META FLOW MATCHING: INTEGRATING VECTOR FIELDS ON THE WASSERSTEIN MANIFOLD

Anonymous authors
Paper under double-blind review

ABSTRACT

Numerous biological and physical processes can be modeled as systems of interacting entities evolving continuously over time, e.g. the dynamics of communicating cells or physical particles. Learning the dynamics of such systems is essential for predicting the temporal evolution of populations across novel samples and unseen environments. Flow-based models allow for learning these dynamics at the population level — they model the evolution of the entire distribution of samples. However, current flow-based models are limited to a single initial population and a set of predefined conditions which describe different dynamics. We argue that multiple processes in natural sciences have to be represented as vector fields on the Wasserstein manifold of probability densities. That is, the change of the population at any moment in time depends on the population itself due to the interactions between samples. In particular, this is crucial for personalized medicine where the development of diseases and their respective treatment response depends on the microenvironment of cells specific to each patient. We propose *Meta Flow Matching* (MFM), a practical approach to integrate along these vector fields on the Wasserstein manifold by amortizing the flow model over the initial populations. Namely, we embed the population of samples using a Graph Neural Network (GNN) and use these embeddings to train a Flow Matching model. This gives MFM the ability to generalize over the initial distributions unlike previously proposed methods. We demonstrate the ability of MFM to improve prediction of individual treatment responses on a large scale multi-patient single-cell drug screen dataset.

1 INTRODUCTION

Understanding the dynamics of many-body problems is a focal challenge across the natural sciences. In the field of cell biology, a central focus is the understanding of the dynamic processes that cells undergo in response to their environment, and in particular their response and interaction with other cells. Cells communicate with one other in close proximity using *cell signaling*, exerting influence over each other's trajectories (Armingol et al., 2020; Goodenough and Paul, 2009). This signaling presents an obstacle for modeling due to the complex nature of intercellular regulation, but is essential for understanding and eventually controlling cell dynamics during development (Gulati et al., 2020; Rizvi et al., 2017), in diseased states (Molè et al., 2021; Binnewies et al., 2018; Zeng and Dai, 2019; Chung et al., 2017), and in response to perturbations (Ji et al., 2021; Peidli et al., 2024).

The exponential decrease of sequencing costs and advances in microfluidics has enabled the rapid advancement of single-cell sequencing and related technologies over the past decade (Svensson et al., 2018). While single-cell sequencing has been used to great effect to understand the heterogeneity in cell systems, it is also destructive, making longitudinal measurements extremely difficult. Hence, learning dynamical system models of cells while also capturing their inherent heterogeneity and stochasticity of cellular systems remains a central challenge in biology.

Instead, most existing approaches model cell dynamics at the population level (Hashimoto et al., 2016; Weinreb et al., 2018; Schiebinger et al., 2019; Tong et al., 2020; Neklyudov et al., 2022; Bunne et al., 2023a). These approaches involve the formalisms of optimal transport (Villani, 2009; Peyré and Cuturi, 2019), diffusion (De Bortoli et al., 2021), or normalizing flows (Lipman et al., 2023), which learn a map between empirical measures. While these methods are able to model the dynamics of the population, they are fundamentally limited in that they model the evolution of cells as independent particles evolving according to the learned model of a global vector field. Furthermore, these models can be trained to match any given set of measures, but they are restricted to modeling of a single population and can at best condition on a number of different dynamics that are in the training data.

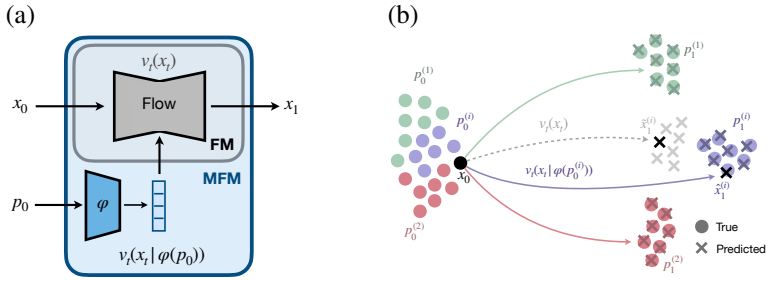

Figure 1: **Illustration of *Meta Flow Matching* (MFM, Eq. (17)).** (a) Comparison between *Flow Matching* (FM, Eq. (6)) and MFM. (b) Depiction of differences between MFM and FM generated predictions. Given a point x_t, a vector field (flow) model trained with MFM can generate different points $\hat{x}_1$ for different initial distributions p_0 (represented by red, green, and purple). FM trained models can only predict an aggregate response over populations (shown in gray). FM at best can incorporate known (seen) conditional information available in the training data, denoted as *Conditional Generative Flow Matching* (CGFM, Eq. (9)). In contrast, MFM jointly learns a population embedding model $\varphi(p_0)$ and a vector field v_t, allowing generalization to unseen populations.

