# Optimal Flow Matching: Learning Straight Trajectories in Just One Step

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

Over the several recent years, there has been a boom in development of Flow Matching (FM) methods for generative modeling. One intriguing property pursued by the community is the ability to learn flows with straight trajectories which realize the Optimal Transport (OT) displacements. Straightness is crucial for the fast integration (inference) of the learned flow's paths. Unfortunately, most existing flow straightening methods are based on non-trivial iterative FM procedures which accumulate the error during training or exploit heuristics based on minibatch OT. To address these issues, we develop and theoretically justify the novel **Optimal Flow Matching** (OFM) approach which allows recovering the straight OT displacement for the quadratic transport in just one FM step. The main idea of our approach is the employment of vector field for FM which are parameterized by convex functions.

## 1 Introduction

Recent success in generative modeling [27, 12, 7] is mostly driven by pervasive Flow Matching (FM) [24] 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 original FM [25, 26, 30], which aim to recover more straight paths.

Rectified Flow (RF) method [25, 26] iteratively solves FM and gradually rectifies trajectories. Unfortunately, in each FM iteration, it **accumulates the error** that may spoil the performance.

A popular branch of approaches to straighten trajectories is based on the connection between straight paths and Optimal Transport (OT) [38]. The main goal of OT is to find the way to move one probability distribution to another with the minimal effort. Such optimal transportations are usually described by ODEs with straight trajectories. In OT Conditional Flow Matching (OT-CFM) [30, 35], the authors propose to apply FM on top of OT solution between batches from considered distributions. Unfortunately, such a heuristic does not actually guarantee straight paths because of errors of **minibatch OT biases**.

**Contributions.** In this paper, we fix the above-mentioned problems of the straightening methods.

1. We 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.
2. In OFM, one can optionally use minibatch OT or any other transport plan, which is completely theoretically justified.
3. We demonstrate the potential of OFM in the series of experiments and benchmarks.

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

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 [3].

**Notations.** For vectors $x, y \in \mathbb{R}^D$, we denote the inner product by $\langle x, y\rangle$ and the corresponding $\ell_2$ norm by $\|x\| := \sqrt{\langle x, x\rangle}$. We use $\mathcal{P}_{2,ac}(\mathbb{R}^D)$ to refer to the set of absolute continuous probability distributions with the finite second moment. For the push-forward operator, we use symbol $\#$.

## 2 Background and Related Works

In this section, we provide all necessary backgrounds for the theory. Firstly, we recall static (§2.1) and dynamic (§2.2) formulations of Optimal Transport and solvers (§2.3) for them. Then, we recall Flow Matching (§2.4.1) and flow straightening approaches such as OT-CFM (§2.4.2) and Rectified Flow (§2.4.3).

### 2.1 Static Optimal Transport

**Monge's and Kantorovich's formulations.** Consider two probability distributions $p_0, p_1 \in \mathcal{P}_{2,ac}(\mathbb{R}^D)$ and a cost function $c : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$. Monge's Optimal Transport formulation is the following minimization problem:

$$\inf_{T\#p_0=p_1} \int_{\mathbb{R}^D} c(x_0, T(x_0))p(x_0)dx_0, \tag{1}$$

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 (1) 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

$$\inf_{\pi\in\Pi(p_0,p_1)} \int_{\mathbb{R}^D\times\mathbb{R}^D} c(x_0, x_1)\pi(x_0, x_1)dx_0dx_1. \tag{2}$$

With mild assumptions on $p_0, p_1$, the infimum is always achieved (possibly not uniquely). An optimal $\pi^* \in \Pi(p_0, p_1)$ is called optimal transport plan. If optimal $\pi^*$ exists and has a form $[\mathrm{id}, T^*]\#p_0$, then $T^*$ is the solution of Monge's formulation (1).

**Quadratic cost function.** In our paper, we mostly consider the quadratic cost function $c(x_0, x_1) = \frac{\|x_0-x_1\|^2}{2}$. In this case, infimums in both Monge's and Kantorovich's OT are always uniquely attained [38, Brenier's Theorem 2.12]. They are related by $\pi^* = [\mathrm{id}, T^*]\#p_0$. Moreover, the optimal values of (1) and (2) are equal to each other. The square root of the optimal value is called Wasserstein-2 distance $\mathbb{W}_2(p_0, p_1)$ between distributions $p_0$ and $p_1$, i.e.,

$$\mathbb{W}_2^2(p_0, p_1) := \min_{\pi\in\Pi(p_0,p_1)} \int_{\mathbb{R}^D\times\mathbb{R}^D} \frac{\|x_1 - x_0\|^2}{2}\pi(x_0, x_1)dx_0dx_1 = \min_{T\#p_0=p_1} \int_{\mathbb{R}^D} \frac{\|x_0 - T(x_0)\|^2}{2}p(x_0)dx_0. \tag{3}$$

**Dual formulation.** For the quadratic cost, problem (3) has the equivalent dual form [38]:

$$\mathbb{W}_2^2(p_0, p_1) = \textsc{Const}(p_0, p_1) - \min_{\text{convex } \Psi} \underbrace{\left[\int_{\mathbb{R}^D} \Psi(x_0)p_0(x_0)dx_0 + \int_{\mathbb{R}^D} \overline{\Psi}(x_1)p_1(x_1)dx_1\right]}_{=:\mathcal{L}_{OT}(\Psi)}, \tag{4}$$

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 $\Psi$.

The term $\textsc{Const}(p_0, p_1)$ does not depend on $\Psi$. Therefore, the minimization (3) over transport plans $\pi$ is equivalent to the minimization of $\mathcal{L}_{OT}(\Psi)$ from (4) over convex functions $\Psi$. Moreover, the optimal transport map $T^*$ can be expressed via an optimal $\Psi^*$ [38], namely,

$$T^* = \nabla\Psi^*. \tag{5}$$

The optimal $\Psi^*$ is called the *Brenier potential*.

### 2.2 Dynamic Optimal Transport

In [4], the authors show that calculating of Optimal Transport (3) for the quadratic cost can be equivalently reformulated in a dynamic form. This form operates with a vector fields defining time-dependent mass transport instead of just static transport maps.

**Preliminaries.** We consider the fixed time interval $[0, 1]$. Let $u(t, \cdot) \equiv u_t(\cdot) : [0, 1] \times \mathbb{R}^D \to \mathbb{R}^D$ be a vector field and $\{\{z_t\}_{t\in[0,1]}\}$ be the set of random trajectories such that for each trajectory $\{z_t\}_{t\in[0,1]}$ the starting point $z_0$ is sampled from $p_0$ and $z_t$ satisfies the differential equation

$$dz_t \quad = \quad u_t(z_t)dt, \quad z_0 \sim p_0. \tag{6}$$

In other words, the trajectory $\{z_t\}_{t\in[0,1]}$ is defined by its initial point $z_0 \sim p_0$ and goes along the speed vector $u_t(z_t)$. Under mild assumptions on $u$, for each initial $z_0$, the trajectory is unique.

