Skoltech

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Optimal Flow Matching: новый подход к генеративному моделированию и оптимальному транспорту с прямыми траекториями после одной процедуры минимизации

Магистерская образовательная программа: Науки о данных

Студент: _____ Никита Корнилов
подпись

Научный руководитель: _____ Александр Гасников
подпись
д.ф.-м.н., профессор

Со-руководитель _____ Александр Коротин
подпись
к.ф.-м.н., научный сотрудник

Москва 2025

Optimal Flow Matching: новый подход к генеративному моделированию и оптимальному транспорту с прямыми траекториями после одной процедуры минимизации

Никита Корнилов

Представлено в Сколковский институт науки и технологий
Июнь 18

Аннотация

В последние несколько лет в области генеративного моделирования активно развиваются Flow Matching (FM) методы. Одно из интригующих свойств этих методов, это возможность обучать потоки с прямыми траекториями, реализующими оптимальные транспортные (Optimal Transport, OT) перемещения. Прямолинейность траекторий критически важна для быстрого и качетсвенного семплирования из моделей.

Однако большинство существующих методов спрямления траекторий основаны на нетривиальных итеративных процедурах, которые накапливают ошибку в процессе обучения, или используют эвристики, опирающиеся на мини-батч OT.

Чтобы исправить эти недостатки, я разработал и теоретически обосновал новый подход — Optimal Flow Matching (OFM), который позволяет восстановить решение оптимального транспорта для квадратичной функции потерь всего за один шаг FM минимизации. Основная идея этого подхода заключается в использовании прямых векторных полей, параметризованных выпуклыми функциями.

Contents

Chapter 1
Introduction

Recent success in generative modeling [1, 2, 3] is mostly driven by Flow Matching (FM) [4] models. These models move a known distribution to a target one via ordinary differential equations (ODE) describing the mass movement. However, such processes usually have curved trajectories, resulting in time-consuming ODE integration for sampling. To overcome this issue, researches developed several improvements of the FM [5, 6, 7], which aim to recover more straight paths.

Rectified Flow (RF) method [5, 6] iteratively solves FM and gradually rectifies trajectories. Unfortunately, in each FM iteration, it **accumulates the error**, see [6, §2.2] and [5, §6]. This may spoil the performance of the method. The other popular branch of approaches to straighten trajectories is based on the connection between straight paths and Optimal Transport (OT) [8]. The main goal of OT is to find the way to move one probability distribution to another with the minimal effort. Such OT maps are usually described by ODEs with straight trajectories. In OT Conditional Flow Matching (OT-CFM) [7, 9], the authors propose to apply FM on top of OT solution between batches from considered distributions. Unfortunately, such a heuristic does not guarantee straight paths because of **minibatch OT biases**, see, e.g., [9, Figure 1, right] for the practical illustration.

Statements for defense. In my thesis, I fix the above-mentioned problems of the straightening methods. I propose a novel Optimal Flow Matching (OFM) approach that after a **single** FM iteration obtains straight trajectories which can be simulated without ODE solving. It recovers OT flow for the quadratic transport cost function, i.e., it solves the Benamou–Brenier problem.

Scientific novelty. The main idea of our OFM is to consider during FM only specific vector fields which yield straight paths by design. These vector fields are the gradients of convex functions, which in practice are parametrized by Input Convex Neural Networks [10]. In OFM, one can optionally use minibatch OT or any other transport plan as the input, and this is completely theoretically justified.

Chapter 2
Literature review

Firstly, I give theoretical background to Optimal Transport needed for proofs.

Dynamic Optimal Transport

Before generative modeling gained their popularity, researchers mostly studied related Dynamic Optimal Transport Problem (Dynamic OT) [11]. The main goal of Dynamic OT is to find the vector field u that moves the probability distribution p_0 to the distribution p_1 with the minimal efforts. Such maps are usually described by ODEs with straight trajectories. The intermediate distribution generated by u at the time $t \in [0, 1]$ is denoted as p_t^u . Namely, Dynamic OT is the following minimization problem:

$$\begin{aligned} \mathbb{W}_2^2(p_0, p_1) = \min_u \quad & \int_0^1 \int_{\mathbb{R}^D} \frac{\|u_t(x_t)\|_2^2}{2} p_t^u(x_t) dx dt, \\ s.t. \quad & p_1^u = p_1. \end{aligned} \tag{2.1}$$

In (2.1), one looks for the vector fields u that define the flows which start at p_0 and end at p_1. Among such flows, one seeks for the field which has minimal kinetic energy over the time interval. In practice, OT problem is usually solved without dynamic using static statement.

Static Optimal Transport

Monge's and Kantorovich's formulations. Monge's Optimal Transport formulation is given by

$$\inf_{T\#p_0 = p_1} \int_{\mathbb{R}^D} c(x_0, T(x_0)) p_0(x_0) dx_0, \tag{2.2}$$

where the infimum is taken over measurable functions $T : \mathbb{R}^D \to \mathbb{R}^D$ which satisfy the mass-preserving constraint $T\#p_0 = p_1$. Such functions are called transport maps. If there exists a transport map T^* that achieves the infimum, then it is called the optimal transport map.

Since the optimal transport map T^* in Monge's formulation may not exist, there is Kantorovich's relaxation for problem (2.2) which addresses this issue. Consider the set of transport plans $\Pi(p_0, p_1)$, i.e., the set of joint distributions on $\mathbb{R}^D \times \mathbb{R}^D$ which marginals are equal to p_0 and p_1, respectively. Kantorovich's Optimal Transport formulation is

$$\inf_{\pi \in \Pi(p_0, p_1)} \int_{\mathbb{R}^D \times \mathbb{R}^D} c(x_0, x_1) \pi(x_0, x_1) dx_0 dx_1. \tag{2.3}$$

With mild assumptions on p_0, p_1, the infimum is always achieved (possibly not uniquely). An optimal plan $\pi^* \in \Pi(p_0, p_1)$ is called an optimal transport plan. If optimal π^* has the form $[\mathrm{id}, T^*]\#p_0$, then T^* is the solution of Monge's formulation (2.2).

Quadratic cost function. In the case of the quadratic cost function $c(x_0, x_1) = \frac{\|x_0 - x_1\|^2}{2}$, infimums in both Monge's and Kantorovich's OT are always uniquely attained [8, Brenier's Theorem 2.12]. They are related by $\pi^* = [\mathrm{id}, T^*]\#p_0$. Moreover, the optimal values of (2.2) and (2.3)

are equal to each other. The square root of the optimal value is called the Wasserstein-2 distance $\mathbb{W}_2(p_0, p_1)$ between distributions p_0 and p_1, i.e.,

$$\mathbb{W}_2^2(p_0, p_1) := \min_{\pi \in \Pi(p_0, p_1)} \int_{\mathbb{R}^D \times \mathbb{R}^D} \frac{\|x_1 - x_0\|^2}{2} \pi(x_0, x_1) dx_0 dx_1$$
$$= \min_{T \# p_0 = p_1} \int_{\mathbb{R}^D} \frac{\|x_0 - T(x_0)\|^2}{2} p_0(x_0) dx_0. \tag{2.4}$$

The problem (2.4) has the equivalent dual form [8]:

$$\mathbb{W}_2^2(p_0, p_1) = \text{Const}(p_0, p_1) - \min_{\text{convex } \Psi} \underbrace{\left[\int_{\mathbb{R}^D} \Psi(x_0) p_0(x_0) dx_0 + \int_{\mathbb{R}^D} \overline{\Psi}(x_1) p_1(x_1) dx_1 \right]}_{=: \mathcal{L}_{OT}(\Psi)}, \tag{2.5}$$

where the minimum is taken over convex functions $\Psi(x) : \mathbb{R}^D \to \mathbb{R}$.
Here $\overline{\Psi}(x_1) := \sup_{x_0 \in \mathbb{R}^D} [\langle x_0, x_1 \rangle - \Psi(x_0)]$ is the convex (Fenchel) conjugate function of Ψ. It is also convex. The term $\text{Const}(p_0, p_1)$ does not depend on Ψ. Therefore, the minimization (2.4) over transport plans π is equivalent to the minimization of $\mathcal{L}_{OT}(\Psi)$ from (2.5) over convex functions Ψ. Moreover, the optimal transport map T^* can be expressed via an optimal Ψ^* (the *Brenier potential* [8]), namely,

$$T^* = \nabla \Psi^*. \tag{2.6}$$

The solution u^* for the dynamic OT (2.1) is constructed to generate straight trajectories connecting points x and $\nabla \Psi^*(x), \forall x \in \mathbb{R}^D$.

OT Solvers. There exist a variety of continuous OT solvers [12, 13, 14, 15, 16, 17, 18, 19, 20], which descriptions can be found in the survey [21]. ICNN-based solvers [13, 14, 20] directly minimize objective $\mathcal{L}_{OT}$ from (2.5) parametrizing a class of convex functions with convex in input neural networks called ICNNs [10]. Solvers details may differ, but the main idea remains the same. To calculate the conjugate function $\overline{\Psi}(x_1)$ at the point x_1, they solve the convex optimization problem from conjugate definition.

Flow Matching (FM)

Flow Matching [4] was the first approach to successfully apply diffusion ideas to arbitrary distributions. The authors suggest and theoretically and practically justify the novel FM loss function for the robust and effective training. They encourage the vector field u to follow the direction $x_1 - x_0$ of the linear interpolation $x_t = (1-t)x_0 + tx_1$ at any moment $t \in [0, 1]$, where x_0 and x_1 are sampled from a joint distribution π which martingale equal to p_0 and p_1, respectively. Distribution π is called transport plan. It is achieved via solving:

$$\min_u \int_0^1 \int_{\mathbb{R}^D \times \mathbb{R}^D} \|u_t(x_t) - (x_1 - x_0)\|^2 \pi(x_0, x_1) dx_0 dx_1 dt,$$
$$x_t = (1-t)x_0 + tx_1.$$

The set of trajectories generated by the solution of FM has a useful property: the generated final distribution equals to the p_1 for any initial transport plan π or conditional vector field v. Moreover, marginal distributions equal to the distribution $p_t(x_t|x_1), x_1 \sim p_1$. This feature is called marginal preserving property. To push point x_0 according to the learned vector field u, one needs to integrate ODE $dx_t = u_t(x_t)dt$ via numerical solvers. The vector fields with straight (or nearly straight) paths incur much smaller time-discretization error and increase effectiveness of computations, which is

in high demand for applications. Usually, the vector field u is parametrized via standard neural networks. Loss function is optimized with stochastic solvers. To solve ODE one can use numerical integration techniques, in particular, Runge–Kutta methods.

Drawbacks. FM approach inherits the main drawback of diffusion models: it usually generates curved trajectories, resulting in time-consuming ODE integration for sampling and large number of function estimations. Researchers noticed that some initial plans π can result in more straight paths after FM rather than the standard independent plan $p_0 \times p_1$. The two most popular modifications are Optimal Transport Conditional Flow Matching [7] and Rectified Flow [6, 22, 5].

Action Matching

In work [23], the authors propose a novel method called Action Matching for learning a broad class of dynamics using only independent samples from the system's temporal evolution. This approach provides a tractable training objective that avoids explicit assumptions about the underlying dynamics and eliminates the need for backpropagation through differential equations or optimal transport solvers. Building on connections with optimal transport, Action Matching can be extended to stochastic differential equations and dynamics involving probability mass creation and destruction.

