimum on the same function as the loss used in Conditional Flow Matching, but has a smaller variance;
2. The explicit expression in integral form for the vector field delivering the minimum to this loss (therefore for Flow Matching loss) is presented.
3. As a consequence, we derive expressions for the flow matching vector field and score in several particular cases (when linear conditional mapping is used, normal distribution, etc.);
4. Analytical analysis of SGD convergence showed that our formula have better training variance on several cases;
5. Numerical experiments show that we can achieve better learning results in fewer steps.

1.1 PRELIMINARIES

Flow matching is well known method for finding a flow to connect samples from two distribution with densities ρ_0 and ρ_1. It is done by solving continuity equation with respect to the time dependent vector field $\overline{v}(x,t)$ and time-dependent density $\rho(x,t)$ with boundary conditions:

$$\begin{cases} \dfrac{\partial \rho(x,t)}{\partial t} = -\operatorname{div}(\rho(x,t)\overline{v}(x,t)), \\ \rho(x,0) = \rho_0(x), \quad \rho(x,1) = \rho_1(x). \end{cases} \tag{1}$$

Function $\rho(x,t)$ is called *probability density path*. Typically, the distribution ρ_0 is known and it is chosen for convenience reasons, for example, as standard normal distribution $\rho(x) = \mathcal{N}(x \mid 0, I)$. The distribution ρ_1 is unknown and we only know the set of samples from it, so the problem is to approximate the vector field $v(x,t) \approx \overline{v}(x,t)$ using these samples. To make problem (1) well defined, one usually imposes additional regularity conditions on the densities, such as smoothness. The rigorous justification of the obtained results we put in the Appendix, leaving the general formulations of theorems and ideas in the main text.

From a given vector field, we can construct a *flow* ϕ_t, *i. e.*, a time-dependent map, satisfying the ODE $\frac{\partial \phi_t(x)}{\partial t} = v(\phi_t(x), t)$ with initial condition $\phi_0(x) = x$. Thus, one can sample a point x_0 from the distribution ρ_0 and then using this ODE obtain a point $x_1 = \phi_1(x_0)$ which have a distribution approximately equal to ρ_1. For given boundary ρ_0 and ρ_1, the vector field or path solutions are not the only solutions, but if we have found any solution, it will already allow us to sample from the unknown density ρ_1. However, if the problem is more narrowly defined, *e. g.*, one need to have a map that is close to the Optimal Transport (OT) map, we have to impose additional constraints.

The problem of finding any vector field v is solved in conditional manner in the paper Lipman et al. (2023), where so-called Conditional Flow Matching (CFM) is present. Namely, the following loss function was introduced for the training a model v_θ which depends on parameters θ

$$L_{\text{CFM}}(\theta) = \mathbb{E}_t \mathbb{E}_{x_1, x_0} \left\| v_\theta(\phi_{t,x_1}(x_0),\, t) - \phi'_{t,x_1}(x_0) \right\|^2, \tag{2}$$

where $\phi_{t,x_1}(x_0)$ is some flow, conditioned on x_1 (one can take $\phi_{t,x_1}(x_0) = (1-t)x_0 + tx_1 + \sigma_s t x_0$ in the simplest case, where $\sigma_s > 0$ is a small parameter needed for this map to be invertible at any $0 \le t \le 1$). Hereinafter the dash indicates the time derivative: $\phi'_{t,Y}(X) := \frac{\partial}{\partial t}\phi_{t,y}(x)|_{x=X, y=Y}$.

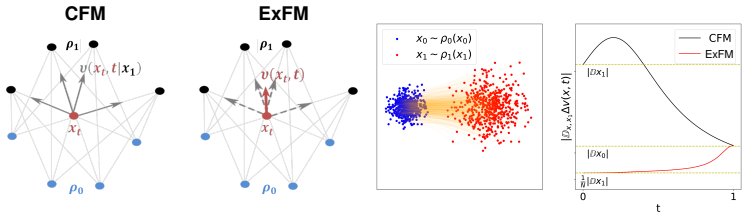


Figure 1: (Left) The key novelty of our approach is that in classical CFM, highly divergent directions can appear in a small spatial area at similar times (left part). In our approach (right part) we average over these vectors, training the model on a smoothed unnoised vector field. (Right) The comparison evaluated variance norm over time parameter t for CFM and ExFM in matching standard Gaussian $\rho_0 = \mathcal{N}(0, I)$ to general Gaussian $\rho_1 = \mathcal{N}(\mu, \sigma^2 I)$ distributions. The y-axis represents the sum of variance vector components, denoted as $|\mathbb{D}_{x,x_1}\Delta v(x,t)|$. The left panel illustrates samples drawn from the ρ_0 and ρ_1 distributions, as well as the corresponding flows. The right panel depicts the variance trend over time for both CFM (black line) and ExFM (red line) objectives. The dotted lines correspond to the variance levels (in top-down order $|\mathbb{D}x_1|$, $|\mathbb{D}x_0|$, $|\mathbb{D}x_1|/N$.

Time variable t is uniformly distributed: $t \sim \mathcal{U}[0, 1]$ and random variables x_0 and x_1 are distributed according to the initial and final distributions, respectively: $x_0 \sim \rho_0$, $x_1 \sim \rho_1$. Below we omit specifying of the symbol $\mathbb{E}$ the distribution by which the expectation is taken where it does not lead to ambiguity.

1.2 Why new method?

Model training using loss (2) have the following disadvantage: during training, due to the randomness of x_0 and x_1, significantly different values can be presented for model as output value at close model argument values (x_t, t). Indeed, a fixed point $x_t = \phi_{t,x_1}(x_0)$ can be obtained by an infinite set of x_0 and x_1 pairs, some of which are directly opposite, and at least for small times t the probability of these different directions may not be significantly different. At the same time, data $\phi'_{t,x_1}(x_0)$ on which the model learns significantly different for such different positions of pairs x_0 and x_1. Thus, the model is forced to do two functions during training: generalize and take the mathematical expectation (clean the data from noise).

In our approach, see Fig. 1(a), we feed the model input with cleaned data with small variance. Thus, the model only needs to generalize the data, which happens much faster (in fewer training steps).

Moreover, in the process of constructing the modified loss, we have developed the exact formula for the vector field, see Eq. (11), (34). The existence of an explicit formula for the vector field is of great importance not only from a theoretical but also from a practical point of view.

2 Main idea

2.1 Modified objective

Lets expand the last two mathematical expectations in the loss (2) and substitute variables using the map ϕ_{t,x_1}, passing from the point x_0 to its position $x_t = \phi_{t,x_1}(x_0)$ at time t:

$$\mathbb{E}_{x_1,x_0}\left\|v_\theta(\phi_{t,x_1}(x_0),\, t) - \phi'_{t,x_1}(x_0)\right\|^2 = \iint \left\|v_\theta(\phi_{t,x_1}(x_0),\, t) - \phi'_{t,x_1}(x_0)\right\|^2 \rho_0(x_0)\rho_1(x_1)\mathrm{d}x_0\mathrm{d}x_1$$

$$= \iint \left\|v_\theta(x_t,\, t) - \phi'_{t,x_1}\left(\phi^{-1}_{t,x_1}(x_t)\right)\right\|^2 \underbrace{\det\left[\partial\phi^{-1}_{t,x_1}(x)/\partial x\,\Big|_{x=x_t}\right]\rho_0\left(\phi^{-1}_{t,x_1}(x_t)\right)}_{\rho_{x_1}(x_t,t)}\,\rho_1(x_1)\,\mathrm{d}x_t\,\mathrm{d}x_1$$

$$= \mathbb{E}_{x_1,x_t\sim\rho_{x_1}(\cdot,t)}\left\|v_\theta(x_t,\, t) - \phi'_{t,x_1}\left(\phi^{-1}_{t,x_1}(x_t)\right)\right\|^2. \quad (3)$$

We assume, that the map ϕ_{t,x_1} is invertible at each $0 < t < 1$, *i. e.* that $\phi^{-1}_{t,x_1}(x_t)$ exits on this time interval and for all $x_t = \{\phi_t(x_0) \mid \forall x_0 : \rho(x_0) > 0\}$. Eq. (3) can be seen as a transition from expectation on the variable $x_0 \sim \rho_0$ to expectation on the variable $x_t \sim \eta_t(\cdot; x_1)$, where $\eta_t(x; x_1) = [\phi_{t,x_1}]_* \rho_0(x) := \rho_0\big(\phi^{-1}_{t,x_1}(x)\big) \det\big[\partial \phi^{-1}_{t,x_1}(x)/\partial x\big]$. See paper Chen et al. (2018) for details about the push-forward operator "*". Our representation (3) is very similar to expression (9) of the cited paper Lipman et al. (2023), only we write it in terms of the conditional flow rather than the conditional vector field.

To obtain the modified loss, we return to the expectation form of the standard CFM loss representation in (3). It is written as the expectation over two random variables x_1 and x_t having a common distribution density

$$\{x_1, x_t\} \sim \rho_{\text{jnt}}(x_1, x_t, t) = \eta_t(x_t; x_1)\rho_1(x_1), \tag{4}$$

which, generally speaking, is not factorizable. Let us rewrite this expectations in terms of two independent random variables, each of which have its marginal distribution. The marginal distribution $\hat{\rho}_t$ of x_t can be obtained via integration:

$$\hat{\rho}_t(x_t) = \int \rho_{\text{jnt}}(x_1, x_t, t)\, \mathrm{d}x_1 = \int \eta_t(x_t; x_1)\rho_1(x_1)\, \mathrm{d}x_1\,, \tag{5}$$

while the marginal distribution of x_1 is just (unknown) function ρ_1. Let for convenience $w(t, x_1, x) = \phi'_{t,x_1}\big(\phi^{-1}_{t,x_1}(x)\big)$[1]. We have

$$\begin{aligned} L_{\text{CFM}}(\theta) = \mathbb{E}_{t,x_1,x_t\sim\eta_t(\cdot;x_1)} \|v_\theta(x_t,\, t) - w(t, x_1, x_t)\|^2 = \\ \int_0^1 \iint \|v_\theta(x_t,\, t) - w(t, x_1, x_t)\|^2 \eta_t(x; x_1)\rho_1(x_1)\, \mathrm{d}x_t\, \mathrm{d}x_1 \mathrm{d}t = \\ \int_0^1 \iint \|v_\theta(x_t,\, t) - w(t, x_1, x_t)\|^2 \, \big(\eta_t(x_t;x_1)/\hat{\rho}_t(x_t)\big)\, \hat{\rho}_t(x_t)\rho_1(x_1)\, \mathrm{d}x_t\, \mathrm{d}x_1 \mathrm{d}t = \\ \mathbb{E}_{t,x_1,x\sim\hat{\rho}_t(\cdot)} \|v_\theta(x,\, t) - w(t, x_1, x)\|^2\, \xi_t(x; x_1)/\rho_1(x_1), \end{aligned} \tag{6}$$

where we introduce a conditional distribution

$$\xi_t(x; x_1) := \eta_t(x; x_1)\rho_1(x_1)/\hat{\rho}_t(x) := \eta_t(x; x_1)\rho_1(x_1) \Big/ \int \eta_t(x; x_1)\rho_1(x_1)\, \mathrm{d}x_1. \tag{7}$$

The key feature of the representation (6) is that the integration variables x_1 and x are independent. Thus, we can evaluate them using Monte Carlo-like schemes in different ways. However, we go further and make a modification to this loss to reduce the variance of Monte Carlo methods.

2.2 New loss and exact expression for vector field

Note that so far the expression for L_{CFM} have not changed, it has just been rewritten in different forms. Now we change this expression so that its numerical value, generally speaking, may be different, but the derivative of the model parameters will be the same. We introduce the following loss

$$\begin{aligned} L_{\text{ExFM}}(\theta) = \mathbb{E}_t \mathbb{E}_{x\sim\hat{\rho}_t} \Big\| v_\theta(x,\, t) - \mathbb{E}_{x_1\sim\rho_1} w(t, x_1, x)\xi_t(x; x_1)/\rho_1(x_1) \Big\|^2 = \\ \int_0^1 \int \Big\| v_\theta(x,\, t) - \int w(t, x_1, x) \times \xi_t(x; x_1)\, \mathrm{d}x_1 \Big\|^2 \hat{\rho}_t(x)\, \mathrm{d}x\, \mathrm{d}t. \end{aligned} \tag{8}$$

Theorem 2.1. *Losses L_{CFM} in Eq.* (2) *and L_{ExFM} in Eq.* (8) *have the same derivative with respect to model parameters:*

$$\mathrm{d}L_{CFM}(\theta)/\mathrm{d}\theta \,=\, \mathrm{d}L_{ExFM}(\theta)/\mathrm{d}\theta\,. \tag{9}$$

Proof is in the Appendix A.1.

In the presented loss L_{ExFM}, the integration (outside the norm operator) proceeds on those variables on which the model depends, while inside this operator there are no other free variables. Thus, using

[1]Note, that $w(t, x_1, x)$ is the conditional velocity at the given point x.

this kind of loss, it is possible to find an exact analytical expression for the vector field for which the minimum of this loss is zero (unlike the loss L_{CFM}). Namely, we have

$$v(x,t) = \int w(t,x_1,x)\xi_t(x;x_1)\,\mathrm{d}x_1\,. \tag{10}$$

We can obtain the exact form of this vector field given the particular map ϕ_{t,x_1}. For example, the following statement holds:

Corollary 2.2. *Consider the linear conditioned flow $\phi_{t,x_1}(x_0) = (1-t)x_0 + tx_1$ which is invertible as $0 \leq t < 1$.*

Then $w(t,x_1,x) = \frac{x_1 - x}{1-t}$, $\rho_{x_1}(x,t) = \rho_0\left(\frac{x - x_1 t}{1-t}\right)\frac{1}{(1-t)^d}$ and the loss L_{ExFM} in Eq. (8) *reaches zero value when the model of the vector field has the following analytical form*

$$v(x,t) = \int (x_1 - x)\rho_0\left(\frac{x - x_1 t}{1-t}\right)\rho_1(x_1)\,\mathrm{d}x_1 \Bigg/ \left((1-t)\int \rho_0\left(\frac{x - x_1 t}{1-t}\right)\rho_1(x_1)\,\mathrm{d}x_1\right). \tag{11}$$

This is the exact value of the vector field whose flow translates the given distribution ρ_0 to ρ_1.

Complete proofs are in the Appendix A.3.1. Note that the result (11) is not totally new, for example, a similar result (though in the form of a general expression rather than an explicit formula), was given in Tong et al. (2024a), Eq. (9). However, our contribution consists of both the general form (10) and practical and theoretical conclusions from it (see below).

Remark 2.3. In the case of the initial and final times $t = 0,\ 1$, Eq. (11) is noticeably simpler

$$v(x,0) = \mathbb{E}_{x_1} x_1 - x = \int x_1\rho_1(x_1)\,\mathrm{d}x_1 - x. \quad v(x,1) = x - \int x_0\rho_0(x_0)\,\mathrm{d}x_0\,. \tag{12}$$

This expression for the initial velocity means that each point first tends to the center of mass of the unknown distribution ρ_1 regardless of its initial position.

Extensions to SDE Now let the conditional map be stochastic: $\phi_{t,x_1} = (1-t)x_0 + tx_1 + \sigma_e(t)\epsilon$, where $\epsilon \sim \mathcal{N}(0,1)$. Typically, $\sigma_e(0) = \sigma_e(1) = 0$, for example, $\sigma_e(t) = t(1-t)\sigma_e$.

Note that this formulation covers (with appropriate selection of the $\sigma_e(t)$ parameter) the case of diffusion models Tong et al. (2024b).

Then, we can write the exact solution for a so-called *score and flow matching* objective (see Tong et al. (2024b) for details)

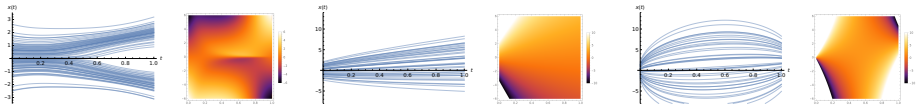
$$\mathcal{L}_{[\text{SF}]^2\text{M}}(\theta) = \mathbb{E}\big[\underbrace{\|v_\theta(x,t) - u_t^\circ(x)\|^2}_{\text{flow matching loss}} + \lambda(t)^2\underbrace{\|s_\theta(x,t) - \nabla\log p_t(x)\|^2}_{\text{score matching loss}}\big].$$

that corresponds to this map. In the last expression, the following explicit conditional expressions are considered in the cited paper for the case $\sigma_e(t) = \sqrt{t(1-t)}\sigma_e$

$$u_t^\circ(x) = \frac{1-2t}{t(1-t)}(x - (tx_1 + (1-t)x_0)) + (x_1 - x_0),\ \ \nabla\log p_t(x) = \frac{tx_1 + (1-t)x_0 - x}{\sigma_e^2 t(1-t)}.$$

The exact solution (our result, explicit analog of the Eq. (10) from Tong et al. (2024b)) under consideration has the form (44) and (46) and, for example for the for the Gaussian ρ_0 this expressions reduced to the Eq. (49) and (50), correspondingly. See Appendix E for the details on this case.

Simple examples Consider the case of Standard Normal Distribution as ρ_0 and Gaussian Mixture of two Gaussians as ρ_1. Vector field have a closed form (37) in this case, and we can fast numerically solve ODE for trajectories. Random generated trajectories and plot of the vector field are shown on Fig. 2 (a)–(b). Detailed explanation of this case is in the Sec. D.2. Another example is related to the case of a stochastic map in the form of Brownian Bridge, which briefly described in the last paragraph and considered in Sec. E.3.2 in details, see Fig. 2 (c)–(f). Note that at some σ_e values the trajectories are a little bit straightened in this case compared to the usual linear map, if we compare cases on the Fig. 6.