Let $\phi^u(t, \cdot) \equiv \phi^u_t(\cdot) : [0, 1] \times \mathbb{R}^D \to \mathbb{R}^D$ denote the flow map, i.e., it is the function that maps the initial $z_0$ to its position at moment of time $t$ according to the ODE (6), i.e.,

$$d\phi^u_t(z_0) \quad = \quad u_t(\phi^u_t(z_0)), \quad \phi^u_0(z_0) = z_0. \tag{7}$$

If initial points $z_0$ of trajectories are distributed according to $p_0$, then (6) defines a distribution $p_t$ of $z_t$ at time $t$, which can be expressed via with the push-forward operator, i.e., $p_t := \phi^u_t \# p_0$.

**Benamou–Brenier problem.** Dynamic OT is the following minimization problem:

$$\begin{aligned} \mathbb{W}_2^2(p_0, p_1) = \inf_u \quad & \int_0^1 \int_{\mathbb{R}^D} \frac{\|u_t(x)\|_2^2}{2} \underbrace{\phi^u_t \# p_0(x)}_{:=p_t(x)} dx dt, \\ s.t. \quad & \phi^u_1 \# p_0 = p_1. \end{aligned} \tag{8}$$

In (8), we look for the vector fields $u$ that define the flows which start at $p_0$ and end at $p_1$. Among such flows, we seek for one which has the minimal kinetic energy over the entire time interval.

There is a connection between the static OT map $T^* = \nabla\Psi^*$ and the dynamic OT solution $u^*$. Namely, for every initial point $z_0$, the vector field $u^*$ defines a linear trajectory $\{z_t\}_{t\in[0,1]}$:

$$z_t = t\nabla\Psi^*(z_0) + (1 - t)z_0, \quad \forall t \in [0, 1]. \tag{9}$$

### 2.3 Continuous Optimal Transport Solvers

There exist a variety of continuous OT solvers, see [17] for a survey. In this paper, we focus only on the most relevant ones, called the ICNN-based solvers [34, 28, 21, 17]. These solvers directly minimize objective $\mathcal{L}_{OT}$ from (4) parametrizing a class of convex functions with convex in input neural networks called ICNNs [3] (for more details, see "Parametrization of $\Psi$" in §3.2). 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. Envelope Theorem [1] allows obtaining closed-form formula for the gradient of the loss.

### 2.4 Flow Matching Framework

In this section, we recall popular approaches [26, 25, 30] to find fields $u$ which transport a given probability distribution $p_0$ to a target $p_1$ and their relation to OT.

#### 2.4.1 Flow Matching (FM)

To find such a field, one considers two points $x_0, x_1$ sampled from a transport plan $\pi \in \Pi(p_0, p_1)$, e.g., the independent plan $p_0 \times p_1$. The vector field $u$ is encouraged to follow the direction $x_1 - x_0$ of the linear interpolation $x_t = (1 - t)x_0 + tx_1$ at any moment $t \in [0, 1]$. It is achieved by solving the following problem:

$$\min_u \mathcal{L}^{\pi}_{FM}(u) := \int_0^1 \left\{ \int_{\mathbb{R}^D \times \mathbb{R}^D} \|u_t(x_t) - (x_1 - x_0)\|^2 \pi(x_0, x_1) dx_0 dx_1 \right\} dt, x_t = (1 - t)x_0 + tx_1. \tag{10}$$

We denote the solution of (10) and the corresponding flow map (7) by $u^\pi$ and $\phi^\pi$, respectively.

The intuition of this procedure is as follows: linear interpolation $x_t = (1-t)x_0 + tx_1$ is an intuitive way to move $p_0$ to $p_1$, but it requires knowing $x_1$. By fitting $u$ with the direction $x_1 - x_0$, one yields the vector field that can construct this interpolation without any information about $x_1$.

The set of trajectories $\{\{z_t\}_{t\in[0,1]}\}$ generated by $u_t^\pi$ (with $z_0 \sim p_0$) has a useful property: the flow map $\phi_1^\pi$ transforms distribution $p_0$ to distribution $p_1$ for any initial transport plan $\pi$. Moreover, marginal distribution $p_t = \phi_t^\pi \# p_0$ is equal to the distribution of linear interpolation $x_t = (1-t)x_0 + tx_1$ for any $t$ and $x_0, x_1 \sim \pi$. This feature is called marginal preserving property.

To push point $x_0$ according to learned $u$, one needs to integrate ODE (6) 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.

Researchers noticed that some initial plans $\pi$ can result in more straight paths after FM rather than the standard independent plan $p_0 \times p_1$. The two most popular approaches to choose better plans are Optimal Transport Conditional Flow Matching [30, 35] and Rectified Flow [26].

### 2.4.2 Optimal Transport Conditional Flow Matching (OT-CFM)

If one uses the OT plan $\pi^*$ as the initial plan for FM, then it returns the Brenier's vector field $u^*$ which generates exactly straight trajectories (9). However, typically, the true OT plan $\pi^*$ 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 $\pi$ to be close to the optimal $\pi^*$. Inspired by this, the authors of OT-CFM [30, 35] take the advantage of minibatch OT plan approximation. 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.

The main drawback of OT-CFM is that it recovers only biased dynamic OT solution. In order to converge to the true transport plan the batch size should be large [5], while with a growth of batch size computational time increases drastically [36]. In practice, batch sizes that ensure approximation good enough for applications are nearly infeasible to work with.

### 2.4.3 Rectified Flow (RF)

In [26], the authors propose an iterative approach to refine the plan $\pi$, straightening the trajectories more and more with each iteration. Formally, Flow Matching procedure denoted by FM takes the transport plan $\pi$ as input and returns an optimal flow map via solving (10):

$$\phi^\pi := \text{FM}(\pi). \tag{11}$$

One can iteratively apply 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

$$\phi^{K+1} \quad = \quad \text{FM}(\pi^K), \quad \pi^{K+1} = [\text{id}, \phi^{K+1}]\# p_0, \tag{12}$$

where $\phi^K, \pi^K$ 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 notice that for any convex cost function $c$ the flow map $\phi_1^\pi$ from Flow Matching yields lower or equal transport cost than initial transport plan $\pi$:

$$\int_{\mathbb{R}^D} c(x_0, \phi_1^\pi(x_0)) p_0(x_0) dx_0 \leq \int_{\mathbb{R}^D \times \mathbb{R}^D} c(x_0, x_1)\pi(x_0, x_1) dx_0 dx_1. \tag{13}$$

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

With each iteration of Rectified Flow (12), transport costs for all convex cost functions do not increase, but, for a given cost function, convergence to its own OT plan is not guaranteed. In [25], the authors address this issue and, for any particular convex cost function $c$, modify Rectified Flow 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 function and $\bar{c}$ is the convex conjugate of the cost function $c$.

Unfortunately, in practice, with each iteration ($c$-)RF accumulates error caused by inexactness from previous iterations. Due to neural approximations, we 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 $\pi^{K+1} = [\mathrm{id}, \phi^{K+1}] \# p_0$ it has to integrate ODE.

# 3 Optimal Flow Matching (OFM)

In this section, we provide the design of our novel Optimal Flow Matching algorithm (1) that fixes main problems of Rectified Flow and OT-CFM approaches described above. In theory, it obtains exactly **straight trajectories** and recovers the unbiased optimal transport map for the quadratic cost **just in one FM iteration** with **any** initial transport plan. Moreover, during inference, OFM does not require solving ODE to transport points.