In comparison with FM, this approach learns arbitrary already established system dynamics between two distributions via intermediate samples, and, in FM, the system evolution is set manually beforehand.

Optimal Transport Conditional Flow Matching (OT-CFM)

The first branch of research [7] is dedicated to the incorporating properties of OT solutions to FM. If one uses the dynamic OT plan π^* as the initial plan for FM, then it returns the vector field u^* which generates exactly straight trajectories. However, typically, the true OT plan π^* is not available. In such a case, in order to achieve some level of straightness in the learned trajectories, a natural idea is to take the initial plan π to be close to the optimal π^*. Inspired by this, the authors of OT-CFM take the advantage of minibatch OT plan approximation and achieve much better practical results in comparison with default FM. Firstly, they independently sample batches of points from p_0 and p_1. Secondly, they join the batches together according to the discrete OT plan between them. The resulting joined batch is then used in FM. Discrete OT is a convex optimization problem, which has emerged much earlier than its continuous analog. It can be solved effectively, namely, modern algorithms require $O(b^3)$ operations, where b is the batch size.

Drawbacks. The main drawback of OT-CFM is that it recovers only biased dynamic OT solution with non-straight paths. Moreover, this approach was proposed as heuristic without proper theoretical justification. In order to converge to the true transport plan the batch size should be large, while with a growth of batch size computational time increases drastically. In practice, batch sizes that ensure approximation good enough for applications are nearly infeasible to work with.

Rectified Flow (RF)

Another branch of research [6, 5, 22] is based on the straightening property of FM minimization, which was observed as interesting phenomena by the authors of the original paper. The authors of [6] are the first to develop this idea until the fully operational method. They propose an iterative approach to refine the plan π, straightening the trajectories more and more with each iteration. This idea of iterative refinement caused a huge impact in community for its effectiveness. For example, the latest Stable Diffusion models utilize it to decrease inference time.

RF. One can iteratively apply Flow Matching FM to the initial transport plan (e.g., the independent plan), gradually rectifying it. Namely, Rectified Flow Algorithm on K-th iteration has update rule

$$\phi^{K+1} \quad = \quad \text{FM}(\pi^K), \quad \pi^{K+1} = [\text{id}, \phi^{K+1}]\#p_0, \tag{2.7}$$

where ϕ^K, π^K denote flow map and transport plan on K-th iteration, respectively.

The trajectories $\{\{z_t\}_{t\in[0,1]}\}^K$ generated after K iteration of Rectified Flow provably become more and more straight, i.e., error in approximation $z_t^K \approx (1-t)z_0^K + tz_1^K, \forall t \in [0,1]$ decreases with K. The authors also state that for any convex cost function c the flow map ϕ_1^π from Flow Matching yields lower or equal transport cost than initial transport plan π:

$$\int_{\mathbb{R}^D} c(x_0, \phi_1^\pi(x_0))p_0(x_0)dx_0 \leq \int_{\mathbb{R}^D\times\mathbb{R}^D} c(x_0, x_1)\pi(x_0, x_1)dx_0dx_1.$$

Intuitively, the transport costs are guaranteed to decrease because the trajectories of FM as solutions of well-defined ODE do not intersect each other, even if the initial lines connecting x_0 and x_1 can.

c-RF. With each iteration of RF (2.7), transport costs for all convex cost functions do not increase, but, for a given cost function, convergence to its own OT plan (transportation with minimal efforts w.r.t. cost c) is not guaranteed. In [5], the authors address this issue and, for any particular convex cost function c, modify RF to converge to OT map for c. In this modification, called c-Rectified Flow (c-RF), the authors slightly change the FM training objective and restrict the optimization domain only to potential vector fields $u_t(\cdot) = \nabla\bar{c}(\nabla f_t(\cdot))$, where $f_t(\cdot) : \mathbb{R}^D \to \mathbb{R}$ is an arbitrary time-dependent scalar valued function and $\bar{c}$ is the convex conjugate of the cost function c. In order to get dynamic OT solution, the training objective should remain the same, and the vector field u_t should be set as the simple gradient $\nabla f_t(\cdot)$ of the scalar valued function f_t.

PeRFlow. In [22], the authors propose piecewise Rectified Flow (PeRFlow), which divides the flow trajectories into several time windows and conducts reflow in each window. By solving the ODEs in the shortened time interval, PeRFlow avoids simulating the entire ODE trajectory for preparing the training data. This significantly reduces the target synthesis time, enabling the simulation to be performed in real time along with the training procedure.

Drawbacks. In practice, with each iteration RF accumulates error caused by inexactness from previous iterations. This issue is mentioned in [5, §6, point 3]. Due to neural approximations, one can not get exact solution of FM (e.g., $\phi_1^K\#p_0 \neq p_1$), and this inexactness only grows with iterations. In addition, training of (c-)RF becomes non-simulation free after the first iteration, since to calculate the plan $\pi^{K+1} = [\text{id}, \phi^{K+1}]\#p_0$ it has to integrate ODE. This dramatically increases the time of training in comparison with original FM. PeRFlow weakens this effect, however, it does not vanish completely, since the foundation of the methods remains unchanged.

The authors demonstrate in experiments that RF might fail to capture the target distribution. Although, in theory, the number of RF rounds needs to tend to infinity to recover optimal transport, but, in practice, straightness of the paths and obtained vector field cease to change after $2-3$ rounds.

Summary

Flow Matching [4] approach made a huge positive impact in the generative modeling, enabling robust and simple translation of arbitrary distribution to each other. Despite all this, retrieving straight trajectories still remains challenging. The number of papers is dedicated to the modification for solving this drawback. Rectified Flow branch [5, 6, 22] utilizes the theoretically justified straightening effect of FM. However, due to iterative nature in practice, RF requires a lot of computational resources and accumulates error failing to capture the target. Another OT-CFM [7] branch

incorporates typically straight OT vector fields between distributions. The proposed heuristics improve obtained trajectories, but never retrieve fully straight paths in both theory and practice.

The challenge of obtaining straight trajectories remains open and in high demand in application. The existing modification branches have fundamental theoretical drawbacks by design, e.g., constitutive minimizations procedures in RF or biased solution in OT-CFM. Hence, a new solution with novel foundational principles is required.

It is worth noticing that even default Rectified Flow in practice gives good and robust results. However, the progress is moving forward, as well as requirements for generations. The fundamental barriers of current methods can become insuperable after achieving particular accuracy. Thus, having other solutions without such theoretical barriers in the future may only benefit.

Chapter 3
Methodology: New method for generative modeling

In this chapter, I thoroughly present my novel method called Optimal Flow Matching which aims to mend the existing issues of the other Flow Matching-based methods. I provide theoretical justification of the proposed algorithm, details for practical implementation and comparison with the previous approaches.

3.1 Theory: Deriving the Optimization Loss

Optimal vector fields. A vector field u^Ψ is optimal if it generates linear trajectories $\{\{z_t\}_{t\in[0,1]}\}$ such that there exist a convex function $\Psi : \mathbb{R}^D \to \mathbb{R}$, which for any path $\{z_t\}_{t\in[0,1]}$ pushes the initial point z_0 to the final one as $z_1 = \nabla\Psi(z_0)$, i.e.,

$$z_t \;=\; (1-t)z_0 + t\nabla\Psi(z_0), \quad t \in [0,1].$$

The function Ψ defines the ODE

$$dz_t \;=\; (\nabla\Psi(z_0) - z_0)dt, \quad z_t|_{t=0} = z_0. \tag{3.1}$$

Equation (3.1) does not provide a closed formula for u^Ψ as it depends on z_0. The explicit formula is constructed as follows: for a time $t \in [0,1]$ and point x_t, a trajectory $\{z_t\}_{t\in[0,1]}$ s.t.

$$x_t = z_t = (1-t)z_0 + t\nabla\Psi(z_0) \tag{3.2}$$

is used to recover the initial point z_0.

Training objective. Optimal Flow Matching (OFM) approach is as follows: the optimization domain of FM loss with a fixed plan π is restricted only to optimal vector fields. Optimal Flow Matching loss is derived via putting the formula for the vector field u_Ψ into FM loss:

$$\begin{aligned}\mathcal{L}^\pi_{OFM}(\Psi) \;:=\; \mathcal{L}^\pi_{FM}(u^\Psi) &= \int_0^1 \left\{ \int_{\mathbb{R}^D\times\mathbb{R}^D} \|u^\Psi_t(x_t) - (x_1 - x_0)\|^2 \pi(x_0, x_1)dx_0dx_1 \right\} dt, \\ x_t \;&=\; (1-t)x_0 + tx_1.\end{aligned} \tag{3.3}$$

Theorem 3.1 states that OFM solves the dynamic OT via single FM minimization for any initial π.

Theorem 3.1 (OFM and OT connection) *Consider two distributions $p_0, p_1 \in \mathcal{P}_{ac,2}(\mathbb{R}^D)$ and **any** transport plan $\pi \in \Pi(p_0, p_1)$ between them. Then, the dual Optimal Transport loss $\mathcal{L}_{OT}$ and Optimal Flow Matching loss $\mathcal{L}^\pi_{OFM}$ have **the same minimizers**, i.e.,*

$$\arg\min_{\text{convex } \Psi} \mathcal{L}^\pi_{OFM}(\Psi) = \arg\min_{\text{convex } \Psi} \mathcal{L}_{OT}(\Psi).$$

OFM properties

In this subchapter, the OFM's theoretical properties are provided, they give an intuition for understanding of its main working principles and behavior.

Proposition 1 (Simplified OFM Loss) *Loss* (3.3) *can be simplified to a more suitable form:*

$$\mathcal{L}^{\pi}_{OFM}(\Psi)=\int_0^1 \left\{ \int_{\mathbb{R}^D\times\mathbb{R}^D} \left\| \frac{(\phi_t^{\Psi})^{-1}(x_t)-x_0}{t} \right\|^2 \pi(x_0,x_1)dx_0dx_1 \right\} dt, x_t=(1-t)x_0+tx_1. \quad (3.4)$$

The simplified form (3.4) shows that OFM loss actually measures how well Ψ restores initial points x_0 of linear interpolations depending on future point x_t and time t.

Generative properties of OFM. In OFM, the main goal is to construct a vector field u which is as close to the dynamic OT field u^* as possible. One can use the least square regression to measure the distance between them:

$$\text{dist}(u,u^*) := \int_0^1 \int_{\mathbb{R}^D} \|u_t(x_t)-u_t^*(x_t)\|^2 \underbrace{\phi_t^*\#p_0(x_t)}_{:=p_t^*(x_t)} dx_t dt. \quad (3.5)$$

Proposition 2 (Intractable Distance) *The distance* $\text{dist}(u,u^*)$ *between an arbitrary vector field* u *and OT field* u^* *equals to the FM loss from* (2.7) *with the optimal plan* π^**, i.e.,*

$$\text{dist}(u,u^*) = \mathcal{L}^{\pi^*}_{FM}(u) - \underbrace{\mathcal{L}^{\pi^*}_{FM}(u^*)}_{=0}.$$

The intractable $\text{dist}(u,u^*)$ can not be minimized since the optimal plan π^* is unknown. In OT-CFM [9], authors heuristically approximate π^* in $\mathcal{L}^{\pi^*}_{FM}(u)$, but obtain biased solution. Surprisingly, for the *optimal* vector fields, the distance can be calculated explicitly via **any** known plan π.