(a) GM trajectories (b) GM VF (c) BB Trajectories, $\sigma_e = 3$ (d) BB VF, $\sigma_e = 3$ (e) BB Trajectories, $\sigma_e = 10$ (f) BB VF, $\sigma_e = 10$

Figure 2: Trajectories and vector field obtained in simple cases: (a) $N = 80$ random trajectories from $\mathcal{N}\left(\cdot \middle| 0, 1^2\right)$ to GM; (b) 2D plot of the vector field in this case (c)–(f) $N = 40$ random trajectories from $\mathcal{N}\left(\cdot \middle| 0, 1^2\right)$ to $\mathcal{N}\left(\cdot \middle| 2, 3^2\right)$ and 2D plot of the vector field for different σ_e for the Brownian Bridge map

2.3 Training scheme based on the modified loss

Let us consider the difference between our new scheme based on loss L_{ExFM} and the classical CFM learning scheme. As a basis for the implementation of the learning scheme, we take the open-source code[2] from the works Tong et al. (2024b;a).

Consider a general framework of numerical schemes in classical CFM. We first sample m random time variables $t \sim \mathcal{U}[0, 1]$. Then we sample several values of x. To do this, we sample a certain number n samples $\{x_0^i\}_{i=1}^n$ from the "noisy" distribution ρ_0, and the same number n of samples $\{x_1^i\}_{i=1}^n$ from the unknown distribution ρ_1. Then we pair them (according to some scheme), and get n samples as $x^{j,i} = \phi_{t^j, x_1^i}(x_0^i)$ (*e. g.* a linear combination in the simple case of linear map: $x^{j,i} = (1 - t^j)x_0^i + t^j x_1^i$), $\forall i = 1, 2, \ldots, n$; $\forall j = 1, 2, \ldots, m$. Note, than one of the variable n or m (or both) can be equal to 1.

At the step 2, the following discrete loss is constructed from the obtained samples

$$L^d_{\text{CFM}}(\theta) = \sum_{j=1}^{m} \sum_{i=1}^{n} \left\| v_\theta(x^{j,i},\, t^j) - \phi'_{t^j, x_1^i}(x_0^i) \right\|^2. \tag{13}$$

Finally, we do a standard gradient descent step to update model parameters θ using this loss.

The first and last step in our algorithm is the same as in the standard algorithm, but the second step is significantly different. Namely, we additionally generate a sufficiently large number $N \gg n \cdot m$ of samples $\overline{x}_1$ from the unknown distribution ρ_1, sampling $(N - n)$ new samples and adding to it the samples $\{x_1^i\}_1^n$ that are already obtained on the previous step.

Then we form the following discrete loss which replaces the integral on x_1 in L_{ExFM} by its evaluation v^d by self-normalized importance sampling or rejection sampling (see Appendix B for details)

$$L^d_{\text{ExFM}}(\theta) = \sum_{j=1}^{m} \sum_{i=1}^{n} \left\| v_\theta(x^{j,i},\, t^j) - v^d(x^{j,i},\, t^j) \right\|^2. \tag{14}$$

For example, if we use self-normalized importance sampling (SIS)[3] and assume that the Jacobian $\det\left[\partial \phi^{-1}_{t,x_1}(x) / \partial x\right]$ do not depend on x_1, we can write

$$v^d(x,\, t) = \left(\sum_{k=1}^{N} w(t, \overline{x}_1^k, x) \rho_0\left(\phi^{-1}_{t,\overline{x}_1^k}(x)\right) \right) \Bigg/ \sum_{k=1}^{N} \rho_0\left(\phi^{-1}_{t,\overline{x}_1^k}(x)\right). \tag{15}$$

Theorem 2.4. *Under mild conditions, the error variance of the integral gradient* (9) *using the Monte Carlo method* (14) *is lower than using formula* (13) *with the same number $n \cdot m$ of samples for $\{x\}$.*

Sketch of the proof is in the Appendix A.2. The steps of our scheme are formally summarized in Algorithm 1.

[2] `https://github.com/atong01/conditional-flow-matching`

[3] SIS may be biased. To avoid this issue we also use rejection sampling to integral estimation, see App. B

Particular case of linear map and Gaussian noise Let ϕ_{t,x_1} be the linear flow: $\phi_{t,x_1}(x_0) = (1-t)x_0 + tx_1$. and consider the case of standard normal distribution for the initial density ρ_0: $\rho_0(x) \sim \mathcal{N}(x \mid 0, I)$. Then in the case of using self-normalized importance sampling, we have

$$v^d(x,\, t) = \sum_{k=1}^{N} \frac{\overline{x}_1^k - x}{1-t} \big(\mathrm{SoftMax}(Y^1,\, \ldots,\, Y^N)\big)_k, \quad \text{where} \quad Y^k = -\frac{1}{2}\frac{\left\|x - t\cdot \overline{x}_1^k\right\|^2_{\mathbb{R}^d}}{1-t}. \tag{16}$$

Here, the lower index k in $\mathrm{SoftMax}$ stands for the k-th component, and the $\mathrm{SoftMax}$ operation itself came about due to exponents in the Gaussian density as a more stable substitute for computing than directly through exponents.

Extension of other maps and initial densities ρ_0 Common expression (10) can be reduced to closed form for the particular choices of density ρ_0 and map ϕ (consequently, expression for w). We summarise several known approaches for which FM-based techniques can be applied in Table 1[4]. See Appendix C and D for derivations of formulas and for more extensions.

Table 1: Correspondence between some methods which can reduced to FM framework and our theoretical descriptions of them.

Probability Path	$q(z)$	$\mu_t(z)$	σ_t	Explicit expressions: vector field (VF) and score (S)
Var. Exploding Song & Ermon (2019)	$\rho_1(x_1)$	x_1	σ_{1-t}	VF: (32)
Var. Preserving Ho et al. (2020)	$\rho_1(x_1)$	$\alpha_{1-t}x_1$	$\sqrt{1-\alpha^2_{1-t}}$	VF: (31)
Flow Matching Lipman et al. (2023)	$\rho_1(x_1)$	tx_1	$t\sigma_s - t + 1$	VF: (11) if $\sigma = 0$; and (26)
Independent CFM	$\rho_0(x_0)\rho_1(x_1)$	$tx_1 + (1-t)x_0$	σ	VF: (10)
Schrödinger Bridge CFM Tong et al. (2024b)	$\rho_0(x_0)\rho_1(x_1)$	$tx_1 + (1-t)x_0$	$\sigma\sqrt{t(1-t)}$	Can be obtained by SDE using VF: (49), S:(50)

Complexity We assume that the main running time of the algorithm is spent on training the model, especially if it is quite complex. Thus, the running time of one training step depends crucially on the number $n \cdot m$ of samples $\{x\}$ and it is approximately the same for both algorithms: the addition of points $\overline{x}_1$ entails only an additional calculation using formula (16), which can be done quickly and, moreover, can be simple parallelized.

2.4 Irreducible variance of gradient for CFM optimization

Ensuring the stability of optimization is vital. Let $\Delta\theta$ be changes in parameters, obtained by SGD with step size $\gamma/2$ applied to the functional from Eq. (13):

$$\Delta v(x^{j,i}, t^j) = -\gamma \cdot \big(v(x^{j,i}, t^j) - v^d(x^{j,i},\, t^j)\big). \tag{17}$$

For simplification, we consider a function, $v_\theta(x,t)$, capable of perfectly fitting the CFM problem and providing an optimal solution for any point x and time t. For a linear conditional flow at a specific point $x^{j,i} \sim \eta_{t^j}(\cdot; x_1^i)$ at time $t^j \sim U(0,1)$, the update $\Delta v(x^{j,i}, t^j)$ can be represented as follows:

$$\Delta v(x^{j,i}, t^j) = \gamma\left(x_1^i - \hat{x}_0^i - v(x^{j,i}, t^j)\right), \tag{18}$$

where $\hat{x}_0^i = \frac{x^{j,i} - t^j x_1^i}{1-t^j}$. We define the variance $\mathbb{D}_{x,x_1} f(x, x_1)$ for $x \sim \eta_t(\cdot; x_1)$ and $x_1 \sim \rho_1$ as:

$$\mathbb{D}_{x,x_1} f(x,x_1) = \mathbb{E}_{x,x_1} f^2(x,x_1) - (\mathbb{E}_{x,x_1} f(x,x_1))^2. \tag{19}$$

Proposition 2.5. *At the time $t = 0$, the variance of update in the form* (18) *have the following element-wise lower bound:*

$$\mathbb{D}_{x^{j,i},x_1^i} \Delta v(x^{j,i}, 0) = \gamma^2 \mathbb{D}_{x_1^i} x_1^i + \gamma^2 \mathbb{D}_{x^{j,i},x_1^i}(x^{j,i} + v(x^{j,i}, 0)) \geq \gamma^2 \mathbb{D}_{x_1^i} x_1^i.$$

Equality is reached when the model $v(x^{j,i}, 0)$ has exact values equal to (12).

[4]The idea and common structure of the Table is taken from Tong et al. (2024b)

Given that the variance cannot be reduced with an increase in batch size, the only available option is to decrease the step size of the optimization method, *i. e.*, reduce the learning rate slowing down the convergence. The situation is much better for the proposed loss in (14). We can express the update $\Delta v(x^{j,i}, t^j)$ in the case of ExFM objective as:

$$\Delta v(x^j, t^j) = \gamma^2 \Big(\sum_{k=1}^{N} x_1^k \tilde{\rho}\left(x^{j,i}; x_1^k, t^j\right) - x^{j,i} - v(x^{j,i}, t^j) \Big), \tag{20}$$

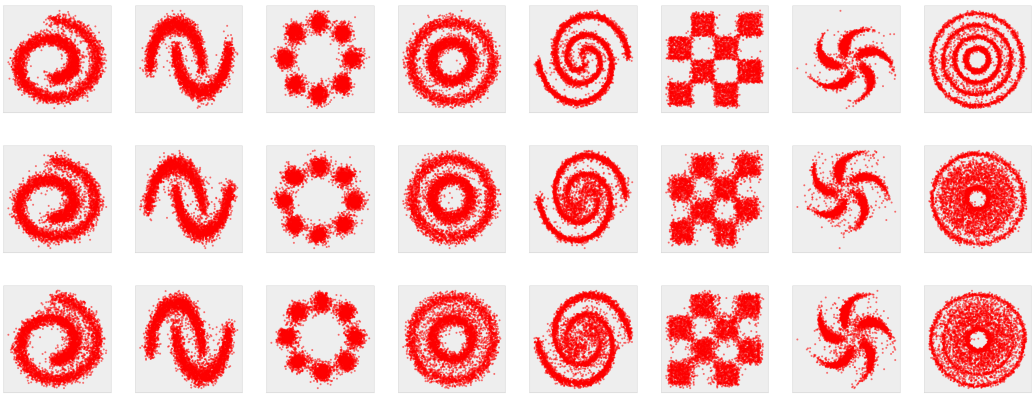
where $x^{j,i} \sim \eta_{t^j}(\cdot; x_1^i)$, $x_1^k \sim \rho_1$ and $\tilde{\rho}\left(x^{j,i}; x_1^k, t^j\right) = \rho_0\left(\frac{x^{j,i} - t^j x_1^k}{1-t^j}\right) / \sum_{k=1}^{N} \rho_0\left(\frac{x^{j,i} - t^j x_1^k}{1-t^j}\right)$. Similar to the derivations in the previous part, we can found simplified form for the variance of update at $t = 0$.

Proposition 2.6. *At the time $t = 0$, the variance of update from* (20) *have the following element-wise lower bound:*

$$\mathbb{D}_{x^{j,i}, x_1^k} \Delta v(x^{j,i}, 0) = \frac{\gamma^2}{N} \mathbb{D}_{x_1^k} x_1^k + \gamma^2 \mathbb{D}_{x^{j,i}, x_1^k}(x^{j,i} + v(x^{j,i}, 0)) \geq \frac{\gamma^2}{N} \mathbb{D}_{x_1^k} x_1^k.$$

Equality is reached when the model $v(x^{j,i}, 0)$ has exact values equal to (12).

In comparison to CFM, the variance of the update is N times smaller than the variance of the target distribution and could be controlled without impeding convergence by adjusting the number of samples N. In Figure 1(b), we visually compare the variances of CFM and ExFM. The illustration aligns a standard normal distribution $\mathcal{N}(0, I)$ with a shifted and scaled variant $\mathcal{N}(\mu, I\sigma^2)$. ExFM yields lower variance throughout the range $t \in [0, 1]$. Detailed analytical calculations of the optimal velocity $v(x, t)$ and variance are provided in the Appendix G.



(a) swissroll (b) moons (c) 8gaussians (d) circles (e) 2spirals (f) checkerboard (g) pinwheel (h) rings

Figure 3: Visual comparison of methods on toy 2D data. First row sampled by ExFM, second row sampled by CFM, third row sampled by OT-CFM.

3 Numerical Experiments

For the foundation of our experiments, we adopted the framework from the open-source code[5]. Specifically, the implementations of CFM and OT-CFM were based on the works Tong et al. (2024b;a). It is important to note that, to our knowledge, the original authors of Lipman et al. (2023) did not publicly release their code. As a result, we elected to utilize the implementation from Tong et al. (2024b) due to its close adherence to the original framework.

[5] https://github.com/atong01/conditional-flow-matching

Table 2: Wasserstein distance comparison for ExFM, CFM and OT-CFM methods for 2D-toy datasets for 15 000 learning steps (30 000 learning steps for `rings` dataset) mean and std taken from 10 sampling iterations.

DATA	EXFM	CFM	OT-CFM
SWISSROLL	**5.95e-02 ± 4.3e-03**	8.68E-02 ± 7.3E-03	6.98E-02 ± 6.1E-03
MOONS	**4.87e-02 ± 4.7e-03**	6.80E-02 ± 8.2E-03	5.94E-02 ± 6.3E-03
8GAUSSIANS	**8.83e-02 ± 1.41e-02**	1.12E-01 ± 1.4E-02	1.00E-01 ± 1.5E-02
CIRCLES	**6.70e-02 ± 3.3e-03**	8.51E-02 ± 3.4E-03	8.47E-02 ± 6.9E-03
2SPIRALS	**6.94e-02 ± 9.5e-03**	1.01E-01 ± 6E-03	1.08E-01 ± 2E-02
CHECKERBOARD	**1.14e-01 ± 1.1e-02**	1.59E-01 ± 1.4E-02	1.22E-01 ± 1.5E-02
PINWHEEL	**6.52e-02 ± 5.9e-03**	1.13E-01 ± 1.1E-02	8.08E-02 ± 5.8E-03
RINGS	**6.35e-02 ± 4.4e-03**	1.16E-01 ± 4E-03	1.08E-01 ± 3E-03

Table 3: NLL comparison for ExFM, CFM and OT-CFM methods for tabular datasets for 10 000 learning steps, mean and std taken from 10 sampling iterations.

DATA	EXFM	CFM	OT-CFM
POWER	**-8.51e-02 ± 4.85e-02**	1.64E-01 ± 4.2E-02	5.22E-02 ± 3.92E-02
GAS	**-5.53e+00 ± 4e-02**	-5.00E+00 ± 3E-02	-5.48E+00 ± 3E-02
HEPMASS	**2.16e+01 ± 6e-02**	2.21E+01 ± 6E-02	**2.16e+01 ± 4e-02**
BSDS300	-1.29E+02 ± 8E-01	-1.29E+02 ± 9E-01	**-1.32e+02 ± 6e-01**
MINIBOONE	**1.34e+01 ± 2e-04**	1.42E+01 ± 1E-04	1.43E+01 ± 9E-05

Toy 2D data We conducted unconditional density estimation among eight distributions. Additional details of the experiments see in the Appendix H. We commence the exposition of our findings by showcasing a series of classical 2-dimensional examples, as depicted in Fig. 3 and Table 2. Our observations indicate that ExFM adeptly handles complex distribution shapes is particularly noteworthy, especially considering its ability to do so within a small number of learning steps. Additionally, the visual comparison underscores the evident superiority of ExFM over the CFM and OT-CFM approaches.

Tabular data We conducted unconditional density estimation on five tabular datasets, namely `power`, `gas`, `hepmass`, `minibone`, and `BSDS300`. Additional details of the experiments see in the Appendix H. The empirical findings obtained from the numerical experiments from Table 3 indicate a statistically significant improvement in the performance of our proposed method. Notably, ExFM demonstrates a notable acceleration in convergence rate.

High-dimensional data and additional experiments We conducted experiments on high-dimensional data, among them experiments on CIFAR10 and MNIST dataset. FID results on CIFAR10 shows slightly better score among sampled images. Additional details of the experiments and sampled images see in the Appendix H.

Stochastic ExFM (ExFM-S) on toy 2D data We evaluated the performance of the stochastic version of ExFM (ExFM-S) with use of expressions given in Sec. E.3.2 on four standard toy datasets. The primary experimental setup follows that used in Tong et al. (2024a). Additional details on the hyperparameters used are available in Appendix H. Based on the findings presented in Table 4, we determine that ExFM-S surpasses I-CFM on all four datasets in terms of generative performance ($\mathcal{W}_2$) and also outperforms in terms of OT optimality (NPE) on two of them, exhibiting similar results on the remaining datasets. It also demonstrates performance similar to OT-CFM. While ExFM-S is not as robust as the basic ExFM, it enables the matching of one dataset to another (moons $\rightarrow$ 8gaussians) as it does not necessitate the presence of an explicit formula for ρ_0. Among other things, this experiment demonstrates the feasibility of our methods when both distributions ρ_0 and ρ_1 are unknown.