We discuss theory behind our approach (§3.1), its practical implementation aspects (§3.2) and the relation to prior works (§3.3). All proofs are located in Appendix 6.

## 3.1 Theory: Deriving the Optimization Loss

Consider the quadratic cost function $c(x_0, x_1) = \frac{\|x_0 - x_1\|^2}{2}$. We aim to solve the Benamou–Brenier problem (8) between distributions $p_0$ and $p_1$ and construct the dynamic OT field $u^*$, since it generates straight trajectories. The main idea of our Optimal Flow Matching (OFM) is to minimize the Flow Matching loss (10) not over all possible vector fields $u$, but only over specific *optimal* ones, which yield straight paths by construction.

**Optimal vector fields.** We say that a vector field $u^\Psi$ 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.,

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

The function $\Psi$ defines the ODE

$$dz_t \quad = \quad (\nabla\Psi(z_0) - z_0)dt, \quad z_t|_{t=0} = z_0. \tag{14}$$

Equation (14) does not provide a closed formula for $u^\Psi$ as it depends on $z_0$. The explicit formula is constructed as follows: for a time $t \in [0,1]$ and point $x_t$, we can find a trajectory $\{z_t\}_{t\in[0,1]}$ s.t.

$$x_t = z_t = (1-t)z_0 + t\nabla\Psi(z_0) \tag{15}$$

and recover the initial point $z_0$. We postpone the solution of this problem to §3.2. For now, we define the inverse of flow map (7) as $(\phi_t^\Psi)^{-1}(x_t) := z_0$ and the vector field $u_t^\Psi(x_t) := \nabla\Psi(z_0) - z_0 = \nabla\Psi((\phi_t^\Psi)^{-1}(x_t)) - (\phi_t^\Psi)^{-1}(x_t)$, which generates ODE (14), i.e., $dz_t = u_t^\Psi(z_t)dt$.

We highlight that the solution of dynamic OT lies in the class of optimal vector fields, since it generates linear trajectories (9) with the Brenier potential $\Psi^*$ (5).

**Training objective.** Our Optimal Flow Matching (OFM) approach is as follows: we restrict the optimization domain of FM (10) with fixed plan $\pi$ only to the optimal vector fields. We put the formula for the vector field $u_\Psi$ into FM loss from (10) and define our Optimal Flow Matching loss:

$$\mathcal{L}_{OFM}^\pi(\Psi) \quad := \quad \mathcal{L}_{FM}^\pi(u^\Psi) = \int_0^1 \left\{ \int_{\mathbb{R}^D\times\mathbb{R}^D} \|u_t^\Psi(x_t) - (x_1 - x_0)\|^2 \pi(x_0, x_1) dx_0 dx_1 \right\} dt, \tag{16}$$

$$x_t \quad = \quad (1-t)x_0 + tx_1.$$

**Proposition 1 (Simplified OFM Loss)** *We can simplify (16) to a more suitable form:*

$$\mathcal{L}_{OFM}^\pi(\Psi) = \int_0^1 \left\{ \int_{\mathbb{R}^D\times\mathbb{R}^D} \left\| \frac{(\phi_t^\Psi)^{-1}(x_t) - x_0}{t} \right\|^2 \pi(x_0, x_1) dx_0 dx_1 \right\} dt, x_t = (1-t)x_0 + tx_1. \tag{17}$$

The simplified form (17) gives a hint for understanding of OFM loss: it measures how well $\Psi$ restores initial points $x_0$ of linear interpolations depending on future point $x_t$ and time $t$. The main technical result, which is used to derive the main properties of OFM, is presented in Lemma 1.

**Lemma 1 (Main Integration Lemma)** *For any two points $x_0, x_1 \in \mathbb{R}^D$ and a convex function $\Psi$, the following equality holds true:*

$$\int_0^1 \|u_t^{\Psi}(x_t) - (x_1 - x_0)\|^2 dt = 2 \cdot [\Psi(x_0) + \overline{\Psi}(x_1) - \langle x_0, x_1 \rangle]. \tag{18}$$

The proof that integration over time in case of linear interpolation can be calculated analytically for any $x_0, x_1, \Psi$ via fancy formula (18) is not trivial and requires tricky integration techniques. As a consequence of Lemma 1, minimization of OFM loss (16) over $\Psi$ recovers desired dynamic OT.

**Theorem 1 (OFM and OT connection)** *Let us 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, losses $\mathcal{L}_{OT}(\Psi)$ and $\mathcal{L}_{OFM}^{\pi}(\Psi)$, defined in (4) and (16), respectively, have the same minimizers, i.e.,*

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

**Generative properties of OFM.** In this paragraph, we provide another view on our OFM approach. In our OFM, we wish to construct a vector field $u$ close to the dynamic OT field $u^*$. We can use the least square regression to measure the distance between them:

$$\text{DIST}(u, u^*) := \int_0^1 \int_{\mathbb{R}^D} \|u_t(x_t) - u_t^*(x_t)\|^2 \underbrace{\phi_t^* \# p_0(x_t)}_{:= p_t^*(x_t)} dx_t dt. \tag{19}$$

**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 (10) with the optimal plan $\pi^*$, i.e.,*

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

We can not minimize intractable $\text{DIST}(u, u^*)$ since the optimal plan $\pi^*$ is unknown. In OT-CFM [35], authors heuristically approximate $\pi^*$ in $\mathcal{L}_{FM}^{\pi^*}(u)$, but obtain biased solution. Surprisingly, for the *optimal* vector fields, the distance can be calculated explicitly via **any** known plan $\pi$.

**Proposition 3 (Tractable Distance For OFM)** *The distance $\text{DIST}(u^{\Psi}, u^{\Psi^*})$ between an **optimal** vector field $u^{\Psi}$ generated by a convex function $\Psi$ and the vector field $u^{\Psi^*}$ with the Brenier potential $\Psi^*$ can be evaluated directly via OFM loss (16) and **any** plan $\pi$:*

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

In (31), the first term is our tractable OFM loss, and the second term does not depend on $\Psi$. Hence, during the whole minimization process in our OFM, we gradually lower the distance (19) between the current vector field and the dynamic OT field up to the complete match.

### 3.2 Practical implementation aspects

In this subsection, we explain the details of optimization of our Optimal Flow Matching loss (16).

**Flow map inversion.** In order to find the initial point $z_0 = (\phi_t^{\Psi})^{-1}(x_t)$, we note that (15)

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

is equivalent to

$$\nabla\left(\frac{(1-t)}{2}\|\cdot\|^2 + t\Psi(\cdot) - \langle x_t, \cdot\rangle\right)(z_0) = 0.$$

The function under gradient operator $\nabla$ has minimum at the required point $z_0$, since at $z_0$ the gradient of it equals 0. If $t < 1$ the function is at least $(1-t)$-strongly convex, and the minimum is unique. The case $t = 1$ is negligible in practice, since it has zero probability to appear during training.