We propose *Meta Flow Matching* (MFM) — the amortization of the Flow Matching generative modeling framework (Lipman et al., 2023) **over the input measures**, which (i) takes into account the interaction of particles (instead of modelling them independently) and (ii) allows for generalization of the learned model to previously unseen input populations. In practice, our method can be used to predict the time-evolution of distributions from a given dataset of the time-evolved examples. Namely, we assume that the collected data undergoes a universal developmental process, which depends only on the population itself as in the setting of the interacting particles or communicating cells. Under this assumption, we learn the vector field model that takes samples from the initial distribution as input and defines the push-forward map on the sample-space that maps the initial distribution to the final distribution (see Fig. 1).

We showcase the utility of our approach on two applications. To illustrate the intuition of the proposed method, we first test MFM on a synthetic task of "letter denoising". We show that MFM is able to generalize the denoising process to unseen letter silhouettes, whereas the standard FM approach cannot. Next, we explore how MFM can be applied to model single-cell perturbation data (Ji et al., 2021; Peidli et al., 2024). We evaluate MFM on predicting the response of patient-derived cells to chemotherapy treatments in a recently published large scale single-cell drug screening dataset where there are known to be patient-specific responses (Ramos Zapatero et al., 2023).[1] This is a challenging task due to the inherent variation that exist across patients, the variability induced by different treatments, and variation due to local cell compositions. Addressing this problem can lead to better prediction of tumor growth and therapeutic response to cancer treatment. We demonstrate that MFM can successfully predict the development of cell populations on replicated experiments, and most importantly, that it generalizes to previously unseen patients, thus, capturing the patient-specific response to the treatment.

2 Background

2.1 Generative Modeling via Flow Matching

Flow Matching is an approach to generative modeling recently proposed independently in different works: Rectified Flows (Liu et al., 2022b), Flow Matching (Lipman et al., 2023), Stochastic Interpolants (Albergo and Vanden-Eijnden, 2022). A flow is a continuous interpolation between densities $p_0(x_0)$ and $p_1(x_1)$ in the sample space. That is, the sample from the intermediate density $p_t(x)$ is produced as follows

$$x_t = f_t(x_0, x_1), \quad (x_0, x_1) \sim \pi(x_0, x_1)\,, \tag{1}$$

$$\text{where} \int dx_1\ \pi(x_0, x_1) = p_0(x_0)\,, \quad \int dx_0\ \pi(x_0, x_1) = p_1(x_1)\,, \tag{2}$$

where f_t is the time-continuous interpolating function such that $f_{t=0}(x_0, x_1) = x_0$ and $f_{t=1}(x_0, x_1) = x_1$ (e.g. linearly between x_0 and x_1 with $f_t(x_0, x_1) = (1-t)\cdot x_0 + t \cdot x_1$);

[1]This dataset includes more than 25 million cells collected for ten patients over 2500 different experimental/treatment conditions.

$\pi(x_0, x_1)$ is the density of the joint distribution, which is usually taken as a distribution of independent random variables $\pi(x_0, x_1) = p_0(x_0)p_1(x_1)$, but can also be generalized to formulate the optimal transport problems (Pooladian et al., 2023; Tong et al., 2024a). The corresponding density can be defined then as the following expectation over the interpolating samples

$$p_t(x) = \int dx_0 dx_1 \; \pi(x_0, x_1)\delta(x - f_t(x_0, x_1)) \,. \tag{3}$$

The essential part of Flow Matching is the continuity equation that describes the change of this density through the vector field on the state space, which admits vector field $v_t^*(x)$ as a solution

$$\frac{\partial p_t(x)}{\partial t} = -\langle \nabla_x, p_t(x) v_t^*(x) \rangle \,, \quad v_t^*(\xi) = \frac{1}{p_t(\xi)} \mathbb{E}_{\pi(x_0,x_1)} \left[\delta(f_t(x_0, x_1) - \xi) \frac{\partial f_t(x_0, x_1)}{\partial t} \right] . \tag{4}$$

The expectation for v_t^* is intractable, but we can model $v_t(x; \omega)$ to approximate v_t^* so that we can efficiently generate new samples. Relying on Eq. (4), one can derive the tractable objective for learning $v_t^*(x)$, i.e.