Table 4: ExFM-S evaluation on four toy datasets ($\mu \pm \sigma$ over three seeds). For comparison we take I-CFM, OT-CFM, and ExFM (no values for moons $\rightarrow$ 8gaussians due to the absence of explicit formula for ρ_0). Performance in generative modeling ($\mathcal{W}_2$) and dynamic OT optimality (NPE) is assessed. The best result for each metric is highlighted in bold. Instances where we outperform CFM are underscored.

Metric $\rightarrow$	$\mathcal{W}_2 \downarrow$				NPE $\downarrow$			
Algorithm $\downarrow$ Dataset $\rightarrow$	$\mathcal{N} \rightarrow$ moons	$\mathcal{N} \rightarrow$ 8gaussians	moons $\rightarrow$ 8gaussians	$\mathcal{N} \rightarrow$ 2spirals	$\mathcal{N} \rightarrow$ moons	$\mathcal{N} \rightarrow$ 8gaussians	moons $\rightarrow$ 8gaussians	$\mathcal{N} \rightarrow$ 2spirals
I-CFM	0.522 ± 0.015	0.647 ± 0.078	0.966 ± 0.21	1.662 ± 0.067	0.328 ± 0.051	0.209 ± 0.009	0.945 ± 0.025	0.098 ± 0.04
OT-CFM	0.427 ± 0.038	0.528 ± 0.053	$\mathbf{0.569 \pm 0.018}$	1.322 ± 0.052	$\mathbf{0.065 \pm 0.068}$	$\mathbf{0.031 \pm 0.018}$	$\mathbf{0.074 \pm 0.026}$	$\mathbf{0.031 \pm 0.02}$
ExFM	$\mathbf{0.318 \pm 0.010}$	$\mathbf{0.445 \pm 0.075}$	–	$\mathbf{1.276 \pm 0.043}$	0.382 ± 0.050	0.213 ± 0.023	–	$\underline{0.069 \pm 0.064}$
ExFM-S	$\underline{0.486 \pm 0.09}$	$\underline{0.570 \pm 0.053}$	$\underline{0.728 \pm 0.063}$	$\underline{1.361 \pm 0.181}$	0.35 ± 0.143	$\underline{0.166 \pm 0.039}$	0.946 ± 0.059	$\underline{0.083 \pm 0.059}$

4 Conclusions and discussion

The presented method introduces a new loss function in tractable form (in terms of integrals) that improves upon the existing Conditional Flow Matching approach. By "tractable", we refer to a loss function formulation that directly enables the discrete loss to be expressed as in Eq. (13) or approximated using the methods outlined in Appendix B. Moreover, our loss formulation facilitates the explicit derivation of the vector field in some particular cases. For example, (Eq. 37) for Gaussian initial distribution and Gaussian Mixture target distribution. Given the ongoing significance of Gaussian separation within the domains of Flow Matching and Diffusion Models, such an explicit velocity expression represents a novel contribution to the field. Our loss is based on Theorems 1–2 from Lipman et al. (2023) on the equivalence of CFM and FM gradients and theorems from Tong et al. (2024a), that extends the first ones; we then carry out a rigorous derivation that involves first considering an invertible map then moving to a non-invertible map by taking a limit $\sigma_s \rightarrow 0$ (Appendix A). To the best of our knowledge, analogous formulas found in the literature are given without rigorous derivation and thus no conditions are given for them to be true.

New loss as a function of the model parameters, reaches zero at its minimum. Thanks to this, we can: a) write an explicit expression for the vector field on which the loss minimum is achieved; b) get a smaller variance when training on the discrete version of the loss, therefore, we can learn the model faster and more accurately. Since one can consider different modifications of the original CFM (such as those in Table 1), we obtain not one but a whole class of formulas for the vector field (as well as score in stochastic cases). This class can be extended, and our goal was not to cover all possible modifications or special cases, we focused on the method of deriving this formula. Many important special cases are placed in the Appendix.

Numerical experiments conducted on toy 2D data show reliable outcomes under uniform conditions and parameters. Comparison of the absolute values of loss for the proposed method and for CFM for the same distributions show that the absolute values of loss for these models differ strikingly, by a factor of 10^2–10^3. Experiments on high-dimensional datasets also confirm the theoretical deductions about the variance reduction of our method. The main difference of our algorithm is the use of two batch sizes, one for training the model, which is considered small, and the other for estimating the integral, which is larger than the first one by 10^1–10^2. In addition, the integral can be estimated with samples using different methods, not only self-normalize important sampling (SIS), but also rejection sampling, or SIS with reduced bias (see Appendix B). Other methods of integral estimation are also possible. However, we emphasize that we do not expect to use the proposed method in its pure form. On the contrary, we expect that the theoretical implications of our formulas will contribute to the construction of better learning or inference algorithms in conjunction with other heuristics or methods.

Algebraic analysis of variance for some cases (in particular, for the case $t = 0$ or for the case of two Gaussians as initial and final distributions) show an improvement in variance when using the new loss. However, it is rather difficult to analyze in the general case, for all times t and general distributions ρ_0 and ρ_1.

Having the expression for the vector field and score in the form of integrals, we can explicitly write out their expressions for some simple cases; in the case of Gaussian distributions we can also write out the exact solution for the trajectories. Thus, our approach allows one to advance the theoretical study of FM-based and Diffusion Model-based frameworks.

REFERENCES

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

Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint 2303.08797*, 2023.

Gabriel Cardoso, Sergey Samsonov, Achille Thin, Eric Moulines, and Jimmy Olsson. Br-snis: Bias reduced self-normalized importance sampling. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 716–729. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/04bd683d5428d91c5fbb5a7d2c27064d-Paper-Conference.pdf.

Ricky T. Q. Chen and Yaron Lipman. Riemannian flow matching on general geometries. *arXiv:2302.03660*, 2023.

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.

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

Alexia Jolicoeur-Martineau, Kilian Fatras, and Tal Kachman. Generating and imputing tabular data via diffusion and flow-based gradient-boosted trees. *arXiv:2309.09968*, 2023.

Puheng Li, Zhong Li, Huishuai Zhang, and Jiang Bian. On the generalization properties of diffusion models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 2097–2127. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/06abed94583030dd50abe6767bd643b1-Paper-Conference.pdf.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh 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 *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pp. 416–423 vol.2, 2001. doi: 10.1109/ICCV.2001.937655.

Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in gan evaluation. In *CVPR*, 2022.

Aram-Alexandre Pooladian, Heli Ben-Hamu, Carles Domingo-Enrich, Brandon Amos, Yaron Lipman, and Ricky T. Q. Chen. Multisample flow matching: Straightening flows with minibatch couplings. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 28100–28127. PMLR, 7 2023. URL https://proceedings.mlr.press/v202/pooladian23a.html.

Aaditya Ramdas, Nicolás García Trillos, and Marco Cuturi. On wasserstein two-sample testing and related families of nonparametric tests. *Entropy*, 19(2):47, 2017.

Kulin Shah, Sitan Chen, and Adam Klivans. Learning mixtures of gaussians using the DDPM objective. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=aig7sgdRfI.

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, Lille, France, 7 2015. PMLR. URL https://proceedings.mlr.press/v37/sohl-dickstein15.html.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Neural Information Processing Systems (NeurIPS)*, 2019.

Gábor J Székely. E-statistics: The energy of statistical samples. *Bowling Green State University, Department of Mathematics and Statistics Technical Report*, 3(05):1–18, 2003.

Alexander Tong, Kilian FATRAS, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024a. ISSN 2835-8856. URL https://openreview.net/forum?id=CD9Snc73AW. Expert Certification.

Alexander Tong, Nikolay Malkin, Kilian Fatras, Lazar Atanackovic, Yanlei Zhang, Guillaume Huguet, Guy Wolf, and Yoshua Bengio. Simulation-free schrödinger bridges via score and flow matching. In *The 27th International Conference on Artificial Intelligence and Statistics*, 2024b. URL https://virtual.aistats.org/virtual/2024/poster/6691.

A PROOF OF THE THEOREMS

A.1 PROOF OF THE THEOREM 2.1

Proof. We need to proof, that $\frac{\mathrm{d}L_{\text{CFM}}(\theta)}{\mathrm{d}\theta} = \frac{\mathrm{d}L_{\text{ExFM}}(\theta)}{\mathrm{d}\theta}$.

To establish the equivalence of L_{CFM} and L_{ExFM} up to a constant term, we begin by expressing L_{CFM} in the format specified by equation (6):

$$L_{\text{CFM}} = \mathbb{E}_{t,x_1,x\sim\hat{\rho}_t(\cdot)}\|v_\theta(x,\,t) - w(t,x_1,x)\|^2 \times \xi_t(x;x_1)/\rho_1(x_1).$$

Utilizing the bilinearity of the 2-norm, we can rewrite L_{CFM} as:

$$L_{\text{CFM}} = \mathbb{E}_{t,x_1,x\sim\hat{\rho}_t(\cdot)}\frac{\|v_\theta(x,\,t)\|^2\xi_t(x;x_1)}{\rho_1(x_1)} - 2\mathbb{E}_{t,x_1,x\sim\hat{\rho}_t(\cdot)}\frac{v_\theta(x,\,t)^T\cdot w(t,x_1,x)\xi_t(x;x_1)}{\rho_1(x_1)} + C. \quad (21)$$

Here, T denotes transposed vector, dot denotes scalar product, C represents a constant independent of θ.

Noting that $\mathbb{E}_{x_1}\xi_t(x;x_1)/\rho_1(x_1) = 1$:

$$\mathbb{E}_{x_1}\frac{\xi_t(x;x_1)}{\rho_1(x_1)} = \int\frac{\eta_t(x;x_1)\rho_1(x_1)\,\mathrm{d}x_1}{\int\eta_t(x;x_1)\rho_1(x_1)\,\mathrm{d}x_1} = 1,$$

we can simplify the first term in the expansion (21):

$$\mathbb{E}_{t,x_1,x\sim\hat{\rho}_t(\cdot)}\frac{\|v_\theta(x,\,t)\|^2\xi_t(x;x_1)}{\rho_1(x_1)} = E_{t,x\sim\hat{\rho}_t(\cdot)}\|v_\theta(x,\,t)\|^2\,\mathbb{E}_{x_1}\frac{\xi_t(x;x_1)}{\rho_1(x_1)} = E_{t,x\sim\hat{\rho}_t(\cdot)}\|v_\theta(x,\,t)\|^2. \quad (22)$$

For our loss L_{ExFM} in the form (8) we also use the bilinearity of the norm:

$$L_{\text{ExFM}} = \mathbb{E}_{t,x\sim\hat{\rho}_t(\cdot)}\|v_\theta(x,\,t)\|^2 - 2\mathbb{E}_{t,x\sim\hat{\rho}_t(\cdot)}\mathbb{E}_{x_1}\frac{v_\theta(x,\,t)^T\cdot w(t,x_1,x)\xi_t(x;x_1)}{\rho_1(x_1)} + C. \quad (23)$$

Comparing the last expression and the Eq. (21) with the modification (22) and also taking into account the independence of random variables x and x_1, we come to the conclusion that L_{ExFM} is equal to L_{CFM} up to some constant independent of the model parameters.

□

A.2 SKETCH OF THE PROOF OF THE THEOREM 2.4

Proof. We need to prove that $\mathbb{D}\frac{\mathrm{d}L^d_{\text{ExFM}}(\theta)}{\mathrm{d}\theta} \leq \mathbb{D}\frac{\mathrm{d}L^d_{\text{CFM}}(\theta)}{\mathrm{d}\theta}$, where $L^d_{\text{ExFM}}(\theta)$ and $L^d_{\text{CFM}}(\theta)$ discrete loss functions presented in (14) and (13). Firstly, let us rewrite the derivative of loss functions using the bilinearity:

$$\frac{\mathrm{d}L^d_{\text{ExFM}}(\theta)}{\mathrm{d}\theta} = 2\sum_{i,j}\left(\frac{\mathrm{d}v_\theta(x^{j,i},\,t^j)}{\mathrm{d}\theta}\right)^T\cdot\big(v_\theta(x^{j,i},\,t^j) - v^d(x^{j,i},t^j)\big).$$

Note that in this expression, values $x^{j,i}$ as well as t^j, which are included in the argument of the function v, are fixed (our goal to calculate the variance with fixed model arguments). Thus, we need to consider the variance of the remaining expression arising from the randomness of $\overline{x}_1^k$.

Recall (below we will omit the indices at variables x and t),

$$v^d(x,\,t) = \frac{\sum_{k=1}^N w(t,\overline{x}_1^k,x)\cdot\rho_0\big(\phi^{-1}_{t,\overline{x}_1^k}(x)\big)}{\sum_{k=1}^N\rho_0\big(\phi^{-1}_{t,\overline{x}_1^k}(x)\big)}.$$

Note, that if $N = 1$, (*i. e.* we do not sample any additional points other than the ones we have already sampled) this expression is exactly the same as the derivative of the common discretized CFM loss $\frac{\mathrm{d}L^d_{\text{CFM}}(\theta)}{\mathrm{d}\theta}$.

Moreover, recall that one of the points (without loss of generality, we can assume that its index is 1) $\overline{x}_1^1$ is added from the set from which point x was derived: $x = \phi_{t,\overline{x}_1^1}(x_0)$. (Here x_0 is the paired point to $\overline{x}_1^1$)

Thus, we can rewrite expression for v^d:

$$v^d(x,\, t) = \frac{w(t, \overline{x}_1^1, x)\rho_0(x_0) + \sum_{k=2}^N w(t, \overline{x}_1^k, x) \cdot \rho_0\left(\phi^{-1}_{t,\overline{x}_1^k}(x)\right)}{\rho_0(x_0) + \sum_{k=2}^N \rho_0\left(\phi^{-1}_{t,\overline{x}_1^k}(x)\right)}. \tag{24}$$

Thus, our task was reduced to evaluating how well the additional terms (for k starting from 2) improve approximate of the original integrals that are in loss (8).

So, we need to estimate the following variance ratio, where in the numerator is the variance of discrete loss CFM, and in the denominator — the variance of loss ExFM:

$$k_D = \frac{\mathbb{D}\big(v_\theta(x,t) - w(t,\overline{x}_1^1, x)\big)}{\mathbb{D}\left(v_\theta(x,t) - \frac{\sum_{k=1}^N w(t,\overline{x}_1^k,x)\cdot\rho_0\left(\phi^{-1}_{t,\overline{x}_1^k}(x)\right)}{\sum_{k=1}^N \rho_0\left(\phi^{-1}_{t,\overline{x}_1^k}(x)\right)}\right)}$$

The smaller coefficient k_D is, the better the proposed loss ExFM works.

Formally, we can write our problem as an importance sampling problem for the following integral:

$$I = \int f(x)p(x)\,\mathrm{d}x\,.$$

This integral we estimate by sample mean of the following expectation over some random variable with density function $q(x)$:

$$I = \mathbb{E}_{x\sim q}\big(w(x)f(x)\big)$$

with

$$w(x) = \frac{p(x)}{q(x)}.$$

We replace the exact value of I with the value

$$\overline{I} = \frac{\sum_{k=1}^N w(\overline{x}_1^i) f(\overline{x}_1^k)}{\sum_{i=k}^N w(\overline{x}_1^k)}.$$

It follows from the strong law of large numbers that in the limit $N \to \infty$, $I \to \overline{I}$ almost surely. From the central limit theorem we can find the asymptotic variance:

$$\mathbb{D}\overline{I} = \frac{1}{N}\mathbb{E}_{x\sim q}\big(w^2(x)(f(x) - I)^2\big). \tag{25}$$

In our case (loss L_{ExFM}), we have $q(x_1) = \rho_1(x_1)$, $f(x_1) = w(t, x_1, x)$ and $w(x_1) = \rho_0\left(\phi^{-1}_{t,x_1}(x)\right)$.

Despite the fact that the equation (25) for the variance contains N in the denominator, it is rather difficult to give an estimate of its behavior in general. The point is that this formula is well suited for the case when w in it is of approximately the same order. In the considered case, this is achieved at times t noticeably less than 1.

But in the case, when t is closed to 1 we have, for example, for the linear map, that

$$w(x_1) = \rho_0\left(\phi^{-1}_{t,x_1}(x)\right) = \rho_0\left(\frac{x - x_1 t}{1 - t}\right)$$

and this function has a sharp peak near the point x/t if it is considered as a function of x_1. Thus, at such values of t, only a small number of summands will give a sufficient contribution to the sum compared to the first term.

Finally, inequality $k_D < 1$ is formally fulfilled, but how much k_D is less than one depends on many factors.

□

A.3 EXPRESSIONS FOR THE REGULARIZED MAP

To justify the expression (11), we use a invertable transformation and then strictly take the limit $\sigma_s \to 0$.