We can reduce the problem of inversion to the following minimization subproblem

$$(\phi_t^{\Psi})^{-1}(x_t) = \arg\min_{z_0 \in \mathbb{R}^D} \left[ \frac{(1-t)}{2} \|z_0\|^2 + t\Psi(z_0) - \langle x_t, z_0 \rangle \right]. \tag{20}$$

Optimization subproblem (20) 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).

**Parametrization of $\Psi$.** In practical implementation, we parametrize the class of convex functions with Input Convex Neural Networks (ICNNs) [3] $\Psi_\theta$ and parameters $\theta$. These are scalar-valued neural networks built in such a way that the network is convex in inputs. They consist of fully-connected or convolution blocks, some weights of which are set to be non-negative in order to keep 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.** The calculation of OFM loss (16) requires solving the minimization subproblem (20). Due to it, here we provide an explicit formula for gradient of (16), such that it does not contain the gradient of $(\phi_t^{\Psi_\theta})^{-1}$ w.r.t. parameters $\theta$.

**Proposition 4 (Explicit Loss Gradient Formula)** *The gradient of $\mathcal{L}^{\pi}_{OFM}$ can be calculated as*

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

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

**Algorithm.** The Optimal Flow Matching pseudocode is presented in 1. We estimate math expectation over plan $\pi$ and time $t$ with uniform distribution on $[0, 1]$ via unbiased Monte Carlo estimate.

**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 $\Psi_\theta$

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

### 3.3 Relation to Prior Works

In this subsection, we compare our Optimal Flow Matching and previous straightening approaches. 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-CFM** [35]. Unlike our 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 $\pi$ in our OFM is completely justified in Theorem 1.

**Rectified Flow** [26, 25]. In Rectified Flows [26], 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 [25] 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, we work only with the quadratic cost function, but retrieve its OT solution in **just one FM iteration** without simulation of the trajectories.

**Light and Optimal Schrödinger Bridge.** In [13], the authors observe the relation between Entropic Optimal Transport (EOT) [23, 10] and Bridge Matching (BM) [32] problems. These are stochastic analogs of OT and FM, respectively. In EOT and BM, instead of deterministic ODE and flows, one considers stochastic processes with non-zero stochasticity. The authors prove that, during BM, one can restrict considered processes only to the specific ones and retrieve the solution of EOT. Hypothetically, our OT/FM case is a limit of their EOT/BM case when the stochasticity tends to zero. Proofs in [13] for EOT are based on sophisticated KL divergence properties. We do not know whether our results for OFM can be derived by taking the limit of their stochastic case. To derive the properties of our OFM, we use **completely different proof techniques** based on computing integrals over curves rather than KL-based techniques. Besides, in practice, the authors of [13] mostly focus on Gaussian mixture parametrization while our method allows using neural networks (ICNNs).

# 4 Experiments

**Minibatch.** Similarly to OT-CFM, we use different initial plans $\pi$ to improve convergence. We construct $\pi$ as follows: for independently sampled minibatches $X_0, X_1$ of the same size $B$, we build the optimal discrete map and apply it to reorder the pairs of samples. We stress that considering minibatch OT for our method is done exclusively to speed up the training process. Theoretically, our method is agnostic to initial plan $\pi$ and is guaranteed to have an optimum in dynamic OT solution.

### 4.0.1 Toy example

We illustrate proof-of-concept of our Optimal Flow Matching on 2D setup. We solve the OT between a standard Gaussian distribution $p_0 = \mathcal{N}(0, I)$ and a Swiss roll $p_1$ depicted in the Figure 1. We run our Algorithm 1 for different stochastic plans $\pi$: independent plan $p_0 \times p_1$, minibatch discrete OT with batch size $B = 64$ and $B = 128$. In more details, we run 20000 iterations of Optimal Flow Matching and use RMSprop with learning rate $10^{-3}$ for both $\theta$ parameters optimization and inversion $z_0 = (\phi_t^{\Psi})^{-1}(x_t)$. The results are presented in Figure 1.

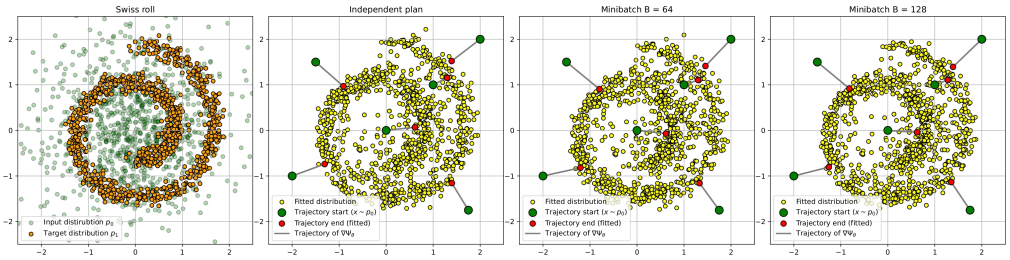


Figure 1: Swiss roll experiment

We empirically see that our Optimal Flow Matching finds the same solution for all considered plans $\pi$, the plan itself effects only necessary time to obtain maps.

#### 4.0.2 High-dimensional OT Benchmarks

To compare our OFM with other FM based methods and OT solvers, we run OT Benchmark [20]. The authors provide high-dimensional continuous distributions $p_0, p_1$ for which the ground truth OT map $T^*$ is known by the construction. To assess the quality of retrieved transport maps, we use standard metric *learned variance percentage* $\mathcal{L}^2$-UVP.

**Solvers.** We evaluate Flow Matching (FM), Conditional Flow Matching (OT-CFM), Rectified Flow (RF), $c$-Rectified Flow ($c$-RF) and the most relevant OT solver MMv-1 [33]. In MMv-1, the authors directly minimize the dual formulation loss $\mathcal{L}_{OT}$ (4) parametrizing $\Psi$ with ICNNs and calculating $\overline{\Psi}(x_1)$ via convex optimization subproblem, which is similar to our inversion (20). We also provide results for linear map translating means and variances of distributions to each other.

We consider 3 initial plans: independent plan with batch size 1024 (Ind 1024), minibatch OT of size 1024 which is either used as a whole batch (MB 1024) or divided into subbatches of size 64 (MB 64). Performance results are presented in Table 1. More details and metrics are located in Appendix 6.1.

| Solver | **Solver type\|DIM** | $D=2$ | $D=4$ | $D=8$ | $D=16$ | $D=32$ | $D=64$ | $D=128$ | $D=256$ |
|---|---|---|---|---|---|---|---|---|---|
| MMv1 | OT solver | 0.2 | 1.0 | 1.8 | 1.4 | 6.9 | 8.1 | 2.2 | 2.6 |
| OFM Ind 1024 **(Ours)** | | 0.51 | – | – | 2.71 | – | 10.98 | 16.78 | – |
| OFM MB 64 **(Ours)** | | – | – | – | – | – | – | – | – |
| OFM MB 1024 **(Ours)** | | 0.70 | – | – | 2.58 | – | 10.66 | 8.56 | – |
| OT-CFM 64 | Flow Matching | 0.68 | 0.99 | 2.98 | 5.0 | 8.2 | 12.0 | 13.8 | 31.4 |
| OT-CFM 1024 | | 0.16 | 0.73 | 2.27 | 4.33 | 7.9 | 11.4 | 12.1 | 27.5 |
| $c$-RF | | 1.56 | 13.11 | 17.87 | 35.39 | 48.46 | 66.52 | 68.08 | 76.48 |
| RF | | 8.58 | 49.46 | 51.25 | 63.33 | 63.52 | 85.13 | 84.49 | 83.13 |
| Linear | Baseline | 14.1 | 14.9 | 27.3 | 41.6 | 55.3 | 63.9 | 63.6 | 67.4 |

Table 1: $\mathcal{L}^2-$UVP values of solvers fitted on high-dimensional benchmarks in dimensions $D = 2, 4, 8, 16, 32, 64, 128, 256$.