Proposition 3 (Tractable Distance For OFM) *The distance* $\text{dist}(u^{\Psi},u^{\Psi^*})$ *between an* ***optimal*** *vector field* u^{Ψ} *generated by a convex function* Ψ *and the vector field* u^{Ψ^*} *with the Brenier potential* Ψ^* *can be evaluated directly via OFM loss* (3.3) *and* ***any*** *plan* π*:*

$$\text{dist}(u^{\Psi},u^{\Psi^*}) = \mathcal{L}^{\pi}_{FM}(u^{\Psi}) - \mathcal{L}^{\pi}_{FM}(u^{\Psi^*}) = \mathcal{L}^{\pi}_{OFM}(\Psi) - \mathcal{L}^{\pi}_{OFM}(\Psi^*). \quad (3.6)$$

In (3.6), the first term is our tractable OFM loss, and the second term does not depend on Ψ. Hence, during the whole minimization process in OFM, the distance (3.5) between the current vector field and the dynamic OT field is gradually lowered up to the complete match.

3.2 Practical implementation aspects

In this subsection, the details of optimization of Optimal Flow Matching loss (3.3) are explained.
Parametrization of Ψ. In practice, the class of convex functions is parametrized with Input Convex Neural Networks (ICNNs) [10] Ψ_θ and parameters θ. These are scalar-valued neural networks built in such a way that the network is convex in its input. They consist of fully-connected or convolution blocks, some weights of which are set to be non-negative in order to keep the convexity. In addition, activation functions are considered to be only non-decreasing and convex in each input coordinate. These networks are able to support most of the popular training techniques (e.g., gradient descent optimization, dropout, skip connection, etc.).
OFM loss calculation. There exists an explicit formula for gradient of OFM loss (3.3) suitable for modern deep learning frameworks.

Theorem 3.2 (Explicit Loss Gradient Formula) *The gradient of $\mathcal{L}^{\pi}_{OFM}$ can be calculated as*

$$\frac{d\mathcal{L}^{\pi}_{OFM}}{d\theta} := \frac{d}{d\theta}\mathbb{E}_{t;x_0,x_1\sim\pi}\left\langle \text{NO-GRAD}\left\{2\left(t\nabla^2\Psi_\theta(z_0)+(1-t)I\right)^{-1}\frac{(x_0-z_0)}{t}\right\}, \nabla\Psi_\theta(z_0)\right\rangle,$$

where variables under NO-GRAD *and z_0 remain constants during differentiation.*

Flow map inversion. The procedure of finding the initial point z_0 can be reduced to the solution of the following minimization problem:

$$\begin{aligned} x_t &= (1-t)z_0 + t\nabla\Psi(z_0), \\ 0 &= \nabla\left(\frac{(1-t)}{2}\|\cdot\|^2 + t\Psi(\cdot) - \langle x_t, \cdot\rangle\right)(z_0), \\ z_0 &= \arg\min_{z_0\in\mathbb{R}^D}\left[\frac{(1-t)}{2}\|z_0\|^2 + t\Psi(z_0) - \langle x_t, z_0\rangle\right]. \end{aligned} \tag{3.7}$$

Optimization subproblem (3.7) is at least $(1-t)$-**strongly convex** and can be effectively solved for any given point x_t (in comparison with typical non-convex optimization tasks).

Algorithm. The Optimal Flow Matching pseudocode is presented in listing 1. The math expectation over plan π and time t with uniform distribution on $[0,1]$ is estimated via Monte Carlo.

Algorithm 1 Optimal Flow Matching

Input: Initial transport plan $\pi \in \Pi(p_0,p_1)$, number of iterations K, batch size B, optimizer Opt, sub-problem optimizer $SubOpt$, ICNN Ψ_θ

1: **for** $k = 0,\dots,K-1$ **do**
2: Sample batch $\{(x_0^i, x_1^i)\}_{i=1}^B$ of size B from plan π;
3: Sample times batch $\{t^i\}_{i=1}^B$ of size B from $U[0,1]$;
4: Calculate linear interpolation $x_{t^i}^i = (1-t^i)x_0^i + t^i x_1^i$ for all $i \in \overline{1,B}$;
5: Find the initial points z_0^i via solving the convex problem with $SubOpt$:

$$z_0^i = \text{NO-GRAD}\left\{\arg\min_{z_0^i}\left[\frac{(1-t^i)}{2}\|z_0^i\|^2 + t^i\Psi_\theta(z_0^i) - \langle x_{t^i}^i, z_0^i\rangle\right]\right\};$$

6: Calculate loss $\hat{\mathcal{L}}_{OFM}$

$$\hat{\mathcal{L}}_{OFM} = \frac{1}{B}\sum_{i=1}^{B}\left\langle \text{NO-GRAD}\left\{2\left(t^i\nabla^2\Psi_\theta(z_0^i)+(1-t^i)I\right)^{-1}\frac{(x_0^i-z_0^i)}{t^i}\right\}, \nabla\Psi_\theta(z_0^i)\right\rangle;$$

7: Update parameters θ via optimizer Opt step with $\frac{d\hat{\mathcal{L}}_{OFM}}{d\theta}$;
8: **end for**

Chapter 4
Numerical experiments

4.1 Experimental Illustrations

In this chapter, I demonstrate the performance of Optimal Flow Matching on illustrative 2D scenario (§4.1) and Wasserstein-2 benchmark [21] (§4.1). Finally, OFM is applied for solving high-dimensional unpaired image-to-image translation in the latent space of pretrained ALAE autoencoder (§4.1). The code of our OFM implementation and the conducted experiments is available at `https://github.com/Jhomanik/Optimal-Flow-Matching`.

Illustrative 2D Example

In this subchapter, I illustrate the proof-of-concept of our OFM on 2D setup and demonstrate that OFM's solutions do not depend on the initial transport plan π. The algorithm 1 is run between a standard Gaussian $p_0 = \mathcal{N}(0, I)$ and a Mixture of eight Gaussians p_1 depicted in the Figure 4.2a. The three different stochastic plans π are considered: independent plan $p_0 \times p_1$ (Figure 4.2b), minibatch and *anti*minibatch (Figures 4.2c, 4.2d) discrete OT (quadratic cost) with batch size $B_{\text{mb}} = 64$. In the **anti**minibatch case, I compose the pairs of source and target points by solving discrete OT with **minus** quadratic cost $-\|x - y\|_2^2$. The fitted OFM maps and trajectories are presented in Figure 4.2. One can empirically see that OFM finds the *same solution for all initial plans* π.

These plans are also applied to the original FM (2.7) in Figure 4.1. In comparison with OFM, the resulting paths obtained by FM considerably depend on the plan.

High-dimensional OT Benchmarks

In this subchapter, OFM and other methods are quantitatively compared via testing their ability to solve OT. OFM, FM based methods and OT solvers are run on OT Benchmark [21]. The authors provide high-dimensional continuous distributions p_0, p_1 for which the ground truth OT map T^* for the quadratic cost is known by the construction.

Metrics. Following the authors of the benchmark [21], to assess the quality of retrieved transport map T between p_0 and p_1, we use *unexplained variance percentage* (UVP): $\mathcal{L}^2\text{-UVP}(T) := 100 \cdot \|T - T^*\|^2_{\mathcal{L}^2(p_0)} / \text{Var}(p_1)\%$. For values $\mathcal{L}^2\text{-UVP}(T) \approx 0\%$, T approximates T^*, while for values $\geq 100\%$ T is far from optimal. We also calculate the *cosine similarity* between ground truth

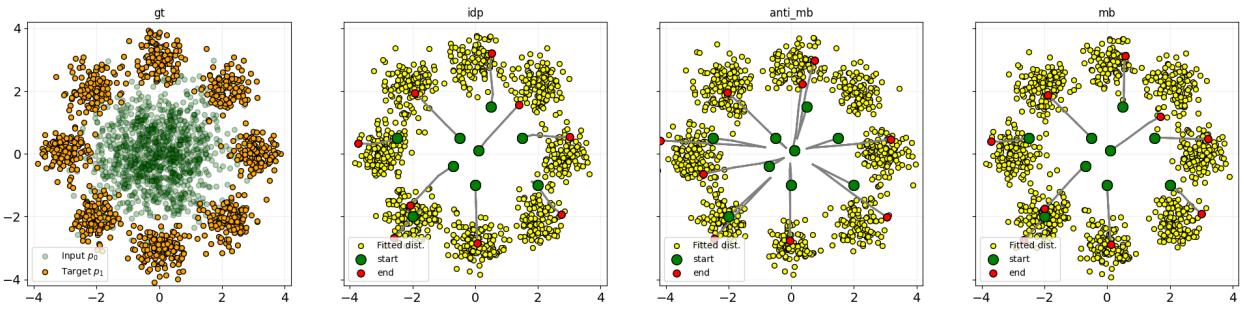
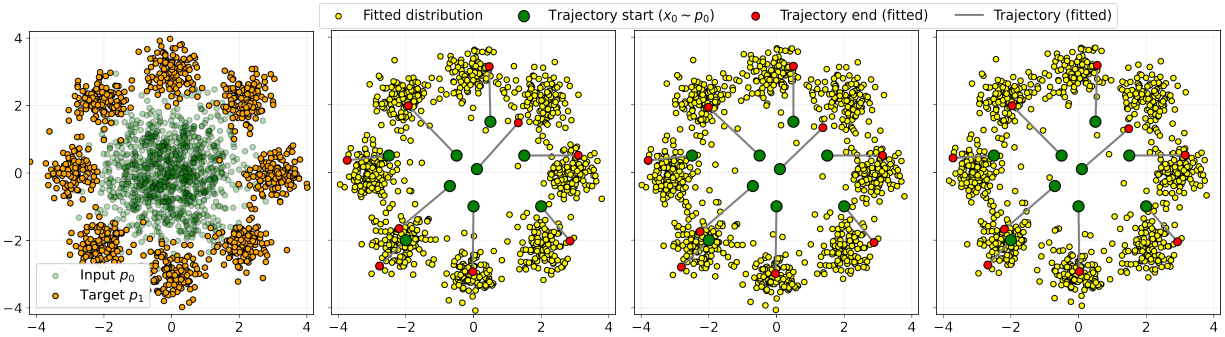


Figure 4.1: Performance of Flow Matching on *Gaussian→Eight Gaussians* 2D setup.



(a) Input and target distributions p_0 and p_1.

(b) Fitted OFM; independent $\pi = p_0 \times p_1$.

(c) Fitted OFM; minibatch π.

(d) Fitted OFM; **anti**minibatch π.

Figure 4.2: Performance of Optimal Flow Matching on *Gaussian→Eight Gaussians* 2D setup.

directions $T^* - \mathrm{id}$ and obtained directions $T - \mathrm{id}$, i.e.,

$$\cos(T - \mathrm{id}, T^* - \mathrm{id}) = \frac{\langle T - \mathrm{id}, T^* - \mathrm{id} \rangle_{\mathcal{L}^2(p_0)}}{\|T - \mathrm{id}\|_{\mathcal{L}^2(p_0)} \cdot \|T^* - \mathrm{id}\|_{\mathcal{L}^2(p_0)}} \in [-1, 1].$$

For good approximations the cosine metric is approaching 1. We estimate $\mathcal{L}^2-$UVP and cos metrics with 2^{14} samples from p_0.