Expression Eq. (11), (16) are obtained for the simple map $\phi_{t,x_1}(x_0) = (1-t)x_0 + tx_1$ which is not invertable at $t = 1$. For the map with small regularizing parameter $\sigma_s > 0$ $\phi_{t,x_1}(x_0) = (1-t)x_0 + tx_1 + \sigma_s x_0$, which is invertable at all time values $0 \le t \le 1$, Eq. (11), (16) needs modifications. Namely, for this map the following exact formulas holds true

$$v(x,t) = \int w(t,x_1,x)\xi_t(x;x_1)\rho_1(x_1)\,\mathrm{d}x_1 = \frac{\int \big(x_1 - x(1-\sigma_s)\big)\rho_0\left(\frac{x-x_1 t}{1+\sigma_s t - t}\right)\rho_1(x_1)\,\mathrm{d}x_1}{(1+\sigma_s t - t)\int \rho_0\left(\frac{x-x_1 t}{1+\sigma_s t - t}\right)\rho_1(x_1)\,\mathrm{d}x_1}. \quad (26)$$

By direct substitution we make sure that for this vector field

$$v(x,\,0) = \int x_1\rho_1(x_1)\,\mathrm{d}x_1 - x(1-\sigma_s) \quad (27)$$

and

$$v(x,\,1) = \frac{\int (x-y)\rho_0(y)\rho_1(x-y\sigma_s)\,\mathrm{d}y}{\int \rho_0(y)\rho_1(x-y\sigma_s)\,\mathrm{d}y}, \quad (28)$$

where we perform change of the variables $y \leftarrow \frac{x_1 - x}{\sigma_s t}$.

A.3.1 PROF OF THE EXPLICIT FORMULA (11) FOR THE VECTOR FIELD

Assumption A.1. Density ρ_1 is continuous at any point $x \in (-\infty,\,\infty)$.

Theorem A.2. *In equations* (26)*,* (27) *and* (28) *we can take the limit $\sigma_s \to 0$ under integrals to get Eq.* (11) *and* (12)*.*

Proof. Assuming that the distribution ρ_1 has a finite first moment: $|\int \xi\rho_1(\xi)\,\mathrm{d}\xi\,| < C_1$ and that the density of ρ_0 is bounded: $\rho_0(x) < C_2$, $\forall x \in (-\infty,\infty)$, we obtain that the integrand functions in the numerator and denominator in the Eq. (26) can be bounded by the following integrable functions independent of σ_s and t:

$$\rho_0\left(\frac{x-x_1 t}{1+\sigma_s - t}\right)\rho_1(x_1) < C_1\rho_1(x_1)$$

and

$$0 \le x_1\rho_0\left(\frac{x-x_1 t}{1+\sigma_s t - t}\right)\rho_1(x_1) < x_1 C_1\rho_1(x_1), \quad x \ge 0,$$
$$0 > x_1\rho_0\left(\frac{x-x_1 t}{1+\sigma_s t - t}\right)\rho_1(x_1) > x_1 C_1\rho_1(x_1), \quad x < 0.$$

It follows that both integrals in expression (26) converge absolutely and uniformly. So, we can swap the operations of taking the limit and integration, and we can take the limit $\sigma_s \to 0$ in the integrand for any time $t \in [0,\,t_0]$ for arbitrary $t_0 < 1$.

Now, let us consider the case $t = 1$. From Assumption A.1 the boundedness of the density ρ_1 follows: $\rho_1(x) < C_2, \forall x \in (-\infty,\,\infty)$. Thus, integrand functions in the numerator and denominator in the Eq. (28) can be bounded by the following integrable functions independent of σ_s:

$$\rho_0(y)\rho_1(x-y\sigma_s) < \rho_0(y)C_2$$

and

$$0 \le y\rho_0(y)\rho_1(x-y\sigma_s) < yC_2\rho_0(y), \quad y \ge 0,$$
$$0 > y\rho_0(y)\rho_1(x-y\sigma_s) > yC_2\rho_0(y), \quad y < 0.$$

The existence of the limit

$$\lim_{\sigma_s \to 0} \rho_1(x-y\sigma_s) = \rho_1(x),$$

follows from Assumption A.1.

Finally, we conclude that formula (11), regarded as the limit $\sigma_s \to 0$ of the (26) at any $t \in [0,\,1]$, is true. □

Theorem A.3. *The vector field in Eq.* (11) *delivers minimum to the Flow Matching objective (see the work Lipman et al. (2023)),*

$$\mathbb{E}_t \mathbb{E}_{x\sim\rho(x,t)} \|\overline{v}(x,t) - v(x,t)\|,$$

where $\rho(x,t)$ *and* $\overline{v}(x,t)$ *satisfy the equation* (1) *with the given densities* ρ_0 *and* ρ_1.

Proof. The proof is based on the previous statements and on a Theorem 1 from Lipman et al. (2023) (that the marginal vector field based on conditional vector fields generates the marginal probability path based on conditional probability paths.

To complete the proof, we must justify that, with σ_s tending to zero, the marginal path at $t = 1$ coincides with a given probability ρ_1.

Consider the marginal probability path $p_t(x,t)$

$$p_t(x,t) = \int p_t(x|x_1,\sigma_s)\rho_1(x_1)\mathrm{d}x_1 \tag{29}$$

where $p_t(x|x_1,\sigma_s)$ is conditional probability paths obtained by regularized linear conditional map. Distribution p_t in the time $t = 0$ is equal to standard normal distribution $p_0(x|x_1,\sigma_s) = \mathcal{N}(x \mid 0, 1)$ and at the time $t = 1$ it is a stretched Gaussian centered at x_1: $p_1(x|x_1,\sigma_s) = \mathcal{N}(x \mid x_1, \sigma_s I)$.

Substituting p_1 into the Eq. (29) and considering that there exists a limit $\sigma_s \to 0$ due to Assumption A.1, we obtain

$$p_1(x) = \lim_{\sigma_s\to 0} \int p_t(x|x_1,\sigma_s)\rho_1(x_1)\mathrm{d}x_1 = \rho_1(x_1).$$

This finish the proof. □

A.3.2 Learning procedure for $\sigma_s > 0$

Using standard normal distribution as initial density ρ_0, and the regularized map $\phi_{t,x_1}(x_0) = (1 - t)x_0 + tx_1 + \sigma_s t x_0$ we obtain the following approximation formula

$$v^d(x,\,t) = \frac{\sum_{k=1}^N \frac{\overline{x}_1^k - x(1-\sigma_s)}{1-t(1-\sigma_s)} \exp\big(Y^k\big)}{\sum_{k=1}^N \exp(Y^k)}, \quad \text{where} \quad Y^k = -\frac{1}{2}\frac{\big\|x - t\cdot\overline{x}_1^k\big\|^2_{\mathbb{R}^d}}{1-t(1-\sigma_s)}.$$

In practical applications, the exponent calculation is replaced by the $\mathrm{SoftMax}$ function calculation, which is more stable.

B Estimation of integrals

In general, we need to estimate the following expression

$$I(\eta) = \frac{\int w(x_1,\,\eta) f(x_1,\,\eta)\rho_1(x_1)\,\mathrm{d}x_1}{\int f(x_1,\,\eta)\rho_1(x_1)\,\mathrm{d}x_1}.$$

In particular, substituting $\eta \to \{x, t\}$, $w(x,\eta) \to (x_1 - x)/(1 - t)$ we obtain formula (11) and similar ones with similar substitutions.

If we can sample from the ρ_1 distribution, we can estimate this integral in two ways: *self-normalized importance sampling* and *rejection sampling*.

Let $\mathcal{X} = \{x_1^k\}_{k=1}^N$ be N samples from the distribution ρ_1.

Self-normalized Importance Sampling In this case

$$I(\eta) \approx \frac{\sum\limits_{k=1}^N w(x_1^k,\eta) f(x_1^k,\,\eta)\rho_1(x_1^k)\,\mathrm{d}x_1}{\sum\limits_{k=1}^N f(x_1^k,\,\eta)\rho_1(x_1^k)\,\mathrm{d}x_1}. \tag{30}$$

This estimate is biased in theory, but there several methods to reduce this bias and improve this estimate, see, for example, Cardoso et al. (2022). Our numerical experiments generally show that the estimation (30) in the form is already sufficient for stable results; we don not observe any bias.

Rejection sampling Let $\mathcal{Y} = \{y^k\}_{k=1}^M \subset \mathcal{X}$ be a subset of the the initially given set of samples, which is formed according to the following rule. Let $C = \sup_x \rho_1(x)$. For a given sample x_1^j we generate a random uniformly distributed variable $\xi_j \sim \mathcal{U}(0,1)$ and if

$$f(x_1^j) \geq C\xi_j,$$

then we put the point x_k^j to the set $\mathcal{Y}$; otherwise we reject it.

Having formed the set $\mathcal{Y}$, we evaluate the integral as

$$I(\eta) \approx \frac{1}{M}\sum_{k=1}^{M} w(y^k, \eta).$$

To justify the last estimation, we note, that the points from the set $\mathcal{Y}$ are distributed according to (non-normalized) density $\rho(x)f(x,\eta)\rho_1(x)$. One can show it using the proof of the rejection sampling method. This is the same density as in Eq. (7) and thus we estimate the expression (10) using Important Sampling without any additional denominator.

Comparison When we apply these techniques to evaluating the expression for the vector field, we know that when the time parameter t is close to 1, the function $f(x_1, \eta)$ (which is a scaled ρ_0) has a peak at the point $x = x_1$. This means that only a small number of points from the original set will end up in the set $\mathcal{Y}$. Moreover, in the case when the time t is very close to one and the data are well separated, only one point x_1 will end up in $\mathcal{Y}$. This explains why we initially put this point in the set $\mathcal{X}$, because otherwise it would be possible that the set $\mathcal{Y}$ is empty and $M = 0$.

As a future work, we indicate a theoretical finding of the probability of hitting a particular point x_1 in the set $\mathcal{Y}$ and, thus, a modification of our algorithm, when the sample x_1 will not always go to the set $\mathcal{X}$, but with some probability — the greater the t the closer this probability to 1.

C THE MAIN ALGORITHM AND EXTENSIONS AND GENERALIZATION OF THE EXACT EXPRESSION

Algorithm 1 Vector field model training algorithm

Require: Sampler from distribution ρ_1 (or a set of samples); parameters n and m (number of spatial and time points, correspondingly); parameter N (number of averaging point); model $v_\theta(x,t)$; algorithm with parameters for SGD

Ensure: quasi-optimal parameters θ for the trained model

1: Initialize θ (maybe random)
2: **while** exit condition is not met **do**
3: Sample m points $\{t^j\}$ from $\mathcal{U}[0,1]$
4: Sample n points pairs $\{x_0^i, x_1^i\}_{i=1}^n$ from joint distribution π ($\pi(x_0, x_1) = \rho_0(x_0)\rho_1(x_1)$ if variables are independent)
5: Sample $N-n$ points $\{\hat{x}_1^l\}$ from ρ_1 and form $\{\overline{x}_1^k\} = \{x_1^i\} \cup \{\hat{x}_1^l\}$ *// We can take all available samples as $\{\overline{x}_1^k\}$ if we don't have access to a sampler, but only ready-made samples.*
6: For all i and j calculate the sum at the right side of (14) (using (16) if ρ_0 is standard Gaussian or (24) in general)
7: Calculate the sum on i and j in discrete loss (14), and take backward derivative, obtaining approximate grad $G \approx \nabla_\theta L_{\text{ExFM}}$ of loss L_{ExFM} on model parameters θ.
8: Update model parameters $\theta \leftarrow SGD(\theta, G)$
9: **end while**

General form of the proposed Algorithm is given in Alg 1.

When using other maps, formula (11) is modified accordingly. For example, if we use the regularized map $\phi_{t,x_1}(x_0) = (1-t)x_0 + tx_1 + \sigma_s t x_0$, we get the formula (26).Note, that in this case the final density $\rho(x,\, 1)$, obtained from the continuity equation is not equal to ρ_1, but is its smoothed modification.

When using a different initial density ρ_0 (not the normal distribution), an obvious modification will be made to formula (16).

Diffusion-like models We can treat so-called Variance Preserving Ho et al. (2020) model as CFM with the map

$$\phi_{t,x_1}(x) = \alpha_{1-t}x + \sqrt{1-\alpha_{1-t}^2}x_1.$$

and ρ_0 as standard normal distribution: $\rho_0 = \mathcal{N}\left(\cdot\middle|\, 0, 1^2\right)$ In this case, the common expression (10) for vector filed transforms to

$$v(x,t) = \frac{\int (x\alpha_{1-t} - x_1)\alpha'_{1-t}\,\rho_0\left(\frac{x-x_1\alpha_{1-t}}{\sqrt{1-\alpha_{1-t}^2}}\right)\rho_1(x_1)\,\mathrm{d}x_1}{(1-\alpha_{1-t}^2)\int \rho_0\left(\frac{x-x_1\alpha_{1-t}}{\sqrt{1-\alpha_{1-t}^2}}\right)\rho_1(x_1)\,\mathrm{d}x_1}, \tag{31}$$

where $\alpha'_s = \frac{\mathrm{d}\alpha_s}{\mathrm{d}s}$.

Similarity we can treat so-called Variance Exploding Song & Ermon (2019) model as CFM with the map

$$\phi_{t,x_1}(x) = \sigma_{1-t}x + x_1.$$

and ρ_0 also as standard normal distribution: $\rho_0 = \mathcal{N}\left(\cdot\middle|\, 0, 1^2\right)$ In this case, the common expression (10) for vector filed transforms to

$$v(x,t) = \frac{\int (x_1 - x)\sigma'_{1-t}\,\rho_0\left(\frac{x-x_1}{\sigma_{1-t}}\right)\rho_1(x_1)\,\mathrm{d}x_1}{\sigma_{1-t}\int \rho_0\left(\frac{x-x_1}{\sigma_{1-t}}\right)\rho_1(x_1)\,\mathrm{d}x_1}, \tag{32}$$

where $\sigma'_s = \frac{\mathrm{d}\sigma_s}{\mathrm{d}s}$.

Joint Distribution Moreover, in addition to the independent densities $x_0 \sim \rho_0$ and $x_1 \sim \rho_1$, we can use the joint density $\{x_0,\, x_1\} \sim \pi(x_0,\, x_1)$. In the papers Tong et al. (2024b;a), optimal transport (OT) and Schrödinger's bridge are taken as π. In this case the expression for the vector field changes insignificantly: the conditional probability ρ_c from Eq. (7) is subject to change:

$$\rho_c(x|x_1,t) = \frac{\pi\left(\phi_{t,x_1}^{-1}(x), x_1\right)\det\left[\frac{\partial\phi_{t,x_1}^{-1}(x)}{\partial x}\right]}{\int \pi\left(\phi_{t,x_1}^{-1}(x), x_1\right)\det\left[\frac{\partial\phi_{t,x_1}^{-1}(x)}{\partial x}\right]\mathrm{d}x_1}. \tag{33}$$

Then, Eq. (10) remains the same in general case. In the case of linear ϕ, the extension of Eq. (11) reads

$$v(x,t) = \frac{\int (x_1 - x)\,\pi\big(\phi_{t,x_1}^{-1}(x), x_1\big)\det\left[\frac{\partial\phi_{t,x_1}^{-1}(x)}{\partial x}\right]\mathrm{d}x_1}{(1-t)\int \pi\big(\phi_{t,x_1}^{-1}(x), x_1\big)\det\left[\frac{\partial\phi_{t,x_1}^{-1}(x)}{\partial x}\right]\mathrm{d}x_1}. \tag{34}$$

In all of the above cases, the essence of Algorithm 1 does not change (except that in the case of dependent x_0 and x_1 we should be able either to calculate the value of $\pi\big(\phi_{t,x_1}^{-1}(x), x_1\big)/\rho_1(x_1)$ or to estimate it).

D Several analytical results, following from the explicit formula

In this section, we present several analytical results that directly follow from our exact formulas for the vector field, which, to the best of our knowledge, have not been published before.

D.1 Exact path from one Gaussian to another Gaussian

Consider the flow from a one-dimensional Gaussian distribution $\rho_0 \sim \mathcal{N}\left(\cdot\middle|\, \mu_0, \sigma_0^2\right)$ into another (with other parameters) Gaussian distribution $\rho_1 \sim \mathcal{N}\left(\cdot\middle|\, \mu_1, \sigma_1^2\right)$. Note that in this case the generalization to the multivariate case is done directly, so the spatial variables are separated.

From the general formula (11) we have:

$$v(x,t) = \frac{\int (x_1 - x)\mathcal{N}\left(\frac{x-tx_1}{1-t}\middle|\, \mu_0, \sigma_0^2\right)\mathcal{N}\left(x_1\middle|\, \mu_1, \sigma_1^2\right)\mathrm{d}x_1}{(1-t)\int \mathcal{N}\left(\frac{x-tx_1}{1-t}\middle|\, \mu_0, \sigma_0^2\right)\mathcal{N}\left(x_1\middle|\, \mu_1, \sigma_1^2\right)\mathrm{d}x_1} = \\ = \frac{\int (x_1 - x)\exp\left(-\left(\frac{x-tx_1}{1-t} - \mu_0\right)^2/(2\sigma_0^2) - (x_1-\mu_1)^2/(2\sigma_1^2)\right)\mathrm{d}x_1}{(1-t)\int \exp\left(-\left(\frac{x-tx_1}{1-t} - \mu_0\right)^2/(2\sigma_0^2) - (x_1-\mu_1)^2/(2\sigma_1^2)\right)\mathrm{d}x_1}.$$

Both integrals in the last expression are taken explicitly:

$$\int \mathcal{N}\left(\frac{x-tx_1}{1-t}\middle|\, \mu_0, \sigma_0^2\right)\mathcal{N}\left(x_1\middle|\, \mu_1, \sigma_1^2\right)\mathrm{d}x_1 = \\ = \frac{\exp\left(-\frac{(x-\mu_0(1-t)-\mu_1 t)^2}{2\left(\sigma_1^2 t^2 + \sigma_0^2(1-t)^2\right)}\right)}{\sqrt{2\pi}\sqrt{\sigma_0^2 + \frac{\sigma_1^2 t^2}{(t-1)^2}}} = \mathcal{N}\left(\frac{x}{1-t}\middle|\, \frac{\mu_0(1-t)+\mu_1 t}{1-t}, \sigma_0^2 + \frac{\sigma_1^2 t^2}{(t-1)^2}\right).$$

Note that the last relation can be obtained as a distribution of two Gaussian random variables with corresponding parameters.