**Results.** Among FM-based methods, OFM with any plan demonstrates the best results in all dimensions. For all 3 plans, OFM convergences to close final solutions and metrics. Moreover, minibatch OT plan achieves better results, especially in high dimensions.

MMv1 beats OFM, since it is designed to solve OT via simple map, while OFM deals with vector fields and have to process whole time interval $[0, 1]$.

RF demonstrates worse performance than even linear baseline, because it is not designed to solve OT. In this sense, $c$-RF works much better, but rapidly deteriorates with increasing dimensions. OT-CFM converges to biased solution, which is especially noticeable in high dimensions. But due to large batchsize this bias is still better than $c-$RF with its error.

### 4.1 Unpaired Image-to-image Transfer

Another task that involves learning a translation between two distributions is unpaired image-to-image translation [40]. We follow the setup of [19] where translation is computed in the 512 dimensional latent space of the pre-trained ALAE autoencoder [29] on $1024 \times 1024$ FFHQ dataset [16].

In this setup, we compare our OFM with independent (Ind) and minibatch OT (MB) plans, OT-CFM and RF. Qualitative results are presented in Fig **??**.

Our OFM converges to nearly the same solution for both ID and MB plans and demonstrates good results.

## 5 Discussion

**Potential impact.** We believe that our novel theoretical results have a huge potential for improving modern straightening methods and inspiring the community for further studies. The direct connection with well-studied Optimal Transport may result in adopting of OT's strong sides to Flow Matching and deepening the understanding of it.

**Limitations** discussion is located in Appendix 6.2.

## References

[1] SN Afriat. Theory of maxima and the method of lagrange. *SIAM Journal on Applied Mathematics*, 20(3):343–357, 1971.

[2] Brandon Amos. On amortizing convex conjugates for optimal transport. In *The Eleventh International Conference on Learning Representations*, 2022.

[3] Brandon Amos, Lei Xu, and J Zico Kolter. Input convex neural networks. In *International Conference on Machine Learning*, pages 146–155. PMLR, 2017.

[4] Jean-David Benamou and Yann Brenier. A computational fluid mechanics solution to the monge-kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, 2000.

[5] Espen Bernton, Pierre E Jacob, Mathieu Gerber, and Christian P Robert. On parameter estimation with the wasserstein distance. *Information and Inference: A Journal of the IMA*, 8(4):657–676, 2019.

[6] Charlotte Bunne, Andreas Krause, and Marco Cuturi. Supervised training of conditional monge maps. *Advances in Neural Information Processing Systems*, 35:6859–6872, 2022.

[7] Hanqun Cao, Cheng Tan, Zhangyang Gao, Yilun Xu, Guangyong Chen, Pheng-Ann Heng, and Stan Z Li. A survey on generative diffusion models. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[8] Shreyas Chaudhari, Srinivasa Pranav, and José MF Moura. Gradient networks. *arXiv preprint arXiv:2404.07361*, 2024.

[9] Yize Chen, Yuanyuan Shi, and Baosen Zhang. Optimal control via neural networks: A convex approach. *arXiv preprint arXiv:1805.11835*, 2018.

[10] Yongxin Chen, Tryphon T Georgiou, and Michele Pavon. On the relation between optimal transport and schrödinger bridges: A stochastic control viewpoint. *Journal of Optimization Theory and Applications*, 169:671–691, 2016.

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

[12] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. *arXiv preprint arXiv:2403.03206*, 2024.

[13] Nikita Gushchin, Sergei Kholkin, Evgeny Burnaev, and Alexander Korotin. Light and optimal schr\" odinger bridge matching. *arXiv preprint arXiv:2402.03207*, 2024.

[14] J Hiriart-Urruty and Yves Lucet. Parametric computation of the legendre-fenchel conjugate with application to the computation of the moreau envelope. *Journal of Convex Analysis*, 14(3):657, 2007.

[15] Pieter-Jan Hoedt and Günter Klambauer. Principled weight initialisation for input-convex neural networks. *Advances in Neural Information Processing Systems*, 36, 2024.

[16] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[17] Alexander Korotin, Vage Egiazarian, Arip Asadulaev, Alexander Safin, and Evgeny Burnaev. Wasserstein-2 generative networks. *arXiv preprint arXiv:1909.13082*, 2019.

[18] Alexander Korotin, Vage Egiazarian, Lingxiao Li, and Evgeny Burnaev. Wasserstein iterative networks for barycenter estimation. *Advances in Neural Information Processing Systems*, 35:15672–15686, 2022.

[19] Alexander Korotin, Nikita Gushchin, and Evgeny Burnaev. Light schrödinger bridge. In *The Twelfth International Conference on Learning Representations*, 2023.

[20] Alexander Korotin, Lingxiao Li, Aude Genevay, Justin M Solomon, Alexander Filippov, and Evgeny Burnaev. Do neural optimal transport solvers work? a continuous wasserstein-2 benchmark. *Advances in neural information processing systems*, 34:14593–14605, 2021.

[21] Alexander Korotin, Lingxiao Li, Justin Solomon, and Evgeny Burnaev. Continuous wasserstein-2 barycenter estimation without minimax optimization. *arXiv preprint arXiv:2102.01752*, 2021.

[22] Maciej Ławryńczuk. Input convex neural networks in nonlinear predictive control: A multi-model approach. *Neurocomputing*, 513:273–293, 2022.

[23] Christian Léonard. A survey of the schr\" odinger problem and some of its connections with optimal transport. *arXiv preprint arXiv:1308.0215*, 2013.

[24] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

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

[26] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

[27] Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, and Qiang Liu. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. *arXiv preprint arXiv:2309.06380*, 2023.

[28] Ashok Makkuva, Amirhossein Taghvaei, Sewoong Oh, and Jason Lee. Optimal transport mapping via input convex neural networks. In *International Conference on Machine Learning*, pages 6672–6681. PMLR, 2020.

[29] Stanislav Pidhorskyi, Donald A Adjeroh, and Gianfranco Doretto. Adversarial latent autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14104–14113, 2020.

[30] 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, 2023.

[31] Jack Richter-Powell, Jonathan Lorraine, and Brandon Amos. Input convex gradient networks. *arXiv preprint arXiv:2111.12187*, 2021.

[32] Yuyang Shi, Valentin De Bortoli, Andrew Campbell, and Arnaud Doucet. Diffusion schrödinger bridge matching. *Advances in Neural Information Processing Systems*, 36, 2024.