$$\mathcal{L}_{\text{FM}}(\omega) = \int_0^1 dt \; \mathbb{E}_{p_t(x)} \| v_t^*(x) - v_t(x; \omega) \|^2 \tag{5}$$

$$= \mathbb{E}_{\pi(x_0,x_1)} \int_0^1 dt \; \left\| \frac{\partial}{\partial t} f_t(x_0, x_1) - v_t(f_t(x_0, x_1); \omega) \right\|^2 + \text{constant} \,. \tag{6}$$

Finally, the vector field $v_t(\xi, \omega) \approx v_t^*(\xi)$ defines the push-forward density that approximately matches $p_{t=1}$, i.e. $T_{\#} p_0 \approx p_{t=1}$, where T is the flow corresponding to vector field $v_t(\cdot, \omega)$ with parameters ω.

2.2 Conditional Generative Modeling via Flow Matching

Conditional image generation is one of the most common applications of generative models nowadays; it includes conditioning on the text prompts (Saharia et al., 2022b; Rombach et al., 2022) as well as conditioning on other images (Saharia et al., 2022a). To learn the conditional generative process with diffusion models, one merely has to pass the conditional variable (sampled jointly with the data point) as an additional input to the parametric model of the vector field. The same applies for the Flow Matching framework.

Conditional Generative Modeling via Flow Matching is independently introduced in several works (Zheng et al., 2023; Dao et al., 2023; Isobe et al., 2024) and it operates as follows. Consider a family of time-continuous densities $p_t(x_t \mid c)$, which corresponds to the distribution of the following random variable

$$x_t = f_t(x_0, x_1), \quad (x_0, x_1) \sim \pi(x_0, x_1 \mid c) \,. \tag{7}$$

For every c, the density $p_t(x_t \mid c)$ follows the continuity equation with the following vector field

$$v_t^*(\xi \mid c) = \frac{1}{p_t(\xi \mid c)} \mathbb{E}_{\pi(x_0,x_1)} \delta(f_t(x_0, x_1) - \xi) \frac{\partial f_t(x_0, x_1)}{\partial t} \,, \tag{8}$$

which depends on c. Thus, the training objective of the conditional model becomes

$$\mathcal{L}_{CGFM}(\omega) = \mathbb{E}_{p(c)} \mathbb{E}_{\pi(x_0, x_1 \mid c)} \int_0^1 dt \; \left\| \frac{\partial}{\partial t} f_t(x_0, x_1) - v_t(f_t(x_0, x_1) \mid c; \omega) \right\|^2 , \tag{9}$$

where, compared to the original Flow Matching formulation, we first have to sample c, then produce the samples from $p_t(x_t \mid c)$ and pass c as input to the parametric model of the vector field.

2.3 Modeling Process in Natural Sciences as Vector Fields on the Wasserstein Manifold

We argue that numerous biological and physical processes cannot be modeled via the vector field propagating the population samples independently. Thus, we propose to model these processes as families of conditional vector fields where we amortize the conditional variable by embedding the population via a Graph Neural Network (GNN).

To provide the reader with the necessary intuition, we are going to use the geometric formalism developed by Otto (2001) (see the complete discussion in Appendix A). That is, time-dependent densities $p_t(x_t)$ define absolutely-continuous curves on the 2-Wasserstein space of distributions

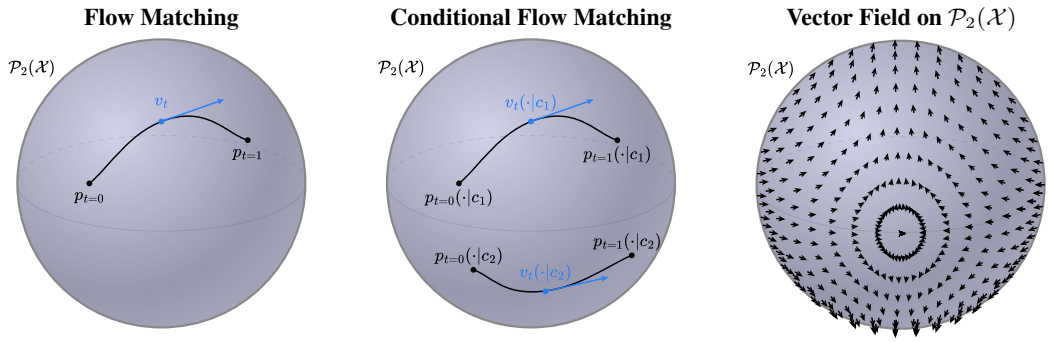

Figure 2: **Illustration of flow matching methods on the 2-Wasserstein manifold, $\mathcal{P}_2(\mathcal{X})$, depicted as a two-dimensional sphere.** *Flow Matching* learns the tangent vectors to a single curve on the manifold. *Conditional* generation corresponds to learning a finite set of curves on the manifold, e.g. classes c_1 and c_2 on the plot. *Meta Flow Matching* learns to integrate a vector field on $\mathcal{P}_2(\mathcal{X})$, i.e. for every starting density p_0, MFM defines a push-forward measure that integrates along the underlying vector field.