The second integral:

$$\int \frac{x_1 - x}{1-t}\mathcal{N}\left(\frac{x-tx_1}{1-t}\middle|\, \mu_0, \sigma_0^2\right)\mathcal{N}\left(x_1\middle|\, \mu_1, \sigma_1^2\right)\mathrm{d}x_1 = \\ = \frac{\exp\left(-\frac{(x-\mu_0(1-t)-\mu_1 t)^2}{2\left(\sigma_1^2 t^2 + \sigma_0^2(1-t)^2\right)}\right)}{\sqrt{2\pi}} \frac{(1-t)\left(\sigma_1^2 t(x-\mu_0) + \sigma_0^2(t-1)(x-\mu_1)\right)}{\left(\sigma_1^2 t^2 + \sigma_0^2(1-t)^2\right)^{3/2}}.$$

Thus, in the considered case we can explicitly write the expression for the vector field v:

$$v(x,t) = \frac{\sigma_1^2 t(x-\mu_0) - \sigma_0^2(1-t)(x-\mu_1)}{\sigma_1^2 t^2 + \sigma_0^2(1-t)^2}. \tag{35}$$

For this vector field we can explicitly solve the equation for the path $x(t)$ starting from the arbitrary point x_0

$$\begin{cases} \dfrac{\partial x(t)}{\partial t} = v(x(t),\, t), \\ x(0) = x_0 \end{cases}.$$

The solution is:

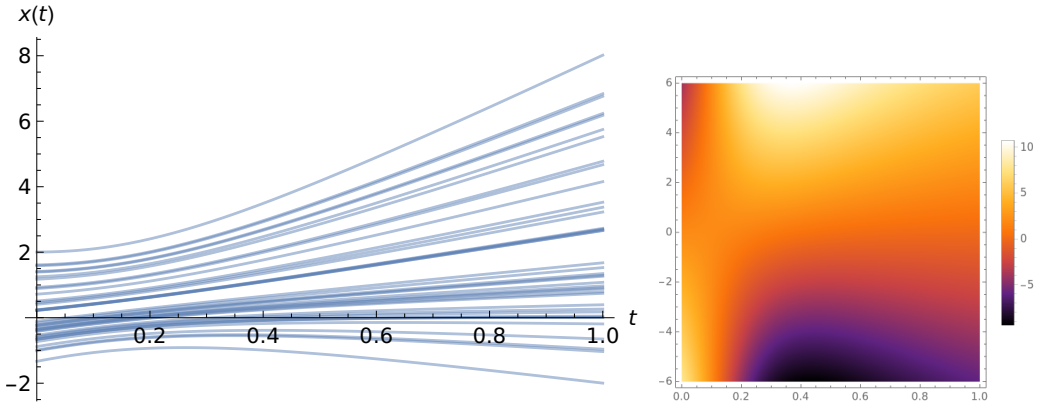
$$x(t) = (1-t)\mu_0 + t\mu_1 + (x_0 - \mu_0)\sqrt{(\sigma_1/\sigma_0)^2 t^2 + (1-t)^2}. \tag{36}$$

Note that although this solution does not correspond to the Optimal Transport joint distribution, since the obtained path is not a straight line in general, (*i. e.* we do not have a solution to the Kantorovich's formulation of the OT problem) the endpoint $x(1) = \mu_1 + (x_0 - \mu_0)\dfrac{\sigma_1}{\sigma_0}$ falls exactly in the one that is optimal if we solve the OT problem in the Monge formultation. Thus, the map $x(0) \to x(1)$ is the OT map for the case of 2 Gaussian.

See the Fig. 4 for the examples of the paths for the obtained solution.

D.2 From one Gaussian to Gaussian Mixture

Let initial distribution be standard Gaussian $\rho_0 = \mathcal{N}\left(\cdot\middle|\, 0, 1^2\right)$, and the target distribution be Gaussian Mixture (GM) of two symmetric Gaussians: $\rho_1(x) = 1/2(\mathcal{N}\left(x\middle|\, \mu, \sigma^2\right) + \mathcal{N}\left(x\middle|\, -\mu, \sigma^2\right))$, In



(a) Trajectories (b) Vector field

Figure 4: a) $N = 40$ random trajectories from from $\mathcal{N}\left(\cdot\middle|\, 0, 1^2\right)$ to $\mathcal{N}\left(\cdot\middle|\, 2, 3^2\right)$; (b) 2D plot of the vector field in this case

this case, we can obtain exact form for v

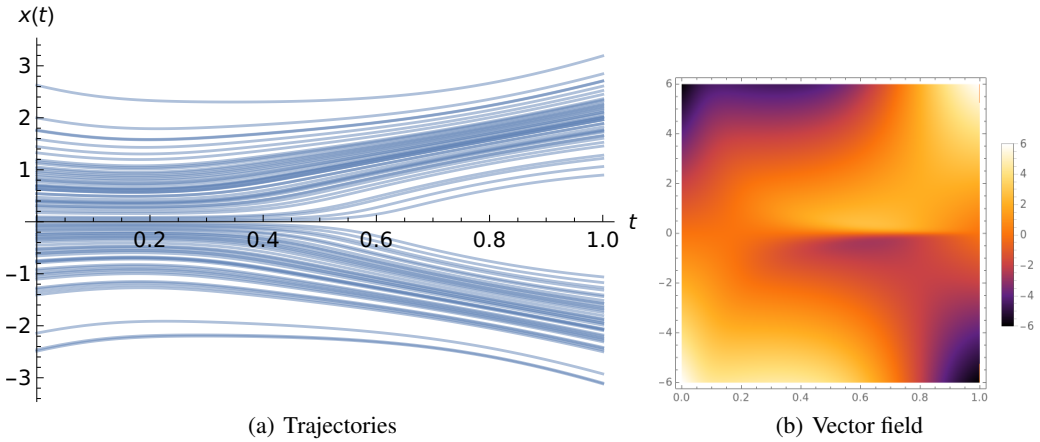
$$v(x,t) = \frac{\exp\left(-\frac{\mu^2}{2\sigma^2} + \frac{\mu^2 t^2 + x^2}{\sigma^2 t^2 + (t-1)^2} - \frac{x^2}{2(t-1)^2}\right)}{\left(\sigma^2 t^2 + (t-1)^2\right)\left(e^{\frac{(x-\mu t)^2}{2\left(\sigma^2 t^2 + (t-1)^2\right)}} + e^{\frac{(\mu t + x)^2}{2\left(\sigma^2 t^2 + (t-1)^2\right)}}\right)} \times$$

$$\left[\mu(t-1)\left(\exp\left(\frac{\left(\mu(t-1)^2 - \sigma^2 t x\right)^2}{2\sigma^2(t-1)^2\left(\sigma^2 t^2 + (t-1)^2\right)}\right) - \exp\left(\frac{\left(\mu(t-1)^2 + \sigma^2 t x\right)^2}{2\sigma^2(t-1)^2\left(\sigma^2 t^2 + (t-1)^2\right)}\right)\right) + \right.$$

$$+ x\left(\sigma^2 t + t - 1\right)\left(\exp\left(\frac{\left(\mu(t-1)^2 - \sigma^2 t x\right)^2}{2\sigma^2(t-1)^2\left(\sigma^2 t^2 + (t-1)^2\right)}\right)\right.$$

$$\left.\left. + \exp\left(\frac{\left(\mu(t-1)^2 + \sigma^2 t x\right)^2}{2\sigma^2(t-1)^2\left(\sigma^2 t^2 + (t-1)^2\right)}\right)\right)\right], \quad (37)$$

but the expression for the path $x(t)$ is unknown.



(a) Trajectories (b) Vector field

Figure 5: a) $N = 80$ random trajectories from $\mathcal{N}\left(\cdot\middle|\, 0, 1^2\right)$ to GM of $\mathcal{N}\left(\cdot\middle|\, -2, 1/2^2\right)$ and $\mathcal{N}\left(\cdot\middle|\, 2, 1/2^2\right)$; (b) 2D plot of the vector field in this case

Numerically solution of the differential equation with the obtained vector field give the trajectories shown in Fig. 5.

D.3 From Gaussian to Gaussian with stochastic

Using Eq. (44)-(46) we can explicitly calculate vector field v and score s with the setup as in Sec. D.1 but with additional noise, *i. e.* in the stochastic case.

D.3.1 Gaussian to Gaussian with noise

Consider like in the Sec. D.1 the flow from a one-dimensional standard Gaussian distribution $\rho_0 \sim \mathcal{N}\left(\cdot \middle| 0, 0^2\right)$ into another (with other parameters) Gaussian distribution $\rho_1 \sim \mathcal{N}\left(\cdot \middle| \mu_1, \sigma_1^2\right)$ but with additional noise as described above.

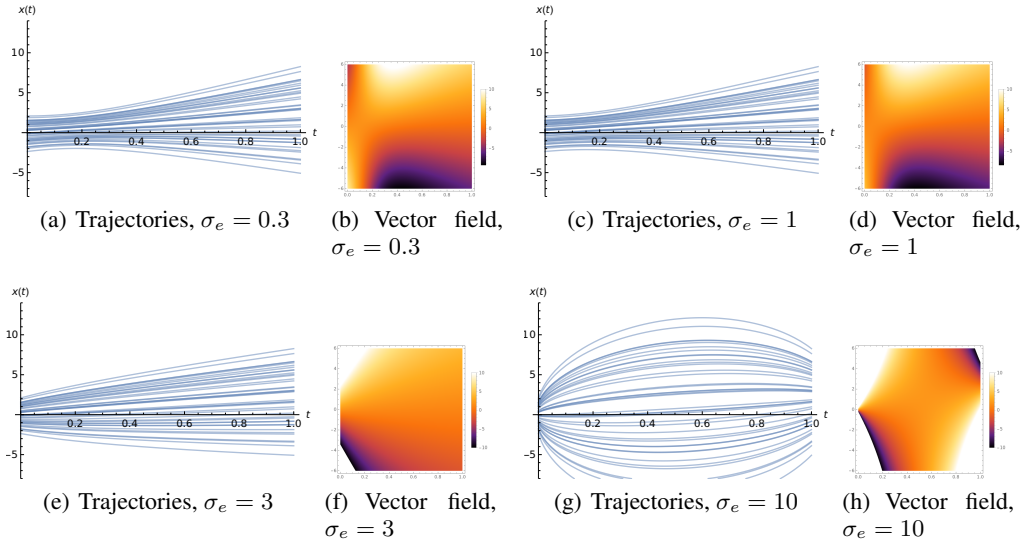
In this case we have for the field.

$$v(x,\, t) = \frac{x\big(t\sigma_1^2 + (1-t)\sigma_e^2/2\big) - (x-\mu_1)\big((1-t) + t\sigma_e^2/2\big)}{t(1-t)\sigma_e^2 + \sigma_1^2 t^2 + (1-t)^2} \tag{38}$$

We can solve ODE with this field and get the expression for the trajectories, starting from the given point x_0:

$$x(t) = \mu_1 t + x_0\sqrt{t(1-t)\sigma_e^2 + \sigma_1^2 t^2 + (1-t)^2}. \tag{39}$$

These trajectories, for different x_0 are depicted in Fig. 6.



(a) Trajectories, $\sigma_e = 0.3$ (b) Vector field, $\sigma_e = 0.3$ (c) Trajectories, $\sigma_e = 1$ (d) Vector field, $\sigma_e = 1$

(e) Trajectories, $\sigma_e = 3$ (f) Vector field, $\sigma_e = 3$ (g) Trajectories, $\sigma_e = 10$ (h) Vector field, $\sigma_e = 10$

Figure 6: a) $N = 40$ random trajectories from $\mathcal{N}\left(\cdot \middle| 0, 1^2\right)$ to $\mathcal{N}\left(\cdot \middle| 2, 3^2\right)$ and 2D plot of the vector field in this case for different σ_e

At the limit $\sigma_e \to 0$ expressions (38) and (39) turn into expressions (35) and (36) as expected.

For the score s in the considered case we have

$$s(x,\, t) = \frac{t\mu_1 - x}{(1-t)^2 + t(1-t)\sigma_e^2 + t^2\sigma_1^2}$$

Thus, we can explicitly write expressions for the stochastic process for the evolution from the initial distribution rho_0 (standard Gaussian) to the final distribution ρ_1:

$$\mathrm{d}x(x) = \left[\frac{x\big(t\sigma_1^2 + (1-t)\sigma_e^2/2\big) - (x-\mu_1)\big((1-t) + t\sigma_e^2/2\big)}{t(1-t)\sigma_e^2 + \sigma_1^2 t^2 + (1-t)^2} + \right.$$
$$\left. + \frac{g^2(t)}{2}\frac{t\mu_1 - x}{(1-t)^2 + t(1-t)\sigma_e^2 + t^2\sigma_1^2}\right]\mathrm{d}t + g(t)\,\mathrm{d}W(t)\,.$$

Here $g(t)$ is arbitrary smooth function. In the case of Shrödinger Bridge we take $g(t) = \sigma_e\sqrt{t(1-t)}$.

E DETAIL ON THE SDE CASE

E.1 OPTIMAL VECTOR FIELD AND SCORE FOR STOCHASTIC MAP

Following Tong et al. (2024b) we consider a so-called *Brownian bridge* $B(t)$ from x_0 to x_1 with constant diffusion rate σ_e. This stochastic process can be expressed through a multidimensional standard Winner process $W(t)$ as

$$B(t \mid x_0, x_1) = (1-t)x_0 + tx_1 + \sigma_e(1-t)W\left(\frac{t}{1-t}\right). \tag{40}$$

Thus, the conditional distribution $p(t, x \mid x_0,\, x_1)$ conditioned on the starting x_0 and end point x_1 is Gaussian:

$$p(x, t \mid x_0,\, x_1) = \mathcal{N}\left(x\middle|\, (1-t)x_0 + tx_1, \sigma_e^2 t(1-t)\right).$$

We can not directly use the results Theorem 3 from Lipman et al. (2023) (or similar Theorem 2.1 from Tong et al. (2024a)) for the Gaussian paths, as in this case $\sigma(0) = 0$. To circumvent this obstacle and to be able to write an expression for the conditional velocity, we assume that we have a Gaussian distribution with a very narrow peak at the initial ($t = 0$) and final ($t = 1$) points. In other words, we will consider conditional probabilities of the form

$$p(x, t \mid x_0,\, x_1) = \mathcal{N}\left(x\middle|\, (1-t)x_0 + tx_1, \sigma_e^2(t+\eta)(1-t+\eta)\right), \tag{41}$$

where parameter η is small enough. Then we can use the above Theorems and immediately write

$$v_{x_0,x_1}(x,t) = \frac{\sigma'(t)}{\sigma(t)}\big(x-\mu(t)\big)+\mu'(t) = \frac{1-2t}{2(t+\eta)(1-t+\eta)}\big(x-(1-t)x_0-tx_1\big)+x_1-x_0. \tag{42}$$

After integrating over x_0 and x_1, we can take the limit $\eta \to 0$. Thus, now for fixed x_0 and x_1 we do not have a fixed value of x_t in which to train the model, but a random one. In general case, we end up to the loss:

$$\mathcal{L}_v = \mathbb{E}_{t\sim\mathcal{U}(0,1),\,\{x_1,x_0\}\sim\pi,\,x\sim p(\cdot,t|x_0,x_1)}\|v_\theta(x,\,t) - v_{x_0,x_1}(x,t)\|^2, \tag{43}$$

where $\pi(x_1,\, x_0)$ is the density of the joint distributions with the marginal equal to the two given probabilities:

$$\int \pi(x_1,\, x_0)\,\mathrm{d}x_1 = \rho_0(x_0), \quad \int \pi(x_1,\, x_0)\,\mathrm{d}x_0 = \rho_1(x_1).$$

In the simple case, $\pi(x_1,\, x_0) = \rho_0(x_0)\rho_1(x_1)$. Vector field in Eq. (43) if taken in the form of Eq. (42).

Now, we can obtain an explicit form for the vector field v at which the written loss is reached its minimum by performing the same calculations as in the derivation of formula (10):

$$v(x,t) = \frac{\iint v_{x_0,x_1}(x,t)\, p(x,t \mid x_0,\, x_1)\, \pi(x_0,x_1)\,\mathrm{d}x_0\,\mathrm{d}x_1}{\iint p(x,t \mid x_0,\, x_1)\, \pi(x_0,x_1)\,\mathrm{d}x_0\,\mathrm{d}x_1}. \tag{44}$$

As in the work Tong et al. (2024b) we can also train score network. Namely, as marginals for Brownian bridge are Gaussian, we can write explicit conditional score for conditional probabilistic path

$$\boldsymbol{\nabla}\log p(x,t \mid x_0,\, x_1) = \frac{\mu(t)-x}{\sigma_e^2(t)} = \frac{x_0(1-t)+x_1t-x}{\sigma_e^2 t(1-t)}.$$

In the work Tong et al. (2024b) the following loss is introduced to train a model for this score

$$\mathcal{L}_s = \mathbb{E}_{t\sim\mathcal{U}(0,1),\,\{x_1,x_0\}\sim\pi,\,x\sim p(\cdot,t|x_0,x_1)}\|s_\theta(x,t) - \nabla\log p(x,t \mid x_0,\, x_1)\|^2. \tag{45}$$

Similar to (44), for the optimal score s we have:

$$s(x,t) = \frac{\iint \boldsymbol{\nabla}\log p(x,t \mid x_0,\, x_1)\, p(x,t \mid x_0,\, x_1)\, \pi(x_0,x_1)\,\mathrm{d}x_0\,\mathrm{d}x_1}{\iint p(x,t \mid x_0,\, x_1)\, \pi(x_0,x_1)\,\mathrm{d}x_0\,\mathrm{d}x_1}, \tag{46}$$

where p is given in (41).