[33] Jan Stanczuk, Christian Etmann, Lisa Maria Kreusser, and Carola-Bibiane Schönlieb. Wasserstein gans work because they fail (to approximate the wasserstein distance). *arXiv preprint arXiv:2103.01678*, 2021.

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

[35] Alexander Tong, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Kilian Fatras, Guy Wolf, and Yoshua Bengio. Conditional flow matching: Simulation-free dynamic optimal transport. *arXiv preprint arXiv:2302.00482*, 2(3), 2023.

[36] Nazarii Tupitsa, Pavel Dvurechensky, Darina Dvinskikh, and Alexander Gasnikov. Numerical methods for large-scale optimal transport. *arXiv preprint arXiv:2210.11368*, 2022.

[37] Bryan Van Scoy, Randy A Freeman, and Kevin M Lynch. The fastest known globally convergent first-order method for minimizing strongly convex functions. *IEEE Control Systems Letters*, 2(1):49–54, 2017.

[38] Cédric Villani. *Topics in optimal transportation*, volume 58. American Mathematical Soc., 2021.

[39] Shu Yang and B Wayne Bequette. Optimization-based control using input convex neural networks. *Computers & Chemical Engineering*, 144:107143, 2021.

[40] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

# 6 Appendix

## 6.1 Benchmark details

**Metrics.** To assess the quality of retrieved transport map $T$ between $p_0$ and $p_1$, we use *learned variance percentage* (UVP): $\mathcal{L}^2$-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.

To measure quality of retrieved linear trajectories, we calculate the *cosine similarity* between ground truth directions $T^* - \text{id}$ and obtained directions $T - \text{id}$, i.e.,

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

For good approximations the cosine metric is approaching 1.

We estimate $\mathcal{L}^2-\text{UVP}$ and $\cos$ metrics with $2^{14}$ samples from $p_0$. Solvers results for $\cos$ metric are presented in Table 2.

| Solver | **Solver type\|DIM** | $D=2$ | $D=4$ | $D=8$ | $D=16$ | $D=32$ | $D=64$ | $D=128$ | $D=256$ |
|---|---|---|---|---|---|---|---|---|---|
| MMv1 | OT solver | 0.99 | 0.99 | 0.99 | 0.99 | 0.98 | 0.99 | 0.99 | 0.99 |
| OFM Ind 1024 **(Ours)** | | 0.99 | – | – | 0.98 | – | 0.96 | 0.94 | – |
| OFM MB 64 **(Ours)** | | – | – | – | – | – | – | – | – |
| OFM MB 1024 **(Ours)** | | 0.99 | – | – | 0.98 | – | 0.964 | 0.97 | – |
| OT-CFM MB 64 | Flow Matching | 0.99 | 0.981 | 0.971 | 0.961 | 0.972 | 0.95 | 0.944 | 0.9 |
| OT-CFM MB 1024 | | 0.99 | 0.985 | 0.978 | 0.968 | 0.975 | 0.96 | 0.949 | 0.915 |
| $c$-RF | | 0.989 | 0.83 | 0.83 | 0.78 | 0.778 | 0.762 | 0.748 | 0.73 |
| RF | | 0.87 | 0.75 | 0.65 | 0.67 | 0.72 | 0.70 | 0.70 | 0.70 |
| Lin | Baseline | 0.75 | 0.80 | 0.73 | 0.73 | 0.76 | 0.75 | 0.77 | 0.77 |

Table 2: $\cos$ values of solvers fitted on high-dimensional benchmarks in dimensions $D = 2, 4, 8, 16, 32, 64, 128, 256$.

**Implementation details.** All solvers run approximately 200.000 iterations of RMSprop optimizer with the learning rate $10^{-3} - 10^{-4}$. In RF and $c$-RF, ODE are solved via Explicit Runge-Kutta method of order $5(4)$ [11] with absolute tolerance $10^{-4} - 10^{-6}$. ICNNs or NNs architectures of the solvers are designed to have the same number of learnable parameters, in particular, we use Fully Connected layers with hidden sizes $[128, 128, 64, 64]$. All algorithms converge in several hours on the single GPU.

## 6.2 Limitations

**Flow map inversion.** During training, we need to compute $(\phi_t^{\Psi_\theta})^{-1}(\cdot)$ via solving strongly convex subproblem (20). In practice, we approach it by the standard gradient descent (with Adam optimizer), but actually there exist many improved methods to solve such conjugation problems more effectively in both the optimization [37, 14] and OT [2, 28]. This provides a dozen of opportunities for improvement, but leave such advanced methods for future research.

**ICNNs.** It is known that ICNNs may underperform compared to regular neural networks [20, 18]. Thus, ICNN parametrization may limit the performance of our OFM. Fortunately, deep learning community actively study ways to improve ICNNs [8, 6, 31, 15] due to their growing popularity in various tasks [39, 22, 9]. We believe that the really expressive ICNN architectures are yet to come.

**Hessian inversion.** We get the gradient of our OFM loss via formula from Proposition 4. There we have to invert the hessian $\nabla^2\Psi(\cdot)$, which is expensive. We point to addressing this limitation as a promising avenue for future studies.

## 6.3 Proofs

**Proposition 1 (Simplified OFM Loss)**

$$\mathcal{L}^{\pi}_{OFM}(\Psi) = \int_0^1 \left\{ \int_{\mathbb{R}^D \times \mathbb{R}^D} \left\| \frac{(\phi_t^{\Psi})^{-1}(x_t) - x_0}{t} \right\|^2 \pi(x_0, x_1) dx_0 dx_1 \right\} dt, x_t = (1-t)x_0 + tx_1.$$

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

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

For fixed points $x_0, x_1$ and time $t$ in integrand, we find a point $z_0=(\phi_t^{\Psi})^{-1}(x_t)$ such that in moment $t\in[0,1]$ it is transported to point $x_t=(1-t)x_0+tx_1$. This point $z_0$ satisfies equality

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

We define the vector field $u_t^{\Psi}$ as

$$u_t^{\Psi}(x_t) \quad = \quad \nabla\Psi(z_0)-z_0=\frac{x_t-z_0}{t}.$$

Putting $u_t^{\Psi}(x_t)$ in the integrand of (21), we obtain simplified integrand

$$\begin{aligned}\|x_1-x_0-u_t^{\Psi}(x_t)\|^2 &= \left\|x_1-x_0-\left(\frac{x_t-z_0}{t}\right)\right\|^2\\ &= \frac{1}{t^2}\|tx_1-tx_0-((1-t)x_0+tx_1)+z_0\|^2\\ &= \frac{1}{t^2}\|z_0-x_0\|^2=\left\|\frac{(\phi_t^{\Psi})^{-1}(x_t)-x_0}{t}\right\|^2.\end{aligned}$$

■

**Lemma 1 (Main Integration Lemma)** *For any two points $x_0, x_1\in\mathbb{R}^D$ and a convex function $\Psi$, the following equality holds true:*

$$\int_0^1\|u_t^{\Psi}(x_t)-(x_1-x_0)\|^2dt=2\cdot[\Psi(x_0)+\overline{\Psi}(x_1)-\langle x_0,x_1\rangle]. \tag{21}$$