Competitors. In experiments, Conditional Flow Matching (OT-CFM), Rectified Flow (RF), c-Rectified Flow (c-RF), the most relevant OT solver MMv-1 [13] and its amortized version from [20] are compared. In [13] and [20], the authors directly minimize the dual formulation loss $\mathcal{L}_{OT}$ (2.5) by parametrizing Ψ with ICNNs and calculating $\overline{\Psi}(x_1)$ via solving a convex optimization subproblem. The latter is similar to our inversion (3.7). Additionally, in [20], the authors use MLPs to parametrize Ψ, and these results are included as well. Following [21], I also provide results for a linear OT map (baseline) which translates means and variances of distributions to each other. For OFM, two initial plans are considered: independent plan (Ind) and minibatch OT (MB), the batch size for the latter is $B_{\text{mb}} = 64$.

Results. The overall results are presented in Table 4.1.

Solver	Solver type	$D=2$	$D=4$	$D=8$	$D=16$	$D=32$	$D=64$	$D=128$	$D=256$
MMv1* [13]		0.2	1.0	1.8	1.4	6.9	8.1	2.2	2.6
Amortization, ICNN** [20]	Dual OT solver	0.26	0.78	1.6	1.1	1.9	4.2	1.6	2.0
Amortization, MLP** [20]		0.03	0.22	0.6	0.8	2.0	2.1	0.67	0.59
Linear* [21]	Baseline	14.1	14.9	27.3	41.6	55.3	63.9	63.6	67.4
OT-CFM [9]		0.16	0.73	2.27	4.33	7.9	11.4	12.1	27.5
RF [6]		8.58	49.46	51.25	63.33	63.52	85.13	84.49	83.13
c-RF [5]	Flow Matching	1.56	13.11	17.87	35.39	48.46	66.52	68.08	76.48
OFM Ind		0.19	0.61	1.4	1.1	1.47	8.35	1.96	3.96
OFM MB		**0.15**	**0.52**	**1.2**	**1.0**	**1.2**	**7.2**	**1.5**	**2.9**

Table 4.1: $\mathcal{L}^2-$UVP values of solvers fitted on high-dimensional benchmarks in dimensions $D = 2, 4, 8, 16, 32, 64, 128, 256$. The best metric over *Flow Matching based* methods is **bolded**. * Metrics are taken from [21]. ** Metrics are taken from [20].

Solvers' results for cos metric are presented in Table 4.2.

Solver	Solver type	$D=2$	$D=4$	$D=8$	$D=16$	$D=32$	$D=64$	$D=128$	$D=256$
MMv1* [13]	Dual OT solver	0.99	0.99	0.99	0.99	0.98	0.99	0.99	0.99
Linear*	Baseline	0.75	0.80	0.73	0.73	0.76	0.75	0.77	0.77
OT-CFM MB [9]		0.999	0.985	0.978	0.968	0.975	0.96	0.949	0.915
RF [6]		0.87	0.75	0.65	0.67	0.72	0.70	0.70	0.70
c-RF [5]	Flow Matching	0.989	0.83	0.83	0.78	0.778	0.762	0.748	0.73
OFM Ind		0.999	0.993	0.993	0.993	0.999	0.966	0.992	0.981
OFM MB		**0.999**	**0.994**	**0.995**	**0.994**	**0.999**	**0.970**	**0.994**	**0.986**

Table 4.2: cos values of solvers fitted on high-dimensional benchmarks in dimensions $D = 2, 4, 8, 16, 32, 64, 128, 256$. The best metric over *Flow Matching based* solvers is **bolded**. * Metrics for MMv1 and linear baseline are taken from [21].

Discussion. Among FM-based methods, OFM with any plan demonstrates the best results. For all plans, OFM convergences to close final solutions and metrics. Minibatch plan provides a little bit better results, especially in high dimensions. In theory, the OFM results for any plan π must be similar. However, in stochastic optimization, plans with large variance yield convergence to slightly worse solutions.

MLP-based OT solver usually beats our OFM, since MLPs do not have ICNNs' limitations in practice. However, usage of MLP is an empirical trick and is not completely justified. ONe can run OFM with MLP instead ICNN, and, unfortunately, the method fails to converge.

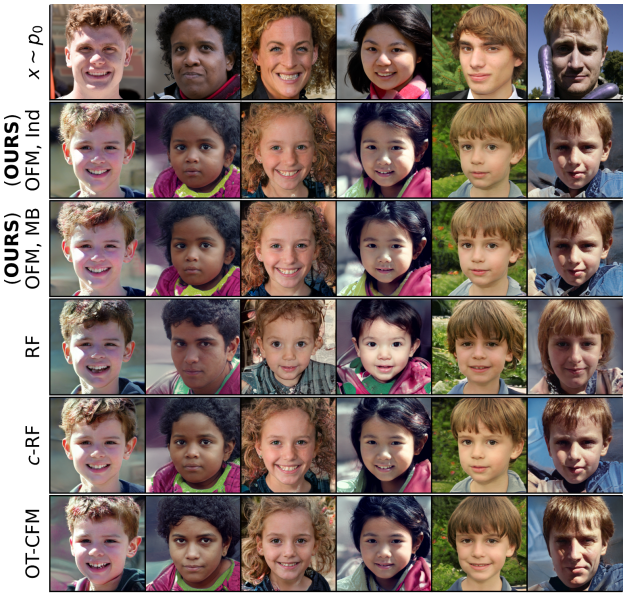


Figure 4.3: Unpaired I2I *Adult*→*Child* by FM solvers, ALAE 1024×1024 FFHQ latent space.

RF demonstrates worse performance than even linear baseline, but it is expected since it is not designed to solve $\mathbb{W}_2$ OT. In turn, c-RF works better, but rapidly deteriorates with increasing dimensions. OT-CFM demonstrates the best results among baseline FM-based methods, but still underperforms compared to our OFM solver in high dimensions.

Unpaired Image-to-image Transfer

Another task that involves learning a translation between two distributions is unpaired image-to-image translation [24]. The setup is taken from [25] where translation is computed in the 512 dimensional latent space of the pre-trained ALAE autoencoder [26] on 1024×1024 FFHQ dataset [27]. In particular, the train FFHQ samples (60K faces) are divided into $children$ and $adults$ subsets and the corresponding ALAE latent codes are considered as the source and target distributions p_0 and p_1. At the inference stage, a new (unseen) $adult$ face from a test FFHQ sample is taken, its latent code is extracted, processed with learned model and then decoded back to the image space. The qualitative translation results and FID metric [28] are presented in Figure 4.3 and Table 4.3, respectively.

Method	OFM, Ind Fitted	OFM, MB Fitted	RF	$c-$RF	OT-CFM
FID	11.8	**11.0**	21.0	13.5	12.9

Table 4.3: FID metric on Adult→Child translation task for the Flow Matching based methods.

The batch size for minibatch OT methods ($\lfloor$OFM, MB$\rceil$, $\lfloor$OT-CFM$\rceil$) is $B_{\text{mb}} = 128$. OFM converges to nearly the same solution for both independent and MB plans, and demonstrates qualitatively plausible translations. The most similar results to OFM are demonstrated by $\lfloor c$-RF$\rceil$. Similar to OFM, this method (in the limit of RF steps) also recovers the quadratic OT mapping.

Unpaired Image-to-image transfer details To conduct the experiments with high-dimensional I2I translation empowered with pretrained ALAE autoencoder, the publicly available code was adopted:

https://github.com/SKholkin/LightSB-Matching.

Additional qualitative results for our method are provided in Figure 4.4.

Figure 4.4: Unpaired I2I *Adult*→*Child* by **our** OFM solver, ALAE 1024×1024 FFHQ latent space. The samples are uncurated.

Computation time

In what follows, approximate running times for training OFM and other FM-based method in different experiments.

In the Illustrative 2D experiment, the training takes ≈ 1.5 hours on a single 1080 ti GPU. In the Wasserstein-2 benchmark, the computation time depends on the dimensionality $D = 2, 4, \dots, 256$. Totally, all the benchmark experiments (both with Ind and MB plan π) take ≈ 3 days on three A100 GPUs. In the ALAE experiment, the training stage lasts for ≈ 5 hours on a single 1080 ti GPU.

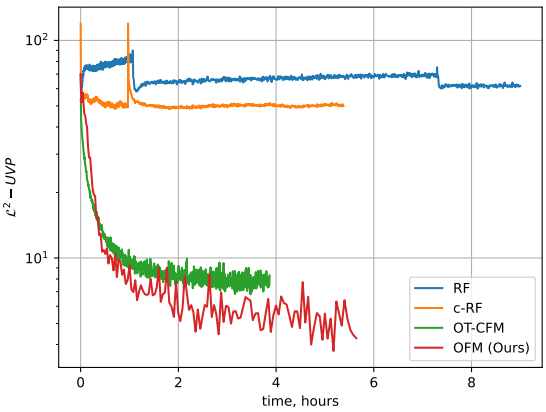


Figure 4.5: $\mathcal{L}^2-$UVP metric depending on the elapsed training time in dimension $D = 32$.

For better understanding of methods' behaviour over time, I depict achieved $\mathcal{L}^2-$UVP metric on the benchmark ($D = 32$) depending on elapsed training time in Figure 4.5. The training iteration of OFM is computationally expensive, but it requires less steps to achieve the best results.

Amortization technique

In order to train OFM, one needs to efficiently solve subproblem (3.7). As an example of more advanced technique rather than LBFGS solver, the amortization trick proposed in [20] is discussed.

Namely, one can find an approximate solution of (3.7) at point x_t and time t with an extra MLP $A_\phi(\cdot, \cdot) : \mathbb{R}^D \times [0, 1] \to \mathbb{R}^D$:

$$A_\phi(x_t, t) \approx \arg\min_{z_0 \in \mathbb{R}^D} \left[\frac{(1-t)}{2} \|z_0\|^2 + t\Psi(z_0) - \langle x_t, z_0 \rangle \right], \tag{4.1}$$

and then run sub-problem solver (LBFGS) initialized with $A_\phi(x_t, t)$ until convergence. I modify the training pipeline and include learning of parameters ϕ of A_ϕ in Algorithm 2.