E.2 USE STOCHASTIC

Note that the obtained vector field gives marginal distributions $p(x,t)$, which (in the limit $\eta \to 0$) at $t = 1$ leads to the distribution we need: $p(x, t = 1) = \rho_1(x)$. However, the addition of the stochastic term allows us to extend the scope of application of the explicit formula for the vector field. In particular, it can be applied to the situation when we have two sets of samples and both distributions are unknown, as well as the possibility of constructing SDE and solving it using, for example, the Euler–Maruyama method (see examples below).

As consequence of Theorem 3.1 from Tong et al. (2024b) we have that, if v is given by Eq. (44) then ODE

$$\frac{\partial \rho(x,t)}{\partial t} = -\operatorname{div}\big(\rho(x,t)v(x,t)\big) \tag{47}$$

recovers the marginal $\rho(x,t)$ (with the given initial conditions) of the stochastic process $P(t)$ which is obtained by marginalization conditional Brownian bridge (40) over initial and target distribution

$$P(t) = \int B(t \mid x_0, x_1)\pi(x_0, x_1)\,\mathrm{d}x_0\,\mathrm{d}x_1\,.$$

As the second consequence of this Theorem, the SDE

$$\mathrm{d}x(t) = \Big(v\big(x(t),t\big) + \frac{g^2(t)}{2}s\big(x(t),t\big)\Big)\,\mathrm{d}t + g(t)\,\mathrm{d}W(t) \tag{48}$$

generates so-called Markovization of the process $P(t)$. Indeed, we can rewrite PDE Eq. (47) in the form

$$\frac{\partial \rho(x,t)}{\partial t} = -\operatorname{div}\Big(\rho(x,t)v(x,t) + \frac{g^2(t)}{2}\boldsymbol{\nabla}\rho(x,t)\Big) + \frac{g^2(t)}{2}\Delta\rho(x,t),$$

where nabla operator is defined as $\Delta = \operatorname{div}\boldsymbol{\nabla}$. Thus, we get the Fokker–Planck equation for the density of the stochastic process (48).

E.3 PARTICULAR CASES

In particular case of Brownian bridge when $\sigma_e(t) = \sigma_\epsilon\sqrt{t(1-t)}$, then $\sigma_e'(t) = \sigma_\epsilon(1-2t)/\big(2\sqrt{t(1-t)}\big)$. In this section we consider simple case of separable variables $\pi(x_0,x_1) = \rho_0(x_0)\rho_1(x_1)$.

E.3.1 GAUSSIAN INITIAL DISTRIBUTION

In the case, when ρ_0 is standard Gaussian distribution: $\rho_0 = \mathcal{N}\left(\cdot\middle|\, 0, 1^2\right)$, we can take integral on x_0 and then take the limit $\eta \to 0$ in the expressions for v and s. First, consider the expression for v: where we use explicit expression (41) for conditional density path and Eq. (42) for conditional velocity:

$$v(x,t) = \frac{\int w(x,t \mid x_1)\mathcal{N}\left(x\middle|\, x_1 t, \sigma_e^2 t(1-t) + (1-t)^2\right)\rho_1(x_1)\mathrm{d}x_1}{\int \mathcal{N}\left(x\middle|\, x_1 t, \sigma_e^2 t(1-t) + (1-t)^2\right)\rho_1(x_1)\mathrm{d}x_1} = \\ = \frac{\int w(x,t \mid x_1)\rho_0\Big(\frac{x - x_1 t}{\sqrt{\sigma_e^2 t(1-t) + (1-t)^2}}\Big)\rho_1(x_1)\mathrm{d}x_1}{\int \rho_0\Big(\frac{x - x_1 t}{\sqrt{\sigma_e^2 t(1-t) + (1-t)^2}}\Big)\rho_1(x_1)\mathrm{d}x_1}, \tag{49}$$

where $w(x,t \mid x_1)$ is the conditional velocity, generated by the conditional map $\phi_{t,x_1}(x) = \sqrt{\sigma_e^2 t(1-t) + (1-t)^2} + t x_1$:

$$w(x,t \mid x_1) = \frac{x_1 - x}{1 - t + t\sigma_e^2} + \sigma_e^2\frac{(1-2t)x + t x_1}{2\big((1-t)^2 + (1-t)t\sigma_e^2\big)}.$$

Thus, note that in the case of Gaussian distributions, all the difference between this expression and the expression without the stochastic part is the appearance of additional (time-dependent, in general) variance. Marginal distributions are still Gaussian's.

Similar, using Eq. (46) we have for the score s:

$$s(x,t) = \frac{\int (tx_1 - x)\mathcal{N}\left(x\middle|\, x_1 t, \sigma_e^2 t(1-t) + (1-t)^2\right)\rho_1(x_1)\mathrm{d}x_1}{\left((1-t)^2 + (1-t)t\sigma_e^2\right)\int \mathcal{N}\left(x|\, x_1 t, \sigma_e^2 t(1-t) + (1-t)^2\right)\rho_1(x_1)\mathrm{d}x_1} = \\ = \frac{\int (tx_1 - x)\rho_0\Big(\frac{x - x_1 t}{\sqrt{\sigma_e^2 t(1-t) + (1-t)^2}}\Big)\rho_1(x_1)\mathrm{d}x_1}{\left((1-t)^2 + (1-t)t\sigma_e^2\right)\int \rho_0\Big(\frac{x - x_1 t}{\sqrt{\sigma_e^2 t(1-t) + (1-t)^2}}\Big)\rho_1(x_1)\mathrm{d}x_1}. \quad (50)$$

E.3.2 Samples instead of distributions

Consider the case where we only have access to the samples $\{x_0^i\}_{i=1}^{N_0}$ and $\{x_1^i\}_{i=1}^{N_1}$ from both distributions, ρ_0 and ρ_1, but do not know their explicit expressions. In this case, we can estimate the vector field using by a method similar to the one we used to estimate the vector field in (15):

$$v(x,t) \approx \frac{\sum_{i=1}^{N_0}\sum_{j=1}^{N_1} v_{x_0^i, x_1^j}(x,t)\, p(x,t \mid x_0^i,\, x_1^j)}{\sum_{i=1}^{N_0}\sum_{j=1}^{N_1} p(x,t \mid x_0^i,\, x_1^j)}. \quad (51)$$

Similar for the score

$$s(x,t) \approx \frac{\sum_{i=1}^{N_0}\sum_{j=1}^{N_1} \boldsymbol{\nabla} p(x,t \mid x_0^i, x_1^j)\, p(x,t \mid x_0^i,\, x_1^j)}{\sum_{i=1}^{N_0}\sum_{j=1}^{N_1} p(x,t \mid x_0^i,\, x_1^j)}. \quad (52)$$

In addition, we can also use the importance sampling method in this case. Namely we can use both approaches: self-normalized importance sampling and rejection sampling, similar to what is described in Sec. B

F Consistency of Eq. (24) in the case of optimal transport

Let us analyze what happens if in formula (24) the joint density π represents the following Dirac delta-function[6]:

$$\pi(x_0, x_1) = \delta\big(x_0 - F(x_1)\big),$$

i. e. we have a deterministic mapping F from x_1 to x_0. Then, the Eq. (34) come to

$$v(x,t) = \frac{\int (x_1 - x)\,\delta\big(\phi_{t,x_1}^{-1}(x) - F(x_1)\big)\,\mathrm{d}x_1}{(1-t)\int \delta\big(\phi_{t,x_1}^{-1}(x) - F(x_1)\big)\,\mathrm{d}x_1}.$$

Let $y(x,t)$ be the unique solution of the equation

$$\phi_{t,y}^{-1}(x) = F(y), \quad (53)$$

considered as an equation on y. Then

$$v(x,t) = \frac{x - y(x,t)}{1-t}.$$

Now, let us use linear mapping $\phi_{t,x_1}(x) = x_1 t + x(1-t)$, with inverse $\phi_{t,x_1}^{-1}(x) = \frac{x - t x_1}{1-t}$, and consider the simplest case when the original distribution is a d-dimensional standard Gaussian and ρ_1 is a d-dimensional Gaussian with mean μ and diagonal variance $\Sigma = \mathrm{diag}(\sigma)$. We know the OT correspondence between Gaussians, namely

$$\big(F(x_1)\big)_i = \frac{(x_1 - \mu)_i}{\Sigma_{ii}}, \quad \forall 1 \geq i \geq d.$$

[6]Further reasoning is not absolutely rigorous, and in order not to introduce the axiomatics of generalized functions, we can assume that the delta function is the limit of the density of a normal distribution with mean 0 and variance tending to zero.

Here and further by index i we denote ith component of the corresponding vector. Then, the Eq. (53) reads as

$$\frac{(x - yt)_i}{1 - t} = \frac{(y - \mu)_i}{\Sigma_{ii}},$$

with the solution

$$\big(y(x,t)\big)_i = \frac{\mu_i(1-t) + x_i\Sigma_{ii}}{1 + (\Sigma_{ii} - 1)t}.$$

Then the expression for the vector field is

$$\big(v(x,t)\big)_i = \frac{\mu_i + x_i(\Sigma_{ii} - 1)}{1 + (\Sigma_{ii} - 1)t}.$$

Now, knowing the expression for velocity, we can write the equations for the trajectories $x(t)$:

$$\begin{cases} \big(x'(t)\big)_i = \dfrac{\mu_i + (x(t))_i(\Sigma_{ii} - 1)}{1 + (\Sigma_{ii} - 1)t}, \\ x(0)_i = (x_0)_i \end{cases}.$$

This equation have closed-form solution:

$$x(t) = \mu t + x_0 - (1 - \sigma)\, t x_0.$$

Analyzing the obtained solution, we conclude that, first, the trajectories obey the given mapping F:

$$\big(F(x(1))\big)_i = (x_0)_i = \frac{(x(1) - \mu)_i}{\Sigma_{ii}},$$

And, second, the trajectories are straight lines (in space), as they should be when the flow carries points along the optimal transport.

As a final conclusion, note that, of course, if we are mapping optimal transport F, then it is meaningless to use numerical formula (16). However, usually the exact value of the mapping F is not known, and our theoretical formula (34) can help to rigorously establish the error that is committed when an approximate mapping is used instead of the optimal one.

G Analytical derivations for example in Fig. 1(b)

G.1 CFM dispersion

To derive the analytical expression for the optimal flow velocity in the case of two normal distributions $\rho_0 \sim N(0, I)$ and $\rho_1 \sim N(\mu, \sigma^2 I)$, we start by substituting $\mu_0 = 0$, $\sigma_0 = 1$, $\mu_1 = \mu$, $\sigma_1 = \sigma$, to the exact expression (35) to get

$$v(x,t) = \frac{t\sigma^2 + t - 1}{(1-t)^2 + t^2\sigma^2}x + \frac{1}{(1-t)^2 + t^2\sigma^2}(\mu - t\mu) = w(t)x + C, \tag{54}$$

where

$$w(t) = \frac{t\sigma^2 + t - 1}{(1-t)^2 + t^2\sigma^2},$$

and C is constant independent of x. We then redefine the dispersion based on Eq. (19) using $x = (1-t)x_0 + tx_1$ with $x_0 \sim \rho_0$ and $x_1 \sim \rho_1$:

$$\mathbb{D}_{x,x_1} f(x,\, x_1) = \mathbb{D}_{x_0,x_1} f\big((1-t)x_0 + tx_1,\, x_1\big) \tag{55}$$

This leads us to the final expression:

$$\begin{aligned} \mathbb{D}_{x,x_1}\Delta v(x,t) &= \mathbb{D}_{x_0,x_1}((1 - w(t))x_1 - (1 + w(t)(1-t))x_0) = \\ &= (1 + w(t)(1-t))^2\mathbb{D}_{x_0}x_0 + (1 - w(t))^2\mathbb{D}_{x_1}x_1. \end{aligned}$$

This provides a comprehensive representation of the updated dispersion for the CFM objective at any given time t.

Algorithm 2 Computation ExFM dispersion algorithm

Require: Density function for initial distribution ρ_0; sampler for target distribution ρ_1; parameter M (number of samples for evaluation); parameter N (number of samples from ρ_1 for certain samples $x \sim \rho_m(x,t)$); optimal model $v(x,t)$; time for evaluation t.

Ensure: numerical evaluation of dispersion update for ExFM objective

1: Sample $(M \cdot N)$ samples $x_1^{i,j}$ from ρ_1, where $i \in [1, M]$ and $j \in [1, N]$

2: Sample (M) samples x_0^i from ρ_0, where $i \in [1, M]$

3: Compute points x^i as $(1-t)x_0^i + tx_1^{i,0}$

4: Compute $v^d(x^i, t) = \sum_{j=1}^{N} \tilde{\rho}^{i,j}(t) \frac{x_1^{i,j} - x^i}{1-t}$, where $\tilde{\rho}^{i,j}(t) = \rho_0\left(\frac{x^i - t x_1^{i,j}}{1-t}\right) / \sum_{j=1}^{N} \rho_0\left(\frac{x^i - t x_1^{i,j}}{1-t}\right)$

5: Compute and return dispersion $\mathbb{D}_i(v(x^i,t) - v^d(x^i,t))$

G.2 ExFM dispersion

The analytical derivation of the updated dispersion for the ExFM objective proves to be complex in practice. Therefore, for the example at hand, a numerical scheme was employed for evaluation. The procedure outlined in Alg. 2 was utilized for this task. The experiment's parameters for the algorithm were as follows: $M = 200k$, $N = 128$, $\rho_0 = N(0, I)$, $\rho_1 = N(\mu, \sigma^2 I)$, and the optimal model $v(x,t)$ was derived from equation (54).

H Additional Experiments

H.1 2D toy examples

To ensure the reliability and impartiality of the outcomes, we carried out the experiment under uniform conditions and parameters. Initially, we generated a training set of batch size $N = 10{,}000$ points. The employed model was a simple Multilayer Perceptron with ReLu activations and 2 hidden layers of 512 neurons, `Adam` optimizer with a learning rate of 10^{-3}, EMA with rate of 0.9 and no learning rate scheduler. We determined the number of learning steps equal to $15\,000$ and $30\,000$ learning steps for `rings` dataset since the more comprehensive structure of the data. Subsequently, we configured the mini batch size $n = 512$ during the training procedure, with the primary objective of minimizing the Mean Squared Error (MSE) loss. The full training algorithm and notations can be seen in Algorithm 1. To perform sampling, we employed the function `odeint` with `dopri5` method from the python package `torchdiffeq` with `atol` and `rtol` equal 10^{-5}.

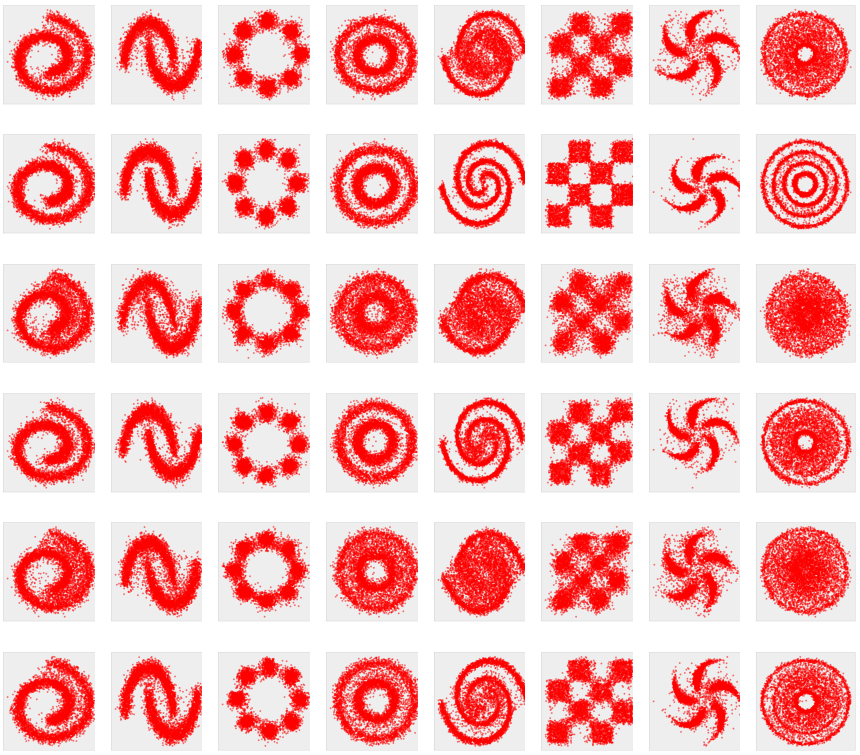
Table 5: Energy Distance comparison for ExFM, CFM and OT-CFM methods for 2D-toy datasets for $15\,000$ learning steps ($30\,000$ learning steps for `rings` dataset), mean and std taken from 10 sampling iterations.