**Proof.** In order to find a point $z_0(t)=z_0=(\phi_t^{\Psi})^{-1}(x_t)$ for fixed $x_0, x_1$ such that in moment $t\in(0,1)$ it is transported to point $x_t=(1-t)x_0+tx_1$ we need to satisfy equality

$$x_t=t\nabla\Psi(z_0)+(1-t)z_0. \tag{22}$$

We use the simplified loss form from Proposition 1, i.e.,

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

Next, we integrate (21) w.r.t. time $t$ from 0 excluding to 1 excluding (This exclusion does not change the integral). We notice, that set of points $z_0(t)=(\phi_t^{\Psi})^{-1}(x_t), t\in(0,1)$ forms a curve in $\mathbb{R}^D$ with parameter $t$, and one can integrate along it. The limits of integration along the curve $z_0(t)$ are

$$z_0(t)|_{t=0}=x_0,\quad z_0(t)|_{t=1}=\nabla\overline{\Psi}(x_1). \tag{23}$$

Further, we change the time variable $t$ to $s=\frac{1}{t}, ds=-\frac{dt}{t^2}$ and get

$$\int_0^1\frac{1}{t^2}\|z_0(t)-x_0\|^2dt=-\int_{+\infty}^1\|z_0(s)-x_0\|^2ds=-\int_{+\infty}^1\langle z_0(s)-x_0,z_0(s)-x_0\rangle ds. \tag{24}$$

In the condition (22), we also consider the change

$$\begin{aligned}x_t=t\nabla\Psi(z_0)+(1-t)z_0 &= (1-t)x_0+tx_1,\\ t(\nabla\Psi(z_0)-x_1) &= (1-t)(x_0-z_0),\\ (\nabla\Psi(z_0)-x_1) &= \left(\frac{1}{t}-1\right)(x_0-z_0),\\ (\nabla\Psi(z_0)-x_1) &= (s-1)(x_0-z_0).\end{aligned}$$

We make one more substation from $s$ to $s' = \frac{1}{s-1}, ds' = -\frac{ds}{(s-1)^2} = -(s')^2 ds$ and obtain

$$\begin{aligned}(\nabla\Psi(z_0) - x_1) &= (s-1)(x_0 - z_0),\\ (\nabla\Psi(z_0) - x_1) &= \frac{(x_0 - z_0)}{s'},\\ s'(\nabla\Psi(z_0) - x_1) &= (x_0 - z_0).\end{aligned} \tag{25}$$

The integral (24) changes as

$$\begin{aligned}-\int_{+\infty}^{1} \|x_0 - z_0(s)\|^2 ds &= \int_0^{\infty} \left\langle \frac{x_0 - z_0(s')}{s'}, \frac{x_0 - z_0(s')}{s'} \right\rangle ds'\\ &= \int_0^{\infty} \langle \nabla\Psi(z_0(s')) - x_1, \nabla\Psi(z_0(s')) - x_1 \rangle \, ds'.\end{aligned}$$

In order to eliminate differential $ds'$, we take differential from both sides of (25) w.r.t. $s'$

$$\begin{aligned}d[(\nabla\Psi(z_0) - x_1)s'] &= d[x_0 - z_0],\\ s'\nabla^2\Psi(z_0)dz_0 + (\nabla\Psi(z_0) - x_1)ds' &= -dz_0,\\ (\nabla\Psi(z_0) - x_1)ds' &= -(s'\nabla^2\Psi(z_0) + I)dz_0.\end{aligned}$$

Next, we continue

$$\begin{aligned}\|x_1 - x_0 - u_t^{\Psi}(x_t)\|^2 &= \int_0^{\infty} \langle \nabla\Psi(z_0) - x_1, \nabla\Psi(z_0) - x_1 \rangle \, ds'\\ &= \int_{z_0} \langle x_1 - \nabla\Psi(z_0), (s'\nabla^2\Psi(z_0) + I)dz_0 \rangle\\ &= \int_{z_0} \langle x_1 - \nabla\Psi(z_0), dz_0 \rangle + \int_{z_0} \langle s'(x_1 - \nabla\Psi(z_0)), \nabla^2\Psi(z_0)dz_0 \rangle\\ &\overset{(25)}{=} \int_{z_0} \langle x_1 - \nabla\Psi(z_0), dz_0 \rangle + \int_{z_0} \langle z_0 - x_0, \nabla^2\Psi(z_0)dz_0 \rangle.\end{aligned} \tag{26}$$

We notice that

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

As a consequence, we write down

$$\begin{aligned}(26) &= \int_{z_0} \langle x_1 - \nabla\Psi(z_0), dz_0 \rangle + \int_{z_0} \langle z_0 - x_0, \nabla^2\Psi(z_0)dz_0 \rangle\\ &= \int_{z_0} \langle x_1, dz_0 \rangle - \int_{z_0} \langle \nabla\Psi(z_0), dz_0 \rangle\\ &+ \int_{z_0} d\langle z_0, \nabla\Psi(z_0)\rangle - \int_{z_0} \langle \nabla\Psi(z_0), dz_0\rangle - \int_{z_0} \langle x_0, \nabla^2\Psi(z_0)dz_0\rangle\\ &= \int_{z_0} \langle x_1, dz_0 \rangle - 2\int_{z_0} \langle \nabla\Psi(z_0), dz_0 \rangle + \int_{z_0} d\langle z_0, \nabla\Psi(z_0)\rangle - \int_{z_0} \langle x_0, \nabla^2\Psi(z_0)dz_0\rangle.\end{aligned} \tag{27}$$

Under all integrals we have closed form differentials

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

We integrate them from initial point $x_0$ to the final $\nabla\overline{\Psi}(x_1)$ according to limits (23) and get

$$\begin{aligned}(27) &= \int_{z_0} d\langle x_1, z_0\rangle - 2\int_{z_0} d\Psi(z_0) + \int_{z_0} d\langle z_0, \nabla\Psi(z_0)\rangle - \int_{z_0} d\langle x_0, \nabla\Psi(z_0)\rangle\\ &= \langle x_1, \nabla\overline{\Psi}(x_1)\rangle - \langle x_1, x_0\rangle + 2(\Psi(x_0) - \Psi(\nabla\overline{\Psi}(x_1))) + \langle(\nabla\overline{\Psi}(x_1), \nabla\Psi(\nabla\overline{\Psi}(x_1))\rangle\\ &- \langle x_0, \nabla\Psi(x_0)\rangle + \langle x_0, \nabla\Psi(x_0)\rangle - \langle x_0, \nabla\Psi(\nabla\overline{\Psi}(x_1))\rangle. \qquad (28)\end{aligned}$$

Finally, we use properties of conjugate functions:

- Invertability:

$$\nabla\Psi(\nabla\overline{\Psi}(x_1)) = \nabla\Psi(\nabla\Psi^{-1}(x_1)) = x_1, \quad \forall x_1 \in \mathbb{R}^D,$$

- Fenchel-Young's equality:

$$\Psi(\nabla\overline{\Psi}(x_1)) + \overline{\Psi}(x_1) = \langle\nabla\overline{\Psi}(x_1), x_1\rangle, \quad \forall x_1 \in \mathbb{R}^D.$$