Algorithm 2 Optimal Flow Matching with Amortization

Input: Initial transport plan $\pi \in \Pi(p_0, p_1)$, number of iterations K, batch size B, optimizer Opt, amortization optimizer $AmorOpt$, sub-problem optimizer $SubOpt$, ICNN Ψ_θ, MLP A_ϕ

1: **for** $k = 0, \dots, K-1$ **do**
2: Sample batch $\{(x_0^i, x_1^i)\}_{i=1}^B$ of size B from plan π;
3: Sample times batch $\{t^i\}_{i=1}^B$ of size B from $U[0, 1]$;
4: Calculate linear interpolation $x_{t^i}^i = (1 - t^i)x_0^i + t^i x_1^i$ for all $i \in \overline{1, B}$;
5: Compute initialization $z_{init}^i = A_\phi(x_{t^i}^i, t^i)$ for all $i \in \overline{1, B}$;
6: Find detached solution z_0^i of (3.7) via $SubOpt$ initialized with z_{init}^i for all $i \in \overline{1, B}$;
7: Calculate OFM loss $\hat{\mathcal{L}}_{OFM}$

$$\hat{\mathcal{L}}_{OFM} = \frac{1}{B}\sum_{i=1}^{B}\left\langle \text{NO-GRAD}\left\{2\left(t^i\nabla^2\Psi_\theta(z_0^i) + (1-t^i)I\right)^{-1}\frac{(x_0^i - z_0^i)}{t^i}\right\}, \nabla\Psi_\theta(z_0^i)\right\rangle;$$

8: Update parameters θ via optimizer Opt step with $\frac{d\hat{\mathcal{L}}_{OFM}}{d\theta}$;
9: Calculate Amortization loss $\mathcal{L}_{Amor}$

$$\mathcal{L}_{Amor} = \frac{1}{B}\sum_{i=1}^{B}\|z_{init}^i - z_0^i\|^2;$$

10: Update parameters ϕ via optimizer $AmorOpt$ step with $\frac{d\mathcal{L}_{Amor}}{d\phi}$;
11: **end for**

During the experiments, no improvements of the final metrics were observed, in comparison with the original OFM with the same hyperparameters. However, this augmentation potentially can cause a shrinking of the overall training time. During training, $A_\phi(\cdot, \cdot)$ learns to predict more and more accurate initial solution z_{init}^i and, thus, reduces the required number of the expensive $SubOpt$ steps.

Chapter 5
Discussion and conclusion

5.1 Relation to Prior Works

In this subchapter, Optimal Flow Matching and previous straightening approaches are compared. One unique feature of OFM is that it works only with flows which have straight paths by design and does not require ODE integration to transport points. Other methods may result in non-straight paths during training, and they still have to solve ODE even with near-straight paths.

OT Solvers [13, 14, 20]. According to Theorem 3.1, OFM and dual OT solvers basically minimize the same OT loss. However, OFM actively utilizes the temporal component of the dynamic process. It paves a novel theoretical bridge between OT and FM. Such a direct connection can lead to the adoption of the strengths of both methods and a deeper understanding of them.

OT-CFM branch [7]. Unlike OFM approach, OT-CFM method retrieves biased OT solution, and the recovery of straight paths is not guaranteed. In OT-CFM, minibatch OT plan appears as a heuristic that helps to get better trajectories in practice. In contrast, usage of any initial transport plan π in OFM is completely justified in Theorem 3.1.

Rectified Flow branch [6, 5, 22]. In Rectified Flow [6], the authors iteratively apply Flow Matching to refine the obtained trajectories. However, in each iteration, RF accumulates error since one may not learn the exact flow due to neural approximations. In addition, RF does not guarantee convergence to the OT plan for the quadratic cost. The c-Rectified Flow [5] modification can converge to the OT plan for any cost function c, but still remains iterative. In addition, RF and c-RF both requires ODE simulation after the first iteration to continue training. In OFM, only the quadratic cost function is considered, but the method retrieves its OT solution in just one FM iteration without simulation of the trajectories.

5.2 OFM Limitations

Despite all advantages, Optimal Flow Matching has the following limitations:

(a) Flow map inversion. During training, one needs to solve strongly convex subproblem (3.7) to compute initial z_0. In practice, it can be approached by the standard gradient descent (with LBFGS optimizer), but actually there exist many improved methods to solve such conjugation problems more effectively in both the optimization [29, 30] and OT [20, 14]. This provides a dozen of opportunities for improvement.

(b) ICNNs. It is known that ICNNs may underperform compared to regular neural networks [21, 31]. Thus, ICNN parametrization may limit the performance of our OFM. However, improvements of ICNNs are actively being studied [32, 33, 34, 35] due to their growing popularity in various tasks [36, 37, 38].

(c) Hessian inversion. The gradient of OFM loss is calculated by the means of Theorem 3.2. In the explicit formula there, expensive inversion of the hessian $\nabla^2\Psi(\cdot)$ is required.

5.3 Summary

Optimal Flow Matching is a novel modification of Flow Matching that after a single FM iteration obtains straight trajectories which can be simulated without ODE solving. Moreover, it recovers OT flow for the quadratic cost function for any initial plan. The main idea of our OFM is to consider during FM only specific vector fields which are the gradients of convex functions and yield straight paths by design. OFM does not have fundamental limitations like theoretically unjustified biased solution of OT-CFM or error accumulation of RF. However, it has its own issues which are related rather to optimization procedure than design of the method itself.

I believe that our novel theoretical results have a huge potential for improving modern flow matching-based methods and inspiring the community for further studies. I think this is of high importance especially taking into account that modern generative models start to extensively use flow matching methods [22, 1, 2].

Bibliography

[1] X. Liu, X. Zhang, J. Ma, J. Peng, and qiang liu, "Instaflow: One step is enough for high-quality diffusion-based text-to-image generation," in *The Twelfth International Conference on Learning Representations*, 2024.

[2] P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel, D. Podell, T. Dockhorn, Z. English, and R. Rombach, "Scaling rectified flow transformers for high-resolution image synthesis," in *Forty-first International Conference on Machine Learning*, 2024.

[3] H. Cao, C. Tan, Z. Gao, Y. Xu, G. Chen, P.-A. Heng, and S. Z. Li, "A survey on generative diffusion models," *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[4] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow matching for generative modeling," in *The Eleventh International Conference on Learning Representations*, 2023.

[5] Q. Liu, "Rectified flow: A marginal preserving approach to optimal transport," *arXiv preprint arXiv:2209.14577*, 2022.

[6] X. Liu, C. 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.

[7] A.-A. Pooladian, H. Ben-Hamu, C. Domingo-Enrich, B. Amos, Y. Lipman, and R. T. Chen, "Multisample flow matching: Straightening flows with minibatch couplings," in *International Conference on Machine Learning*, pp. 28100–28127, PMLR, 2023.

[8] C. Villani, *Topics in optimal transportation*, vol. 58. American Mathematical Soc., 2021.

[9] A. Tong, K. FATRAS, N. Malkin, G. Huguet, Y. Zhang, J. Rector-Brooks, G. Wolf, and Y. Bengio, "Improving and generalizing flow-based generative models with minibatch optimal transport," *Transactions on Machine Learning Research*, 2024. Expert Certification.

[10] B. Amos, L. Xu, and J. Z. Kolter, "Input convex neural networks," in *International Conference on Machine Learning*, pp. 146–155, PMLR, 2017.

[11] J.-D. Benamou and Y. Brenier, "A computational fluid mechanics solution to the monge-kantorovich mass transfer problem," *Numerische Mathematik*, vol. 84, no. 3, pp. 375–393, 2000.

[12] A. Genevay, M. Cuturi, G. Peyré, and F. Bach, "Stochastic optimization for large-scale optimal transport," *Advances in neural information processing systems*, vol. 29, 2016.

[13] A. Taghvaei and A. Jalali, "2-wasserstein approximation via restricted convex potentials with application to improved training for gans," *arXiv preprint arXiv:1902.07197*, 2019.

[14] A. Makkuva, A. Taghvaei, S. Oh, and J. Lee, "Optimal transport mapping via input convex neural networks," in *International Conference on Machine Learning*, pp. 6672–6681, PMLR, 2020.

[15] M. Daniels, T. Maunu, and P. Hand, "Score-based generative neural networks for large-scale optimal transport," *Advances in neural information processing systems*, vol. 34, pp. 12955–12965, 2021.

[16] V. De Bortoli, J. Thornton, J. Heng, and A. Doucet, "Diffusion schrödinger bridge with applications to score-based generative modeling," *Advances in Neural Information Processing Systems*, vol. 34, pp. 17695–17709, 2021.

[17] A. Korotin, L. Li, J. Solomon, and E. Burnaev, "Continuous wasserstein-2 barycenter estimation without minimax optimization," in *International Conference on Learning Representations*, 2021.

[18] J. Fan, S. Liu, S. Ma, H.-M. Zhou, and Y. Chen, "Neural monge map estimation and its applications," *Transactions on Machine Learning Research*, 2023. Featured Certification.

[19] T. Uscidda and M. Cuturi, "The monge gap: A regularizer to learn all transport maps," in *International Conference on Machine Learning*, pp. 34709–34733, PMLR, 2023.

[20] B. Amos, "On amortizing convex conjugates for optimal transport," in *The Eleventh International Conference on Learning Representations*, 2023.

[21] A. Korotin, L. Li, A. Genevay, J. M. Solomon, A. Filippov, and E. Burnaev, "Do neural optimal transport solvers work? a continuous wasserstein-2 benchmark," *Advances in neural information processing systems*, vol. 34, pp. 14593–14605, 2021.

[22] H. Yan, X. Liu, J. Pan, J. H. Liew, Q. Liu, and J. Feng, "Perflow: Piecewise rectified flow as universal plug-and-play accelerator," 2024.

[23] K. Neklyudov, R. Brekelmans, D. Severo, and A. Makhzani, "Action matching: Learning stochastic dynamics from samples," in *International conference on machine learning*, pp. 25858–25889, PMLR, 2023.

[24] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017.

[25] A. Korotin, N. Gushchin, and E. Burnaev, "Light schrödinger bridge," in *The Twelfth International Conference on Learning Representations*, 2023.

[26] S. Pidhorskyi, D. A. Adjeroh, and G. Doretto, "Adversarial latent autoencoders," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14104–14113, 2020.

[27] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.

[28] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in neural information processing systems*, vol. 30, 2017.

[29] B. Van Scoy, R. A. Freeman, and K. M. Lynch, "The fastest known globally convergent first-order method for minimizing strongly convex functions," *IEEE Control Systems Letters*, vol. 2, no. 1, pp. 49–54, 2017.

[30] J. Hiriart-Urruty and Y. Lucet, "Parametric computation of the legendre-fenchel conjugate with application to the computation of the moreau envelope," *Journal of Convex Analysis*, vol. 14, no. 3, p. 657, 2007.

[31] A. Korotin, V. Egiazarian, L. Li, and E. Burnaev, "Wasserstein iterative networks for barycenter estimation," *Advances in Neural Information Processing Systems*, vol. 35, pp. 15672–15686, 2022.

[32] S. Chaudhari, S. Pranav, and J. M. Moura, "Gradient networks," *arXiv preprint arXiv:2404.07361*, 2024.

[33] C. Bunne, A. Krause, and M. Cuturi, "Supervised training of conditional monge maps," *Advances in Neural Information Processing Systems*, vol. 35, pp. 6859–6872, 2022.

[34] J. Richter-Powell, J. Lorraine, and B. Amos, "Input convex gradient networks," *arXiv preprint arXiv:2111.12187*, 2021.

[35] P.-J. Hoedt and G. Klambauer, "Principled weight initialisation for input-convex neural networks," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[36] S. Yang and B. W. Bequette, "Optimization-based control using input convex neural networks," *Computers & Chemical Engineering*, vol. 144, p. 107143, 2021.

[37] M. Ławryńczuk, "Input convex neural networks in nonlinear predictive control: A multimodel approach," *Neurocomputing*, vol. 513, pp. 273–293, 2022.