Data	ExFM	CFM	OT-CFM
SWISSROLL	1.20E-03 ± 9.6E-04	1.58E-03 ± 6.4E-04	**8.28e-04 ± 3.12e-04**
MOONS	**5.58e-04 ± 3.45e-04**	1.27E-03 ± 8.2E-04	6.99E-04 ± 4.38E-04
8GAUSSIANS	**1.26e-03 ± 6.4e-04**	1.62E-03 ± 6.0E-04	1.88E-03 ± 8.0E-04
CIRCLES	**6.66e-04 ± 4.69e-04**	8.34E-04 ± 4.72E-04	9.70E-04 ± 5.40E-04
2SPIRALS	**8.15e-04 ± 2.91e-04**	1.91E-03 ± 7.7E-04	1.74E-03 ± 5.5E-04
CHECKERBOARD	**1.32e-03 ± 5.6e-04**	3.41E-03 ± 1.19E-03	2.00E-03 ± 1.00E-03
PINWHEEL	**8.65e-04 ± 6.12e-04**	2.48E-03 ± 8.8E-04	1.11E-03 ± 3.2E-04
RINGS	**5.75e-04 ± 3.61e-04**	1.53E-03 ± 4.3E-04	1.19E-03 ± 3.6E-04

We present visual and quantitative results to evaluate the performance of our proposed method, ExFM. Visualizations of the learned distributions are presented in Figure 7. The corresponding data densities can be found in Figure 8. We sampled data from both the beginning and end of the training process. The results clearly show that ExFM outperforms the baseline CFM and the OT-CFM, particularly on the `rings` dataset. This can be attributed to ExFM's ability to effectively capture the complexities of this challenging distribution.

To further support the effectiveness of ExFM, we analyzed the training losses. The complete progression of these losses is visualized in Figure H.1. This figure highlights the significantly lower variance observed in ExFM's training loss compared to the CFM method.

For quantitative evaluation, we employed the Energy Distance metric and Wasserstein distance. The results of Energy Distance are presented in Table 5, Wasserstein distance in Table 2 while Figure H.1 showcases the progression of this metric during the training procedure. Interestingly, CFM, OT-CFM and ExFM models achieve rapid convergence in terms of this metric at the beginning of learning. Additionally, the metric values remain relatively stable throughout the training process. However, the superior visual quality achieved by ExFM (as observed in Figure 7) suggests that the Energy Distance metric might not be the most suitable choice for evaluating this specific task.

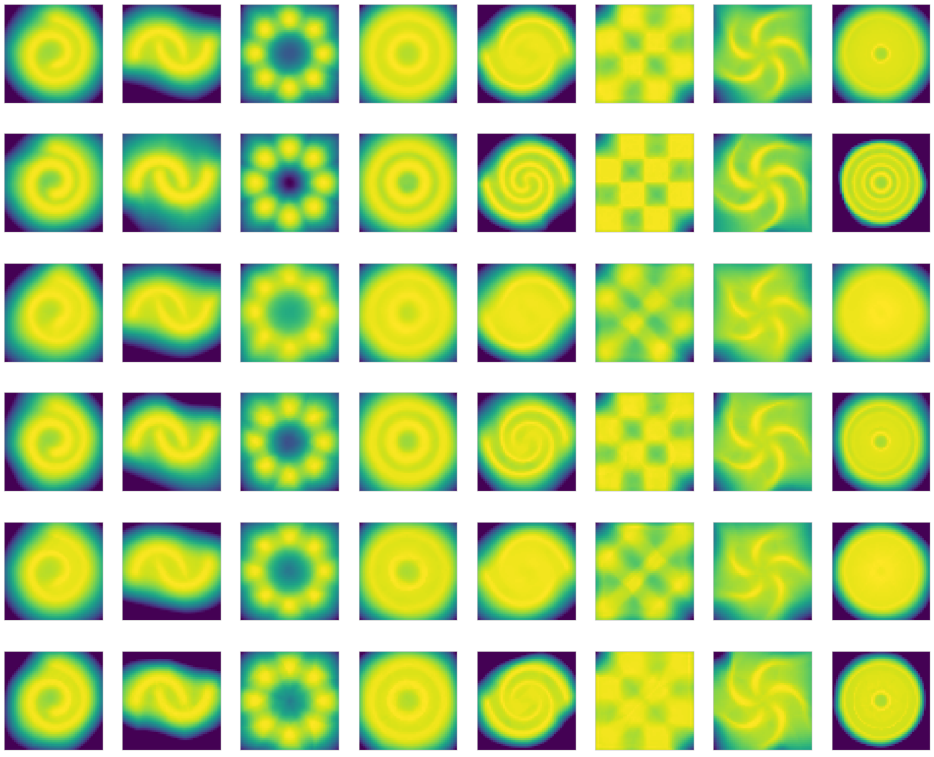


(a) swissroll (b) moons (c) 8gaussians (d) circles (e) 2spirals (f) checkerboard (g) pinwheel (h) rings

Figure 7: Visual comparison of methods on toy 2D data. First and second rows sampled by ExFM, third and fourth rows sampled by CFM, fifth and six rows sampled by OT-CFM. The upper row in pairs of the same method sampled after 1 500 learning iterations (3 000 for `rings` dataset), the lower row in pairs of the same method sampled after 15 000 learning iterations (30 000 for `rings` dataset).

H.2 TABULAR

The `power` dataset (dimension = 6, train size = 1659917, test size = 204928) consisted of electric power consumption data from households over a period of 47 months. The `gas` dataset (dimension = 8, train size = 852174, test size = 105206) recorded readings from 16 chemical sensors exposed to gas mixtures. The `hepmass` dataset (dimension = 21, train size = 315123, test size = 174987) described Monte Carlo simulations for high energy physics experiments. The `minibone` (dimension = 43, train size = 29556, test size = 3648) dataset contained examples of electron neutrino and muon neutrino. Furthermore, we utilized the `BSDS300` dataset (dimension = 63, train size = 1000000, test size = 250000), which involved extracting random 8 x 8 monochrome patches from the BSDS300 datasets of natural images Martin et al. (2001).



(a) swissroll (b) moons (c) 8gaussians (d) circles (e) 2spirals (f) checkerboard (g) pinwheel (h) rings

Figure 8: Densities comparison of methods on toy 2D data. First and second rows sampled by ExFM, third and fourth rows sampled by CFM, fifth and six rows sampled by OT-CFM. The upper row in pairs of the same method sampled after 1500 learning iterations (3000 for `rings` dataset), the lower row in pairs of the same method sampled after $15,000$ learning iterations ($30,000$ for `rings` dataset).

These diverse multivariate datasets are selected to provide a comprehensive evaluation of performance across various domains. To maintain consistency, we followed the code available at the given GitHub link[7] to ensure that the same instances and covariates were used for all the datasets.

To ensure the correctness of the experiments we conduct them with the same parameters. To train the model we use the same MultiLayer Perceptron model with ReLu activations, number of neurons and layers differed for the datasets along with the learning rate for the optimizer, that can be seen in Table 6. Same as for toy data, we use `Adam` as optimizer, EMA with 0.9 rate and no learning rate scheduler. As in the pretrained step, we use separately training and testing sets for training the model and calculating metrics. We train the models for $10,000$ learning steps with batch size $N = 5000$ (`batch_size`) and mini batches $n = 256$ elements (`mini_batch_size`).

For both 2D-toy and tabular data: we take $m = n$ time variable, individual value of variable t corresponds to its pair (x_0, x_1). The notations N, n and m corresponds to those in Algorithm 1. To perform sampling, we employed the function `odeint` with `dopri5` method from the python package `torchdiffeq` with `atol` and `rtol` equal 10^{-5}.

Due to the inherent difficulty in visualizing tabular datasets, Negative Log-Likelihood (NLL) (Table 3) metrics were employed to quantitatively compare the performance of ExFM, CFM, and OT-CFM methods. Figure H.2 presents a comparison of the training losses incurred by each method. As can be observed, all three methods exhibit rapid convergence. Notably, our proposed method demon-

[7] `https://github.com/gpapamak/maf`

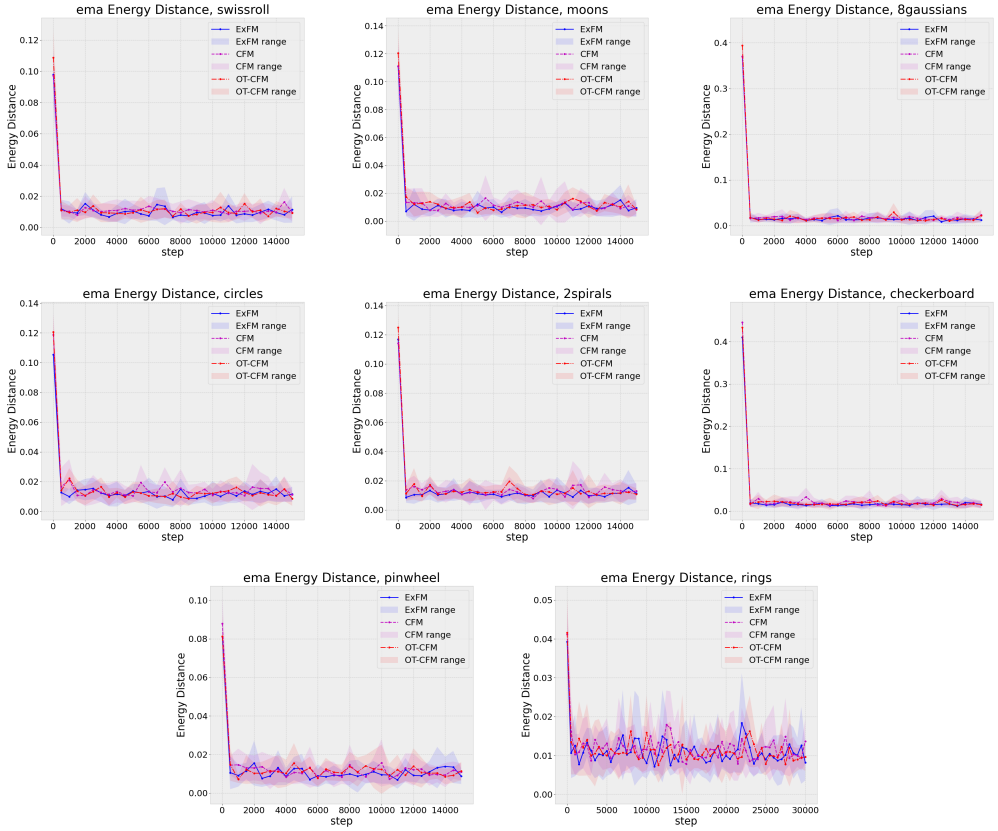


Figure 9: Energy distance comparison for ExFM, CFM and OT-CFM methods for toy datasets for 15 000 learning steps, 30 000 learning steps for `rings` dataset.

Table 6: Learning parameters for Tabular datasets.

DATA	MLP LAYERS	LR
POWER	[512, 1024, 2048]	1E-3
GAS	[512, 1024,1024]	1E-4
HEPMASS	[512, 1024]	1E-3
BSDS300	[512, 1024,1024]	1E-4
MINIBOONE	[512, 1024]	1E-3

strates superior training stability compared to the baseline CFM, as evidenced by its smoother loss curve.

In Figure H.2, we illustrate the NLL values recorded across training steps for all three methods on various datasets. While our method achieves competitive performance, it occasionally yields slightly lower NLL scores compared to OT-CFM on specific datasets.

H.3 EXFM-S EVALUATION

The models were assessed using four toy datasets of two dimensions each. A three-layer MLP network was utilized, featuring SeLU activations and a hidden dimension of 64. Optimization was carried out using the AdamW optimizer with a learning rate of 10^{-3} and a weight decay of 10^{-5}. The model was trained over 2 000 iterations with a batch size of 128. Inference was conducted using the Euler solver for Ordinary Differential Equations (ODE) with 100 steps. To validate the models, the POT library was employed to compute the Wasserstein distance based on 4 000 samples. The experiments were performed on a single Nvidia H100 GPU with 80gb memory.

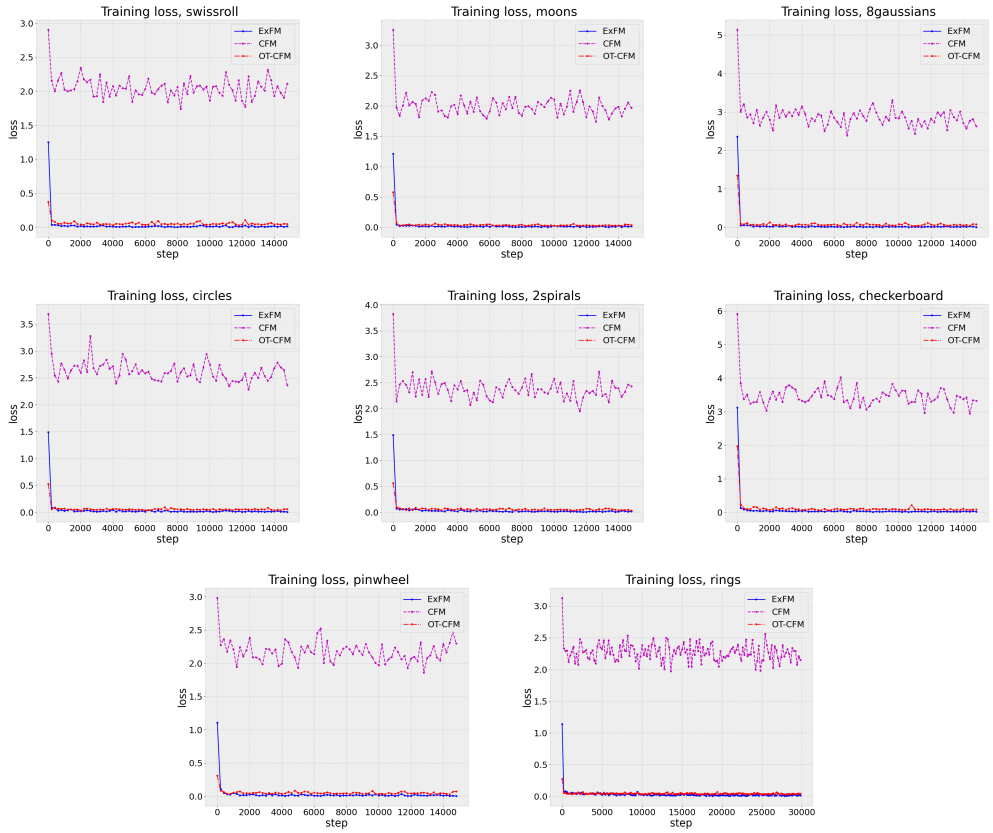


Figure 10: Training loss comparison for ExFM, CFM and OT-CFM methods for toy datasets for 15 000 learning steps, 30 000 learning steps for `rings` dataset.

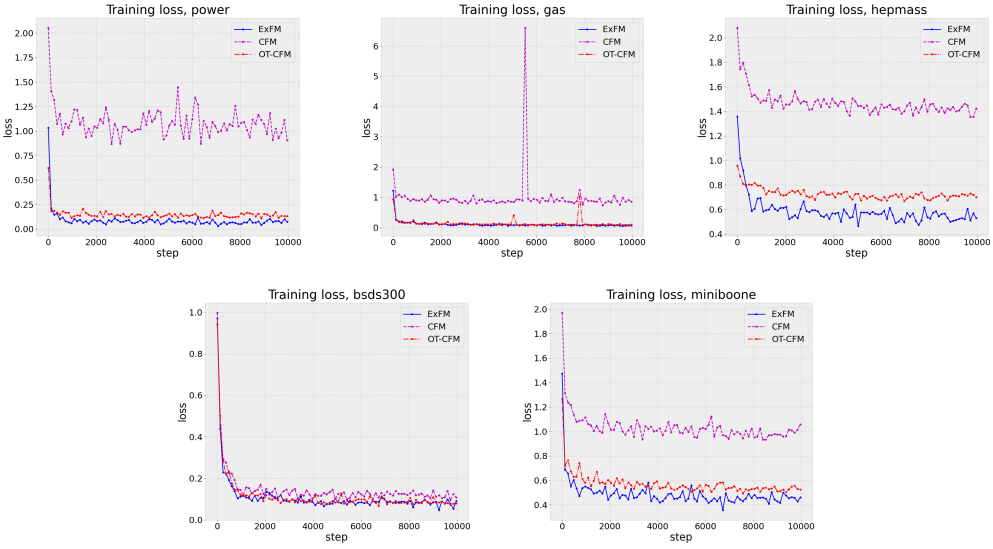


Figure 11: Training loss comparison for tabular datasets for ExFM, CFM and OT-CFM methods over 10 000 learning steps.

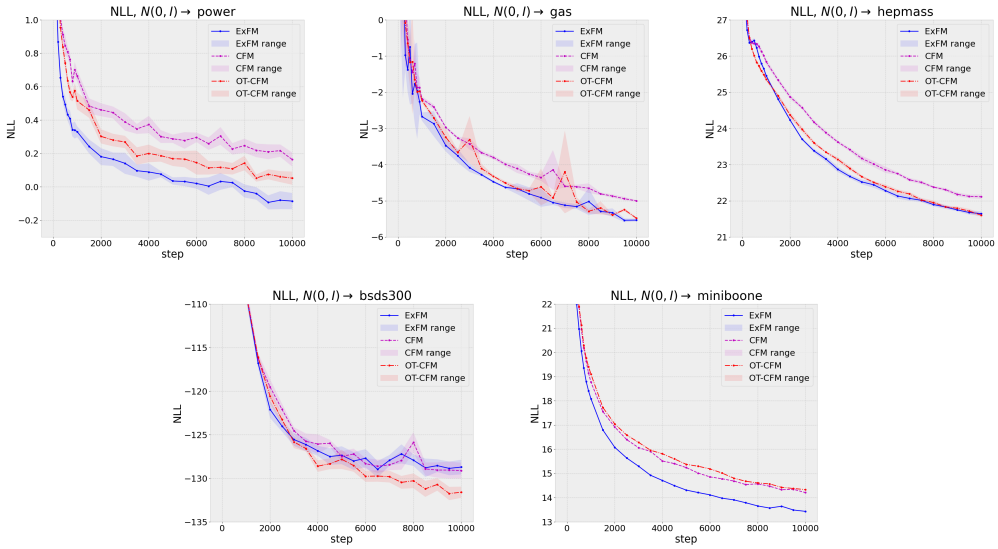


Figure 12: NLL comparison for ExFM, CFM and OT-CFM methods over 10 000 learning steps, mean and std for range taken from 10 sampling iterations.