We simplify (28) to

$$\begin{aligned}(28) &= \langle x_1, \nabla\overline{\Psi}(x_1)\rangle - \langle x_1, x_0\rangle + 2(\Psi(x_0) + \overline{\Psi}(x_1) - \langle\nabla\overline{\Psi}(x_1), x_1\rangle) + \langle(\nabla\overline{\Psi}(x_1), x_1\rangle\\ &- \langle x_0, \nabla\Psi(x_0)\rangle + \langle x_0, \nabla\Psi(x_0)\rangle - \langle x_0, x_1\rangle\\ &= 2[\Psi(x_0) + \overline{\Psi}(x_1) - \langle x_0, x_1\rangle].\end{aligned}$$

■

**Theorem 1 (OFM and OT connection)** *Let us 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, losses $\mathcal{L}_{OT}(\Psi)$ and $\mathcal{L}^{\pi}_{OFM}(\Psi)$, defined in (4) and (16), respectively, have the same minimizers, i.e.,*

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

**Proof.** Main Integration Lemma 1 states that for any fixed points $x_0, x_1$ we 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 $\pi$ (integration w.r.t. points $x_0, x_1 \sim \pi$ ) gives

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

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

**Proposition 2 (Intractable Distance)** *The distance (19) between an arbitrary vector field $u$ and OT field $u^*$ equals to the FM loss from (10) with the optimal plan $\pi^*$, i.e.,*

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

**Proof.** We recall the definitions of $\text{DIST}(u, u^*)$ (19) and FM loss $\mathcal{L}_{FM}^{\pi^*}(u)$ (10):

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

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

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

$$\int_{\mathbb{R}^D \times \mathbb{R}^D} \|u_t(x_t) - (x_1 - x_0)\|^2 \pi^*(x_0, x_1) dx_0 dx_1 = \int_{\mathbb{R}^D} \|u_t(x_t) - (\nabla\Psi^*(x_0) - x_0)\|^2 p_0(x_0) dx_0.$$

$$x_t = (1-t)x_0 + t\nabla\Psi^*(x_0). \tag{29}$$

We notice that dynamic OT vector field $u^* = u^{\Psi^*}$ is the optimal one with potential $\Psi^*$. Moreover, for any point $x_t = (1-t)x_0 + t\nabla\Psi^*(x_0)$ generated by $u^*$, we can calculate $u_t^*(x_t) = u_t^{\Psi^*}(x_t) = \nabla\Psi^*(x_0) - x_0$. It is the same expression as from (29), i.e.,

$$(29) = \int_{\mathbb{R}^D} \|u_t(x_t) - (\nabla\Psi^*(x_0) - x_0)\|^2 p_0(x_0) dx_0$$

$$= \int_{\mathbb{R}^D} \|u_t(x_t) - u_t^*(x_t)\|^2 p_0(x_0) dx_0. \tag{30}$$

Finally, we 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, we obtain the result

$$\int_{\mathbb{R}^D \times \mathbb{R}^D} \|u_t(x_t) - (x_1 - x_0)\|^2 \pi^*(x_0, x_1) dx_0 dx_1 = \int_{\mathbb{R}^D} \|u_t(x_t) - u_t^*(x_t)\|^2 p_t^*(x_t) dx_t.$$

Hence, the integration over time $t$ gives the desired equality

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

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

**Proposition 3 (Tractable Distance For OFM)** *The distance* $\text{DIST}(u^\Psi, u^{\Psi^*})$ *between an* ***optimal*** *vector field* $u^\Psi$ *generated by a convex function* $\Psi$ *and the vector field* $u^{\Psi^*}$ *with the Brenier potential* $\Psi^*$ *can be evaluated directly via OFM loss from* (16) *and* ***any*** *plan* $\pi$*:*

$$\text{DIST}(u^\Psi, u^{\Psi^*}) = \mathcal{L}_{OFM}^{\pi}(\Psi) - \mathcal{L}_{OFM}^{\pi}(\Psi^*). \tag{31}$$

**Proof.** For the vector field $u^\Psi$, we apply the formula for intractable distance from Proposition 3, i.e,

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

According to Main Integration Lemma 1, for any plan $\pi$ and convex function $\Psi$, we have 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 $\textsc{Const}'(\pi)$ does not depend on $\Psi$, we have the same constant with $\Psi = \Psi^*$ and can eliminate it, i.e.,

$$\begin{aligned} \mathcal{L}^{\pi}_{OFM}(\Psi) &= 2 \cdot \mathcal{L}_{OT}(\Psi) - \textsc{Const}'(\pi) \\ \mathcal{L}^{\pi}_{OFM}(\Psi^*) &= 2 \cdot \mathcal{L}_{OT}(\Psi^*) - \textsc{Const}'(\pi) \\ &\Downarrow \\ \mathcal{L}^{\pi}_{OFM}(\Psi) - \mathcal{L}^{\pi}_{OFM}(\Psi^*) &= 2 \cdot \mathcal{L}_{OT}(\Psi) - 2 \cdot \mathcal{L}_{OT}(\Psi^*). \end{aligned}$$

The right part of (32) does not depend on a plan $\pi$, thus, the left part is invariant for any plan including optimal plan $\pi^*$, i.e.,

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

■

**Proposition 4 (Explicit Loss Gradient Formula)** *The gradient of $\mathcal{L}^{\pi}_{OFM}$ can be calculated as*

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

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

**Proof.** Point $z_0 = (\phi_t^{\Psi_\theta})^{-1}(x_t)$ now depends on parameters $\theta$. We differentiate the integrand from the simplified OFM loss (17) for fixed points $x_0, x_1$ and time $t$, i.e.,

$$d\left(\frac{1}{t^2}\|z_0 - x_0\|^2\right) = 2\left\langle \frac{z_0 - x_0}{t^2}, \frac{dz_0}{d\theta}d\theta \right\rangle. \tag{32}$$

For point $z_0$, the equation (22) holds true:

$$x_t = (1-t)z_0 + t\nabla\Psi_\theta(z_0). \tag{33}$$

We differentiate (33) w.r.t. $\theta$ and obtain

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

Therefore, we have

$$\begin{aligned} (32) &= \left\langle 2\frac{x_0 - z_0}{t}, \left(t\nabla^2\Psi_\theta(z_0) + (1-t)I\right)^{-1}\frac{\partial\nabla\Psi_\theta}{\partial\theta}(z_0)d\theta \right\rangle \\ &= \left\langle 2\left(t\nabla^2\Psi_\theta(z_0) + (1-t)I\right)^{-1}\frac{(x_0 - z_0)}{t}, \frac{\partial\nabla\Psi_\theta}{\partial\theta}(z_0)d\theta \right\rangle. \end{aligned} \tag{34}$$

Now the differentiation over $\theta$ is located only in the right part of (34) in the term $\frac{\partial\nabla\Psi_\theta}{\partial\theta}$. Hence, point $z_0$ and the left part of (34) can be considered as constants during differentiation. To get the gradient of OFM loss we also need to take math expectation over plan $\pi$ and time $t$.

■