[38] Y. Chen, Y. Shi, and B. Zhang, "Optimal control via neural networks: A convex approach," *arXiv preprint arXiv:1805.11835*, 2018.

[39] R. T. Rockafellar, "Convex analysis," 2015.

[40] E. Polovinkin and M. Balashov, "Elements of convex and strongly convex analysis," 2007.

[41] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[42] A. Korotin, V. Egiazarian, A. Asadulaev, A. Safin, and E. Burnaev, "Wasserstein-2 generative networks," in *International Conference on Learning Representations*, 2021.

[43] C.-W. Huang, R. T. Q. Chen, C. Tsirigotis, and A. Courville, "Convex potential flows: Universal probability distributions with optimal transport and convex optimization," in *International Conference on Learning Representations*, 2021.

[44] D. Morales-Brotons, T. Vogels, and H. Hendrikx, "Exponential moving average of weights in deep learning: Dynamics and benefits," *Transactions on Machine Learning Research*, 2024.

[45] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[46] J. R. Dormand and P. J. Prince, "A family of embedded runge-kutta formulae," *Journal of computational and applied mathematics*, vol. 6, no. 1, pp. 19–26, 1980.

Appendix

5.4 Proofs and auxiliary statements

In this section, I place the proofs of all our results from the main manuscript and some auxiliary results. The proofs of Prop. 3.2 and Prop. 1 are reordered, since the former is based on the latter.

Note that in all of our theoretical derivations, if not stated explicitly, the differentiability of convex potential Ψ at given points z_0, x_t is assumed. This assumption is done for simplicity and does not spoil our theory. The convex functions are known to be differentiable almost surely w.r.t Lebesgue measure [39]. Therefore, since the absolutely continuous reference distributions p_0, p_1 are considered (see §2), the differentiability of Ψ at the considered points also holds almost surely.

Proof of Proposition 1 (Simplified OFM Loss)

Proof: By definition $\mathcal{L}^{\pi}_{OFM}(\Psi)$ equals to

$$\mathcal{L}^{\pi}_{OFM}(\Psi) := \int_0^1 \left\{ \int_{\mathbb{R}^D \times \mathbb{R}^D} \|u^{\Psi}_t(x_t) - (x_1 - x_0)\|^2 \pi(x_0, x_1) dx_0 dx_1 \right\} dt, x_t = (1-t)x_0 + t x_1. \quad (5.1)$$

For fixed points x_0, x_1 and time t in integrand, I find a point $z_0 = (\phi^{\Psi}_t)^{-1}(x_t)$ such that in moment $t \in [0,1]$ it is transported to point $x_t = (1-t)x_0 + t x_1$. This point z_0 satisfies equality

$$x_t = t\nabla\Psi(z_0) + (1-t)z_0.$$

The vector field u^{Ψ}_t is defined as

$$u^{\Psi}_t(x_t) \;=\; \nabla\Psi(z_0) - z_0 = \frac{x_t - z_0}{t}.$$

Putting $u^{\Psi}_t(x_t)$ in the integrand of (5.1), I obtain simplified integrand

$$\begin{aligned} \|x_1 - x_0 - u^{\Psi}_t(x_t)\|^2 &= \left\| x_1 - x_0 - \left(\frac{x_t - z_0}{t}\right) \right\|^2 \\ &= \frac{1}{t^2}\|t x_1 - t x_0 - ((1-t)x_0 + t x_1) + z_0\|^2 \\ &= \frac{1}{t^2}\|z_0 - x_0\|^2 = \left\| \frac{(\phi^{\Psi}_t)^{-1}(x_t) - x_0}{t} \right\|^2. \end{aligned}$$

□

Proof of Proposition 3.2 (Explicit Loss Gradient Formula)

Proof: Point $z_0 = (\phi^{\Psi_\theta}_t)^{-1}(x_t)$ now depends on parameters θ. I differentiate the integrand from the simplified OFM loss (3.4) for fixed points x_0, x_1 and time t, i.e.,

$$d\left(\frac{1}{t^2}\|z_0 - x_0\|^2\right) \;=\; 2\left\langle \frac{z_0 - x_0}{t^2}, \frac{dz_0}{d\theta} d\theta \right\rangle. \quad (5.2)$$

For point z_0, the equation (5.7) holds true:

$$x_t = (1-t)z_0 + t\nabla\Psi_\theta(z_0). \tag{5.3}$$

I differentiate (5.3) w.r.t. θ and obtain

$$\begin{aligned} 0 &= (1-t)\frac{dz_0}{d\theta} + t\nabla^2\Psi_\theta(z_0)\frac{dz_0}{d\theta} + t\frac{\partial\nabla\Psi_\theta}{\partial\theta}(z_0) \Rightarrow \\ \frac{dz_0}{d\theta} &= -\left(t\nabla^2\Psi_\theta(z_0) + (1-t)I\right)^{-1}\cdot t\frac{\partial\nabla\Psi_\theta}{\partial\theta}(z_0). \end{aligned}$$

Therefore, one have

$$\begin{aligned} (5.2) &= \left\langle 2\frac{x_0 - z_0}{t}, \left(t\nabla^2\Psi_\theta(z_0) + (1-t)I\right)^{-1}\frac{\partial\nabla\Psi_\theta}{\partial\theta}(z_0)d\theta \right\rangle \\ &= \left\langle 2\left(t\nabla^2\Psi_\theta(z_0) + (1-t)I\right)^{-1}\frac{(x_0 - z_0)}{t}, \frac{\partial\nabla\Psi_\theta}{\partial\theta}(z_0)d\theta \right\rangle. \end{aligned} \tag{5.4}$$

Now the differentiation over θ is located only in the right part of (5.4) in the term $\frac{\partial\nabla\Psi_\theta}{\partial\theta}$. Hence, point z_0 and the left part of (5.4) can be considered as constants during differentiation. To get the gradient of OFM loss I also need to take math expectation over plan π and time t. □

The following two Lemmas are used to prove the main theoretical result, Theorem 3.1.

Lemma 1 (Properties of convex functions and their conjugates) *Let $\Psi : \mathbb{R}^D \to \mathbb{R}$ be a convex function; $x_0, x_1 \in \mathbb{R}^D$. Let Ψ and $\overline{\Psi}$ be differentiable at x_0 and x_1 correspondingly. Then the following statements are equivalent:*

1. $x_1 = \nabla\Psi(x_0)$;
2. $x_0 = \arg\max_{z\in\mathbb{R}^D}\left\{\langle x_1, z\rangle - \Psi(z)\right\}$;
3. *Fenchel-Young's equality:* $\Psi(x_0) + \overline{\Psi}(x_1) = \langle x_1, x_0\rangle$;
4. $x_0 = \nabla\overline{\Psi}(x_1)$;
5. $x_1 = \arg\max_{z\in\mathbb{R}^D}\left\{\langle z, x_0\rangle - \overline{\Psi}(z)\right\}$;

Proof: The lemma is a simplified version of [40, Theorem 1.16.4]. Also, the proof can be constructed by combining facts from [41, §3.3]. □

Lemma 2 (Main Integration Lemma) *For any two points $x_0, x_1 \in \mathbb{R}^D$ and a convex function Ψ, the following equality holds true:*

$$\int_0^1 \|u_t^\Psi(x_t) - (x_1 - x_0)\|^2 dt = 2\cdot[\Psi(x_0) + \overline{\Psi}(x_1) - \langle x_0, x_1\rangle], \tag{5.5}$$

where $x_t = tx_0 + (1-t)x_1$.

Proof: Following Proposition 1, I use the simplified loss form, i.e.,

$$\|u_t^\Psi(x_t) - (x_1 - x_0)\|^2 = \frac{1}{t^2}\|z_0 - x_0\|^2, \tag{5.6}$$

where $z_0 = z_0(t) = (\phi_t^\Psi)^{-1}(x_t)$ satisfies the equality:

$$x_t = t\nabla\Psi(z_0) + (1-t)z_0. \tag{5.7}$$

Next, I substitute (5.6) into rhs of (5.5) integrate w.r.t. time t from 0 excluding to 1 excluding (This exclusion does not change the integral):

$$\int_0^1 \|u_t^\Psi(x_t) - (x_1 - x_0)\|^2 dt = \int_0^1 \frac{1}{t^2}\|z_0 - x_0\|^2 dt. \tag{5.8}$$

To further simplify (5.8) I need some preliminary work. Following (5.7), I note:

$$\begin{aligned}
x_t = t\nabla\Psi(z_0) + (1-t)z_0 &= (1-t)x_0 + tx_1 \;\Rightarrow \\
t(\nabla\Psi(z_0) - x_1) &= (1-t)(x_0 - z_0) \;\Rightarrow \\
(\nabla\Psi(z_0) - x_1) &= \left(\frac{1-t}{t}\right)(x_0 - z_0) \;\Rightarrow \qquad (5.9)\\
\|\nabla\Psi(z_0) - x_1\|^2 &= \frac{(1-t)^2}{t^2}\|z_0 - x_0\|^2. \qquad (5.10)
\end{aligned}$$

Changing in (5.8) time variable t to $s = \frac{t}{1-t}$, $ds = \frac{dt}{(1-t)^2}$ and substitution of (5.10) yield:

$$\int_0^1 \frac{1}{t^2}\|z_0(t) - x_0\|^2 dt = \int_0^1 \frac{(1-t)^2}{t^2}\|z_0(t) - x_0\|^2 \frac{dt}{(1-t)^2} = \int_0^\infty \|\nabla\Psi(z_0(s)) - x_1\|^2 ds. \tag{5.11}$$

Notice that set of points $z_0(s(t)) = (\phi_t^\Psi)^{-1}(x_t), t \in (0,1)$ forms a curve in $\mathbb{R}^D$ with parameter t (or $s(t)$). Now I process formula (5.11) by switching from the integration w.r.t. parameter s to the integration along this curve. To do it properly two things are needed:

1. Limits of integration. The limits of integration along the curve $z_0(t)$ are:

$$\begin{aligned}
&z_0(t)|_{t=0} = x_0, \\
&z_0(t)|_{t=1} = (\nabla\Psi)^{-1}(x_1) \overset{\text{Lemma 1; 1, 4}}{=} \nabla\overline{\Psi}(x_1).
\end{aligned} \tag{5.12}$$

2. Expression under integral sign w.r.t. differential dz_0. Starting with (5.9), I derive:

$$\begin{aligned}
(5.9) \quad \Rightarrow \quad s(\nabla\Psi(z_0) - x_1) &= (x_0 - z_0) \;\Rightarrow \\
d[s(\nabla\Psi(z_0) - x_1)] &= d[x_0 - z_0] \;\Rightarrow \\
s\nabla^2\Psi(z_0)dz_0 + (\nabla\Psi(z_0) - x_1)ds &= -dz_0 \;\Rightarrow \\
(\nabla\Psi(z_0) - x_1)ds &= -(s\nabla^2\Psi(z_0) + I)dz_0. \qquad (5.13)
\end{aligned}$$