H.4 CIFAR 10 AND MNIST

We conducted experiments related to high dimensional data, the parameters for training were taken from the open-source code[8] from the works Tong et al. (2024b;a). For training we used proposed U-Net model. We saved the leverage of additional heuristics(EMA, lr scheduler) and their parameters. For the final evaluation of CIFAR 10 dataset we used Fréchet inception distance (FID) metrics, and the values can be seen in Table 7, and we also evaluated FID during training for different learning steps, that can be seen in Table 8 and in Figure H.4.

Table 7: FID comparison for 4 sampling iterations, 400 000 learning steps.

METHOD	FID
EXFM	$\mathbf{3.686 \pm 0.029}$
CFM	3.727 ± 0.026
OT-CFM	3.843 ± 0.033

Our proposed method demonstrates competitive performance on the evaluated datasets. Notably, it consistently achieves slightly better results compared to OT-CFM. This observation aligns with the assumption that highlight the limitations of OT when dealing with high-dimensional data. Figures H.4 and H.4 illustrate the training loss curves for CIFAR 10 and MNIST, respectively. As evident from the figures, our method exhibits a clear advantage in terms of achieving lower training losses throughout the training process. This suggests that our method converges more effectively and is potentially more stable compared to OT-CFM and CFM. Visuals of generated samples for CIFAR 10 dataset are included in Figure H.4 and for MNIST dataset in Figure H.4.

H.5 METRICS

For evaluating 2D toy data we use Energy Distance and W2 metrices, for Tabular datasets we use Negative Log Likelihood, for CIFAR10 we took Fréchet inception distance (FID) metrics. This choice is connected with an instability and poor evaluation quality of Energy Distance metrics and $\mathcal{W}_2$ among high-dimensional data .

[8] https://github.com/atong01/conditional-flow-matching

Table 8: FID comparison for ExFM, CFM and OT-CFM methods over 400 000 learning steps, mean and std taken from 4 sampling iterations.

Step	ExFM FID	CFM FID	OT-CFM FID
0	447.256 ± 0.116	447.106 ± 0.130	447.091 ± 0.081
20000	281.060 ± 0.243	275.044 ± 0.123	281.499 ± 0.287
40000	52.050 ± 0.245	51.436 ± 0.142	45.976 ± 0.109
60000	**9.125 ± 0.060**	9.181 ± 0.035	10.358 ± 0.054
80000	**6.624 ± 0.053**	6.978 ± 0.062	7.492 ± 0.050
100000	**5.641 ± 0.048**	5.894 ± 0.045	6.299 ± 0.031
120000	**5.085 ± 0.031**	5.247 ± 0.051	5.558 ± 0.017
140000	**4.766 ± 0.036**	4.902 ± 0.053	5.120 ± 0.043
160000	**4.486 ± 0.054**	4.593 ± 0.068	4.828 ± 0.046
180000	**4.294 ± 0.023**	4.447 ± 0.045	4.576 ± 0.051
200000	**4.180 ± 0.029**	4.204 ± 0.013	4.434 ± 0.031
220000	**4.022 ± 0.036**	4.182 ± 0.024	4.331 ± 0.036
240000	**3.925 ± 0.028**	4.037 ± 0.036	4.227 ± 0.050
260000	**3.852 ± 0.047**	3.937 ± 0.018	4.125 ± 0.015
280000	**3.842 ± 0.053**	3.870 ± 0.040	4.056 ± 0.029
300000	**3.758 ± 0.032**	3.788 ± 0.024	4.017 ± 0.029
320000	**3.749 ± 0.029**	3.792 ± 0.034	3.937 ± 0.052
340000	**3.724 ± 0.042**	3.747 ± 0.033	3.897 ± 0.037
360000	**3.714 ± 0.022**	3.751 ± 0.041	3.875 ± 0.015
380000	**3.707 ± 0.028**	3.754 ± 0.020	3.917 ± 0.037
400000	**3.686 ± 0.029**	3.727 ± 0.026	3.843 ± 0.033

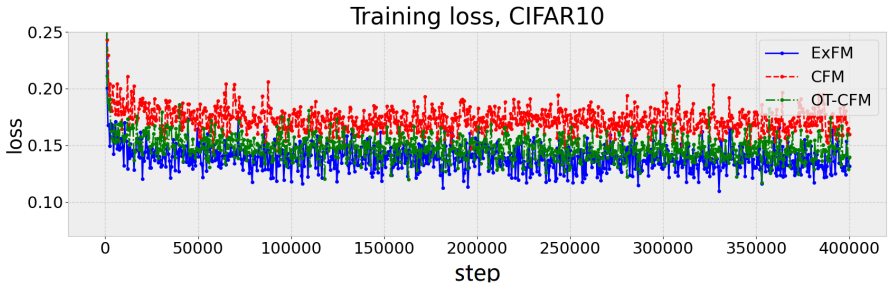


Figure 13: Training loss comparison for ExFM, CFM and OT-CFM methods, CIFAR-10 dataset.

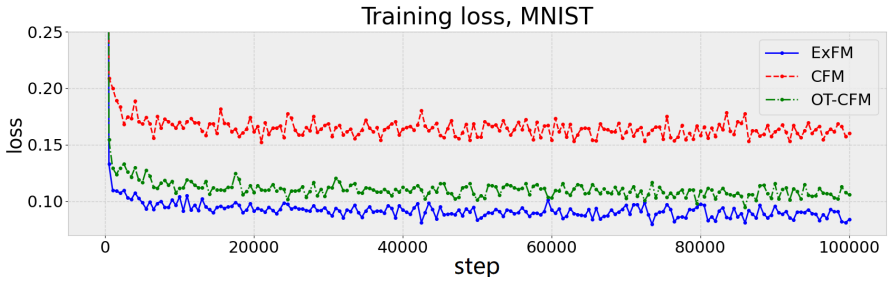


Figure 14: Training loss comparison for ExFM, CFM and OT-CFM methods, MNIST dataset.

H.5.1 Energy Distance

We use the generalized Energy Distance Székely (2003) (or E-metrics) to the metric space.

Consider the null hypothesis that two random variables, X and Y, have the same probability distributions: $\mu = \nu$.

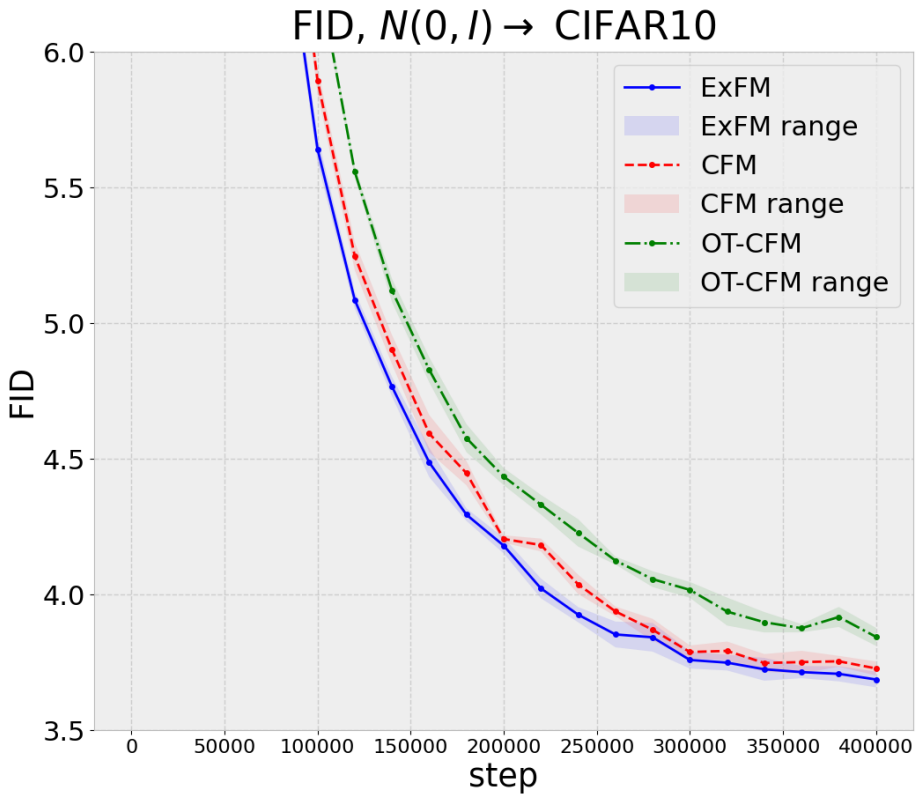


Figure 15: FID comparison for ExFM, CFM and OT-CFM methods, CIFAR-10 dataset.

For statistical samples from $X$and Y:

$$\{x_1, \ldots, x_n\} \quad \text{and} \quad \{y_1, \ldots, y_m\},$$

the following arithmetic averages of distances are computed between the X and the Y samples:

$$A = \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} \|x_i - y_j\|, \quad B = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \|x_i - x_j\|, \quad C = \frac{1}{m^2} \sum_{i=1}^{m} \sum_{j=1}^{m} \|y_i - y_j\|.$$

The E-statistic of the underlying null hypothesis is defined as follows:

$$E_{n,m}(X, Y) := 2A - B - C$$

H.5.2 2-WASSERSTEIN DISTANCE ($\mathcal{W}_2$)

The 2-Wasserstein distance Ramdas et al. (2017), also called the Earth mover's distance or the optimal transport distance W is a metric to describe the distance between two distributions, representing two different subsets A and B. For continuous distributions, it is:

$$W := W(F_A, F_B) = \left(\int_0^1 \left| F_A^{-1}(u) - F_B^{-1}(u) \right|^2 du \right)^{\frac{1}{2}},$$

where F_A and F_B are the corresponding cumulative distribution functions and F_A^{-1} and F_B^{-1} the respective quantile functions.

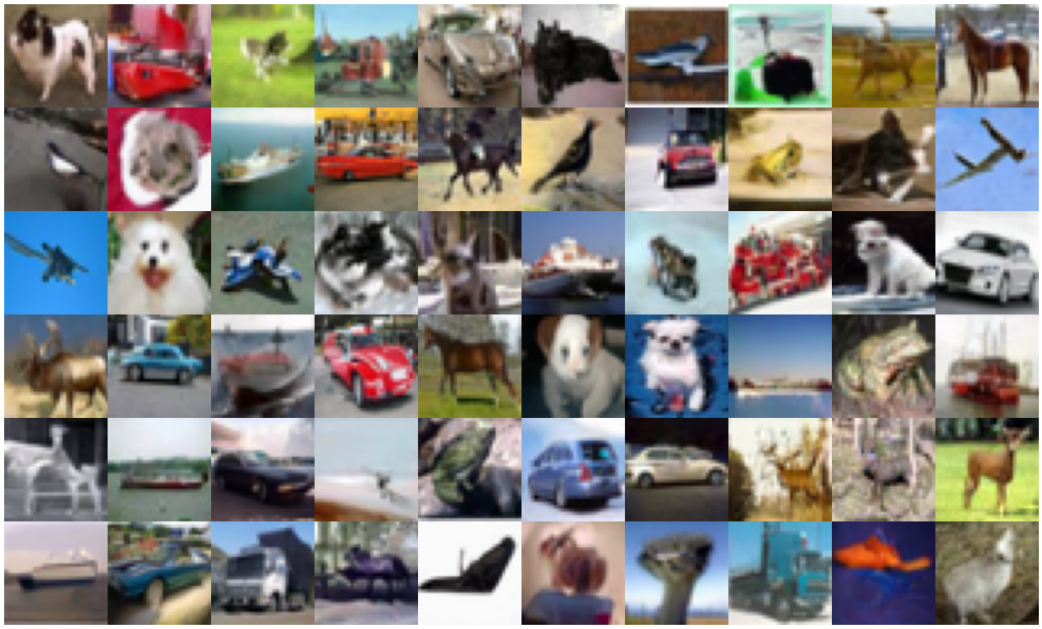


Figure 16: Sampled images from ExFM method, CIFAR-10 dataset.

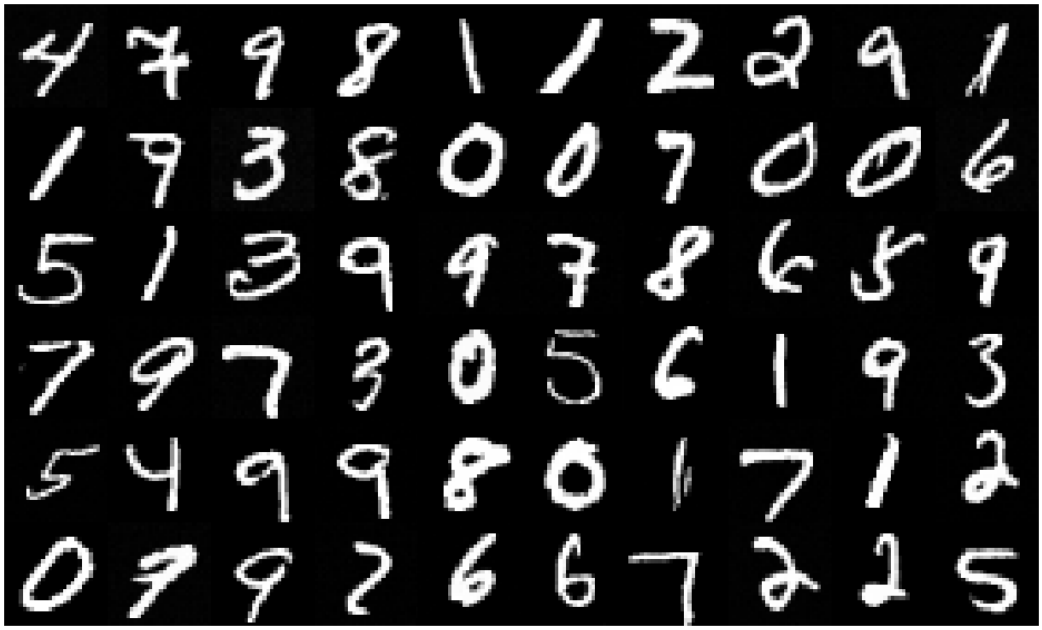


Figure 17: Sampled images from ExFM method, MNIST dataset.

H.5.3 Negative Log Likelihood (NLL)

To compute the NLL, we follow Lipman et al. (2023), Appendix C, Eq. (27)–(33).

Namely, we first sample $N = 5000$ samples $\{x_i^s\}_{i=1}^N$ from the target distribution. Then we solve the following inverse flow ODE:

$$\frac{d}{dt}\begin{bmatrix} x(t) \\ f(t) \end{bmatrix} = \begin{bmatrix} v_\theta(x(t), t) \\ -\text{div}(v_\theta(x(t), t)) \end{bmatrix}$$

for t from 1 to 0 with initial condition

$$\begin{bmatrix} x(1) \\ f(1) \end{bmatrix} = \begin{bmatrix} x^s \\ 0 \end{bmatrix},$$

where x^s is one of the sampled points.

For simplicity, changing time variable $\tau = 1 - t$ we solve the following ODE:

$$\frac{d}{d\tau} \begin{bmatrix} x(\tau) \\ f(\tau) \end{bmatrix} = \begin{bmatrix} -v_\theta(x(\tau), 1-\tau) \\ \text{div}(v_\theta(x(\tau), 1-\tau)) \end{bmatrix}$$

for τ from 0 to 1 with initial condition

$$\begin{bmatrix} x(0) \\ f(0) \end{bmatrix} = \begin{bmatrix} x^s \\ 0 \end{bmatrix}.$$

Thus we obtain N solutions for the spatial variables $\{x_i^0\}_{i=1}^N$ and N solutions $\{f_i^0\}_{i=1}^N$ for the values of f. For the probabilities at $t = 0$ we have

$$\log \rho(x^0, 0) = \log \rho(x^1, 1) - f^0.$$

We expect these N solutions to be distributed according to the standard normal distribution $\mathcal{N}(x \mid 0, I)$. So we calculate NLL as

$$\text{NLL} = -\frac{1}{N} \sum_{i=1}^{N} \Big(\ln \mathcal{N}(x_i^0 \mid 0, I) + f_i^0 \Big).$$

Technically, we calculate the `div` function using the `torch.autograd.grad` function from the `torch` package.

H.5.4 FRÉCHET INCEPTION DISTANCE (FID)

For images evaluation we take Fréchet inception distance (FID) metrics, in particular the implementation from Parmar et al. (2022). The main idea of FID metrics is to measure the gap between two data distributions, such as between a training set and samples from a trained model. After resizing the images, and feature extraction, the mean $(\mu, \hat{\mu})$ and covariance matrix $(\Sigma, \hat{\Sigma})$ of the corresponding features are used to compute FID:

$$\text{FID} = ||\mu - \hat{\mu}||_2^2 + \text{Tr}\big(\Sigma + \widehat{\Sigma} - 2(\Sigma\hat{\Sigma})^{1/2}\big),$$

where Tr is the trace of a matrix.