Now I proceed with (5.11):

$$\begin{aligned}
(5.11) &= \int_0^\infty \langle \nabla\Psi(z_0) - x_1, \nabla\Psi(z_0) - x_1\rangle\, ds \\
&\overset{(5.13)}{=} \int_{z_0} \langle x_1 - \nabla\Psi(z_0), (s\nabla^2\Psi(z_0) + I)dz_0\rangle \\
&= \int_{z_0} \langle x_1 - \nabla\Psi(z_0), dz_0\rangle + \int_{z_0} \langle s(x_1 - \nabla\Psi(z_0)), \nabla^2\Psi(z_0)dz_0\rangle \\
&\overset{(5.9)}{=} \int_{z_0} \langle x_1 - \nabla\Psi(z_0), dz_0\rangle + \int_{z_0} \langle z_0 - x_0, \nabla^2\Psi(z_0)dz_0\rangle. \qquad (5.14)
\end{aligned}$$

I notice that

$$\begin{aligned}
d\langle z_0, \nabla\Psi(z_0)\rangle &= \langle z_0, \nabla^2\Psi(z_0)dz_0\rangle + \langle dz_0, \nabla\Psi(z_0)\rangle \;\Rightarrow \\
\langle z_0, \nabla^2\Psi(z_0)dz_0\rangle &= d\langle z_0, \nabla\Psi(z_0)\rangle - \langle \nabla\Psi(z_0), dz_0\rangle.
\end{aligned}$$

As a consequence, I further proceed with (5.14):

$$\begin{aligned}
(5.14) &= \int_{z_0} \langle x_1 - \nabla\Psi(z_0), dz_0\rangle + \int_{z_0} \langle z_0 - x_0, \nabla^2\Psi(z_0)dz_0\rangle \\
&= \int_{z_0} \langle x_1, dz_0\rangle - \int_{z_0} \langle \nabla\Psi(z_0), dz_0\rangle \\
&\quad + \int_{z_0} d\langle z_0, \nabla\Psi(z_0)\rangle - \int_{z_0} \langle \nabla\Psi(z_0), dz_0\rangle - \int_{z_0} \langle x_0, \nabla^2\Psi(z_0)dz_0\rangle \\
&= \int_{z_0} \langle x_1, dz_0\rangle - 2\int_{z_0} \langle \nabla\Psi(z_0), dz_0\rangle + \int_{z_0} d\langle z_0, \nabla\Psi(z_0)\rangle - \int_{z_0} \langle x_0, \nabla^2\Psi(z_0)dz_0\rangle \qquad (5.15)
\end{aligned}$$

Under all integrals, I have closed form differentials

$$\begin{aligned}
\langle x_1, dz_0\rangle &= d\langle x_1, z_0\rangle, \\
\langle \nabla\Psi(z_0), dz_0\rangle &= d\Psi(z_0), \\
\langle x_0, \nabla^2\Psi(z_0)dz_0\rangle &= d\langle x_0, \nabla\Psi(z_0)\rangle.
\end{aligned}$$

I integrate them from initial point x_0 to the final $\nabla\overline{\Psi}(x_1)$ according to limits (5.12) and get

$$\begin{aligned}
(5.15) &= \int_{z_0} d\langle x_1, z_0\rangle - 2\int_{z_0} d\Psi(z_0) + \int_{z_0} d\langle z_0, \nabla\Psi(z_0)\rangle - \int_{z_0} d\langle x_0, \nabla\Psi(z_0)\rangle \\
&= \langle x_1, \nabla\overline{\Psi}(x_1)\rangle - \langle x_1, x_0\rangle + 2(\Psi(x_0) - \Psi(\nabla\overline{\Psi}(x_1))) + \langle(\nabla\overline{\Psi}(x_1), \nabla\Psi(\nabla\overline{\Psi}(x_1))\rangle \\
&\quad - \langle x_0, \nabla\Psi(x_0)\rangle + \langle x_0, \nabla\Psi(x_0)\rangle - \langle x_0, \nabla\Psi(\nabla\overline{\Psi}(x_1))\rangle. \qquad (5.16)
\end{aligned}$$

Now I use properties of conjugate functions (Lemma (1)):

$$\begin{aligned}
\Psi(\nabla\overline{\Psi}(x_1)) &\overset{4+3}{=} \langle \nabla\overline{\Psi}(x_1), x_1\rangle - \overline{\Psi}(x_1), \\
\nabla\Psi(\nabla\overline{\Psi}(x_1)) &\overset{4+1}{=} x_1.
\end{aligned}$$

This allows us to simplify (5.16):

$$\begin{aligned}(5.16) &= \langle x_1, \nabla\overline{\Psi}(x_1)\rangle - \langle x_1, x_0\rangle + 2(\Psi(x_0) + \overline{\Psi}(x_1) - \langle\nabla\overline{\Psi}(x_1), x_1\rangle) + \langle(\nabla\overline{\Psi}(x_1), x_1\rangle \\ &\quad - \langle x_0, \nabla\Psi(x_0)\rangle + \langle x_0, \nabla\Psi(x_0)\rangle - \langle x_0, x_1\rangle \\ &= 2[\Psi(x_0) + \overline{\Psi}(x_1) - \langle x_0, x_1\rangle].\end{aligned}$$

□

Integrating equality (5.5) over the given transport plan π and considering the formulas for the losses (2.5) and (3.3), I derive our Theorem 3.1.

Proof of Theorem 3.1 (OFM and OT connection)

Proof: Main Integration Lemma 2 states that for any fixed points x_0, x_1 one have

$$\int_0^1 \|x_1 - x_0 - u_t^{\Psi}(x_t)\|^2 dt = 2[\Psi(x_0) + \overline{\Psi}(x_1) - \langle x_0, x_1\rangle].$$

Taking math expectation over any plan π (integration w.r.t. points $x_0, x_1 \sim \pi$) gives

$$\underbrace{\mathbb{E}_{x_0,x_1\sim\pi}\int_0^1 \|u_t^{\Psi}(x_t) - (x_1 - x_0)\|^2 dt}_{=\mathcal{L}_{OFM}^{\pi}(\Psi)} = 2\cdot\underbrace{\mathbb{E}_{x_0,x_1\sim\pi}[\Psi(x_0) + \overline{\Psi}(x_1)]}_{=\mathcal{L}_{OT}(\Psi)} - \underbrace{2\cdot\mathbb{E}_{x_0,x_1\sim\pi}[\langle x_0, x_1\rangle]}_{=:\text{Const}'(\pi)},$$

where $\text{Const}'(\pi)$ does not depend on Ψ. Hence, both minimums of OFM loss $\mathcal{L}_{OFM}^{\pi}(\Psi)$ and of OT dual form loss $\mathcal{L}_{OT}(\Psi)$ are achieved at the same functions. □

Proof of Proposition 2 (Intractable Distance)

Proof: Recall the definitions of $\text{dist}(u, u^*)$ (3.5) and FM loss $\mathcal{L}_{FM}^{\pi^*}(u)$ (2.7):

$$\text{dist}(u, u^*) = \int_0^1 \int_{\mathbb{R}^D} \|u_t(x_t) - u_t^*(x_t)\|^2 \underbrace{\phi_t^*\#p_0(x_t)}_{=p_t^*(x_t)} dx_t dt,$$

$$\mathcal{L}_{FM}^{\pi^*}(u) = \int_0^1 \left\{\int_{\mathbb{R}^D\times\mathbb{R}^D} \|u_t(x_t) - (x_1 - x_0)\|^2 \pi^*(x_0, x_1) dx_0 dx_1\right\}, x_t = (1-t)x_0 + tx_1.$$

In the optimal plan π^*, each point x_0 almost surely goes to the single point $\nabla\Psi^*(x_0)$. Hence, in FM loss, one can leave only integration over initial points x_0 substituting $x_1 = \nabla\Psi^*(x_0)$ for fixed time t:

$$\begin{aligned}&\int_{\mathbb{R}^D\times\mathbb{R}^D} \|u_t(x_t) - (x_1 - x_0)\|^2 \pi^*(x_0, x_1) dx_0 dx_1 \\ &= \int_{\mathbb{R}^D} \|u_t(x_t) - (\nabla\Psi^*(x_0) - x_0)\|^2 p_0(x_0) dx_0\,,\ x_t = (1-t)x_0 + t\nabla\Psi^*(x_0). \quad (5.17)\end{aligned}$$

Notice that dynamic OT vector field $u^* = u^{\Psi^*}$ is the optimal one with potential Ψ^*. Moreover, for any point $x_t = (1-t)x_0 + t\nabla\Psi^*(x_0)$ generated by u^*, one can calculate $u_t^*(x_t) = u_t^{\Psi^*}(x_t) =$

$\nabla\Psi^*(x_0) - x_0$. It is the same expression as from (5.17), i.e.,

$$\begin{aligned}(5.17) &= \int\limits_{\mathbb{R}^D} \|u_t(x_t) - (\nabla\Psi^*(x_0) - x_0)\|^2 p_0(x_0)dx_0 \\ &= \int\limits_{\mathbb{R}^D} \|u_t(x_t) - u_t^*(x_t)\|^2 p_0(x_0)dx_0\,,\ \ x_t = (1-t)x_0 + t\nabla\Psi^*(x_0).\end{aligned}$$

Finally, I change the variable x_0 to $x_t = \phi_t^*(x_0)$, and probability changes as $p_0(x_0)dx_0 = \phi_t^*\#p_0(x_t)dx_t = p_t^*(x_t)dx_t$. In new variables, I obtain the result

$$\int\limits_{\mathbb{R}^D\times\mathbb{R}^D} \|u_t(x_t) - (x_1 - x_0)\|^2 \pi^*(x_0, x_1)dx_0dx_1 = \int_{\mathbb{R}^D} \|u_t(x_t) - u_t^*(x_t)\|^2 p_t^*(x_t)dx_t.$$

Hence, the integration over time t gives the desired equality

$$\text{dist}(u, u^*) = \mathcal{L}_{FM}^{\pi^*}(u),$$

and $\mathcal{L}_{FM}^{\pi^*}(u^*) = \text{dist}(u^*, u^*) = 0$. □

Proof of Proposition 3 (Tractable Distance For OFM)

Proof: For the vector field u^Ψ, I apply the formula for intractable distance from Proposition 2, i.e,

$$\text{dist}(u^\Psi, u^{\Psi^*}) = \mathcal{L}_{FM}^{\pi^*}(u^\Psi) - \mathcal{L}_{FM}^{\pi^*}(u^{\Psi^*}) \overset{(3.3)}{=} \mathcal{L}_{OFM}^{\pi^*}(\Psi) - \mathcal{L}_{OFM}^{\pi^*}(\Psi^*).$$

According to Main Integration Lemma 2, for any plan π and convex function Ψ, one has equality

$$\underbrace{\mathbb{E}_{x_0,x_1\sim\pi}\int_0^1 \|u_t^\Psi(x_t) - (x_1 - x_0)\|^2 dt}_{=\mathcal{L}_{OFM}^{\pi}(\Psi)} = 2\cdot\underbrace{\mathbb{E}_{x_0,x_1\sim\pi}[\Psi(x_0) + \overline{\Psi}(x_1)]}_{=\mathcal{L}_{OT}(\Psi)} - \underbrace{2\cdot\mathbb{E}_{x_0,x_1\sim\pi}[\langle x_0, x_1\rangle]}_{=:\text{Const}'(\pi)}.$$

Since $\text{Const}'(\pi)$ does not depend on Ψ, I have the same constant with $\Psi = \Psi^*$ and can eliminate it, i.e.,

$$\begin{cases}\mathcal{L}_{OFM}^{\pi}(\Psi) = 2\cdot\mathcal{L}_{OT}(\Psi) - \text{Const}'(\pi), \\ \mathcal{L}_{OFM}^{\pi}(\Psi^*) = 2\cdot\mathcal{L}_{OT}(\Psi^*) - \text{Const}'(\pi)\end{cases}$$
$$\Downarrow$$
$$\mathcal{L}_{OFM}^{\pi}(\Psi) - \mathcal{L}_{OFM}^{\pi}(\Psi^*) = 2\cdot\mathcal{L}_{OT}(\Psi) - 2\cdot\mathcal{L}_{OT}(\Psi^*). \tag{5.18}$$

The right part of (5.18) does not depend on a plan π, thus, the left part is invariant for any plan including optimal plan π^*, i.e.,

$$\mathcal{L}_{OFM}^{\pi}(\Psi) - \mathcal{L}_{OFM}^{\pi}(\Psi^*) = \mathcal{L}_{OFM}^{\pi^*}(\Psi) - \mathcal{L}_{OFM}^{\pi^*}(\Psi^*) = \text{dist}(u^\Psi, u^{\Psi^*}).$$

□

5.5 Action Matching

In this section, I demonstrate that OFM ideas can be applied to Action Matching (AM) [23] framework. In this setup, a stochastic process described by intermediate distributions p_t is available for sampling. The goal is to find time-dependent function $s_t : [0, 1] \times \mathbb{R}^D \to \mathbb{R}$, such that vector field $u_t = \nabla s_t$ generates considered process.

The function s_t can be found via minimization of the following objective AM loss:

$$\begin{aligned}\mathcal{L}_{AM}(s) &:= \int_{\mathbb{R}^D} s_0(x_0)p_0(x_0)dx_0 - \int_{\mathbb{R}^D} s_1(x_1)p_1(x_1)dx_1 \\ &+ \int_0^1 \int_{\mathbb{R}^D} \left[\frac{1}{2}\|\nabla s_t(x_t)\|^2 + \frac{\partial s_t}{\partial t}(x_t)\right] p_t(x_t)dx_t dt. \end{aligned} \tag{5.19}$$

Now, the optimal vector fields u^Ψ defined by convex functions Ψ are considered. I want to find explicit formula for the s^Ψ which gradient equals to the u^Ψ, i.e, $u_t^\Psi \equiv \nabla s_t^\Psi$.

Recall that for any $x_t \in \mathbb{R}^D$ the point $z_0 \in \mathbb{R}^D$ is required to satisfy

$$\begin{aligned} x_t &= t\nabla\Psi(z_0) + (1-t)z_0, \\ x_t &= \nabla\left(t\Psi(\cdot) + \frac{(1-t)}{2}\|\cdot\|^2\right)(z_0) := \nabla\varphi_t(z_0), \\ z_0 &= \nabla\overline{\varphi_t}(x_t). \end{aligned}$$

Then, the vector field $u_t^\Psi(x_t)$ can be expressed as:

$$\begin{aligned} u_t^\Psi(x_t) &= \nabla\Psi(z_0) - z_0 = \frac{x_t - z_0}{t} = \nabla\left(\underbrace{\frac{\|\cdot\|^2}{2t} - \frac{\overline{\varphi_t}}{t}}_{=:s_t}\right)(x_t), \\ s_t(x_t) &= \frac{\|x_t\|^2}{2t} - \frac{\overline{\varphi_t}(x_t)}{t}. \end{aligned} \tag{5.20}$$

The corner cases $t = 0$ and $t = 1$ can be written down as:

$$\begin{aligned} s_1(x_1) &= \frac{\|x_1\|^2}{2} - \overline{\Psi}(x_1), \\ s_0(x_0) &= \Psi(x_0) - \frac{\|x_0\|^2}{2}. \end{aligned}$$

In case of intermediate time $t \in (0, 1)$, firstly notice that

$$\frac{1}{2}\|\nabla s_t(x_t)\|^2 = \frac{1}{2}\frac{\|x_t - z_0\|^2}{t^2}.$$

Next, the time derivative from s_t is taken. Since Ψ-dependent terms are important, the $\|\cdot\|^2/2t$ term from (5.20) can be omitted. Hence, the remaining term is

$$\frac{\overline{\varphi_t}(x_t)}{t} = \frac{1}{t}\max_{z\in\mathbb{R}^D}\left\{\langle x_t, z\rangle - t\Psi(z) - \frac{(1-t)}{2}\|z\|^2\right\} = \max_{z\in\mathbb{R}^D}\left\{\frac{\langle x_t, z\rangle}{t} - \Psi(z) - \frac{(1-t)}{2t}\|z\|^2\right\}.$$

Moreover, max is achieved at the point z_0. According to Envelope Theorem, in order to take derivative from maximum w.r.t time t, one needs to take it from the maximized function, and then put

the point z_0 at which maximum is achieved. In other words, it holds

$$\frac{\partial(\overline{\varphi}_t/t)}{\partial t}(x_t) = -\frac{\langle x_t, z_0\rangle}{t^2} + \frac{\|z_0\|^2}{2t^2}. \tag{5.21}$$

Substituting (5.21) to (5.20), I obtain

$$\frac{\partial s_t}{\partial t}(x_t) = -\frac{\|x_t\|^2}{2t^2} + \frac{\langle x_t, z_0\rangle}{t^2} - \frac{\|z_0\|^2}{2t^2} = -\frac{1}{2}\frac{\|x_t - z_0\|^2}{t^2}.$$

Therefore, in case of optimal vector fields, for any $t \in [0, 1]$ and $x_t \in \mathbb{R}^D$, it holds

$$\left[\frac{1}{2}\|\nabla s_t(x_t)\|^2 + \frac{\partial s_t}{\partial t}(x_t)\right] = \frac{1}{2}\frac{\|x_t - z_0\|^2}{t^2} - \frac{1}{2}\frac{\|x_t - z_0\|^2}{t^2} \equiv 0.$$

Moreover, AM Loss is completely equivalent to the dual form OT loss, i.e.,:

$$\begin{aligned}\mathcal{L}_{AM}(s^\Psi) &= \int_{\mathbb{R}^D} \Psi(x_0)p_0(x_0)dx_0 + \int_{\mathbb{R}^D} \overline{\Psi}(x_1)p_1(x_1)dx_1 \\ &- \int_{\mathbb{R}^D} \frac{\|x_0\|^2}{2}p_0(x_0)dx_0 - \int_{\mathbb{R}^D} \frac{\|x_1\|^2}{2}p_1(x_1)dx_1.\end{aligned}$$

5.6 Experiments details

OFM implementation

To implement OFM approach in practice, the fully-connected ICNN architectures proposed in [42, Appendix B2] (`W2GN_ICNN`) and [43, Appendix E1] (`CPF_ICNN`) are adopted. To ensure the convexity, both architectures place some restrictions on the NN's weights and utilized activation functions, see the particular details in the corresponding papers. The implementations are taken from their official repositories:

https://github.com/iamalexkorotin/Wasserstein2Benchmark;
https://github.com/CW-Huang/CP-Flow.

The hyper-parameters of Algorithm 1 and utilized ICNNs for different experiments are aggregated in Table 5.1. In all experiments LBFGS solver (`torch.optim.LBFGS`) with K_{sub} optimization steps and early stopping criteria based on gradient norm is used as the $SubOpt$ optimizer. To find the initial point z_0^i (Step 5 of our Algorithm 1), $SubOpt$ is initialized with x_{ti}^i. As the Opt optimizer Adam is used with learning rate lr and other hyperparameters set to be default.

Experiment	ICNN architecture Ψ_θ	K	B	lr	K_{sub}
Illustrative 2D	`CPF_ICNN`, $\mathbb{R}^2 \to \mathbb{R}$, Softplus, [1024, 1024]	30K	1024	10^{-2}	5
W2 bench., dim. D	`W2GN_ICNN`, $\mathbb{R}^D \to \mathbb{R}$, CELU, [128, 128, 64]	30K	1024	10^{-3}	50
ALAE	`W2GN_ICNN`, $\mathbb{R}^{512} \to \mathbb{R}$, CELU, [1024, 1024]	10K	128	10^{-3}	10

Table 5.1: Hyper-parameters of our OFM solvers in different experiments

Minibatch. Similarly to OT-CFM, in some of experiments non-independent initial plans π are used to improve convergence. The plan π is constructed as follows: for independently sampled minibatches X_0, X_1 of the same size B, the optimal discrete map is built and applied to reorder the pairs of samples. I stress that considering minibatch OT for our method is done exclusively

to speed up the training process. Theoretically, OFM method is agnostic to initial plan π and is guaranteed to have an optimum in dynamic OT solution.

Benchmark details

In the experiments, the exponential moving average (EMA) [44, 45] of the trained model weights is used. EMA creates a smoothed copy of the model whose weights are updated at each new training iteration $t+1$ as $\theta_{t+1}^{\text{ema}} = \alpha\theta_t^{\text{ema}} + (1-\alpha)\theta_{t+1}$, where θ_{t+1} are the newly updated original trained weights. The final metrics are calculated with $\alpha = 0.999$.

Details of Solvers. Neural networks' architectures of competing Flow Matching methods and their parameters used in benchmark experiments are presented in Table 5.2. In this Table, "FC" stands for "fully-connected".

Solver	Architecture	Activation	Hidden layers	Optimizer	Batch size	Learning rate	Iter. per round * rounds
OT CFM [9]	FC NN $\mathbb{R}^D \times [0,1] \to \mathbb{R}^D$	ReLU	[128, 128, 64]	RMSprop	1024	10^{-3}	200.000
RF [6]	FC NN $\mathbb{R}^D \times [0,1] \to \mathbb{R}^D$	ReLU	[128, 128, 64]	RMSProp	1024	10^{-4}	$65.000 * 3$
c-RF [5]	FC NN $\mathbb{R}^D \times [0,1] \to \mathbb{R}$	ReLU	[128, 128, 64]	RMSProp	1024	10^{-5}	$100.000 * 2$

Table 5.2: Parameters of models fitted on benchmark in dimensions $D = 2, 4, 8, 16, 32, 64, 128, 256$.

Time variable t in $(c-)$RF and OT-CFM's architectures is added as one more dimensionality in input without special preprocessing. In RF and c-RF, ODE are solved via Explicit Runge-Kutta method of order $5(4)$ [46] with absolute tolerance $10^{-4} - 10^{-6}$. In OFM and c-RF, gradients over input are calculated via autograd of `PyTorch`.

Following the authors of RF [6], I run only $2-3$ rounds in RF. In further rounds, straightness and metrics change insignificantly, while the error of target distribution learning still accumulates.

The implementations of OT-CFM [9] and RF [6] are based on the official repositories:

`https://github.com/atong01/conditional-flow-matching`
`https://github.com/gnobitab/RectifiedFlow`

Implementation of c-RF follows the RF framework with the modification of optimized NN's architecture. Instead of $\mathbb{R}^D \times [0,1] \to \mathbb{R}^D$ net, I parametrize time-dependent scalar valued model $\mathbb{R}^D \times [0,1] \to \mathbb{R}$ which gradients are set to be the vector field.