$\mathcal{P}_2(\mathcal{X})$ (see (Ambrosio et al., 2008), Chapter 8). The tangent space of this manifold is defined by the gradient flows $\mathcal{S}_t = \{\nabla s_t \,|\, s_t : \mathcal{X} \to \mathbb{R}\}$ on the state space $\mathcal{X}$, to be precise, its $L^2(\mu; \mathcal{X})$ closure $\forall \mu \in \mathcal{P}_2(\mathcal{X})$ (see (Ambrosio et al., 2008), eq. (8.0.2)). In the Flow Matching context, we are going to refer to the tangent vectors as vector fields since one can always project the vector field onto the tangent space by parameterizing it as a gradient flow (Neklyudov et al., 2022).

Under the geometric formalism of the 2-Wasserstein manifold, Flow Matching can be considered as learning the tangent vectors $v_t(\cdot)$ along the density curve $p_t(x_t)$ defined by the sampling process in Eq. (2) (see the left panel in Fig. 2). Furthermore, the conditional generation processes $p_t(x_t \,|\, c)$ would be represented as a finite set of curves if c is discrete (e.g. class-conditional generation of images) or as a family of curves if c is continuous (see the middle panel in Fig. 2).

Finally, one can define a vector field on the 2-Wasserstein manifold via the continuity equation with the vector field $v_t(x, p_t(x))$ on the state space $\mathcal{X}$ that depends on the current density $p_t(x)$ or its derivatives. Below we give two examples of processes defined as vector fields on the 2-Wasserstein manifold.

Example 1 (Mean-field limit of interacting particles)**.** *Consider a system of interacting particles, where the velocity of the particle at point x interacting with the particle at point y is defined as $k(x, y) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$. In the limit of the infinite number of particles one can describe their state using the density function $p_t(x)$. Then the change of the density is described by the following continuity equation*

$$\frac{dx}{dt} = \mathbb{E}_{p_t(y)} k(x, y), \quad \frac{\partial p_t(x)}{\partial t} = -\left\langle \nabla_x, p_t(x) \mathbb{E}_{p_t(y)} k(x, y) \right\rangle, \tag{10}$$

which is the first-order analog of the Vlasov equation (Jabin and Wang, 2016).

Example 2 (Diffusion)**.** *Even when the physical particles evolve independently in nature, the deterministic vector field model might be dependent on the current density of the population. For instance, for the diffusion process, the change of the density is described by the Fokker-Planck equation, which results in the density-dependent vector field when written as a continuity equation, i.e.*

$$\frac{\partial p_t(x)}{\partial t} = \frac{1}{2} \Delta_x p_t(x) = -\left\langle \nabla_x, p_t(x) \left(-\frac{1}{2} \nabla_x \log p_t(x) \right) \right\rangle \implies \frac{dx}{dt} = -\frac{1}{2} \nabla_x \log p_t(x) \,. \tag{11}$$

Motivated by the examples above, we argue that using the information about the current or the initial density is crucial for the modeling of time-evolution of densities in natural processes, to capture this type of dependency one can model the change of the density as the following Cauchy problem

$$\frac{\partial p_t(x)}{\partial t} = -\langle \nabla_x, p_t(x) v_t(x, p_t) \rangle \,, \quad p_{t=0}(x) = p_0(x) \,, \tag{12}$$

where the state-space vector field $v_t(x, p_t)$ depends on the density p_t.

The dependency might vary across models, e.g. in Example 1 the vector field can be modeled as an application of a kernel to the density function, while in Example 2 the vector field depends only on the local value of the density and its derivative.

3 Meta Flow Matching

In this paper, we propose the amortization of the Flow Matching framework over the marginal distributions. Our model is based on the outstanding ability of the Flow Matching framework to learn the push-forward map for any joint distribution $\pi(x_0, x_1)$ given empirically. For the given joint $\pi(x_0, x_1)$, we denote the solution of the Flow Matching optimization problem as follows

$$v_t^*(\cdot, \pi) = \arg\min_{v_t} \mathcal{L}_{GFM}(v_t(\cdot), \pi(x_0, x_1)) \,. \tag{13}$$

Analogous to amortized optimization (Chen et al., 2022; Amos et al., 2023), we aim to learn the model that outputs the solution of Eq. (13) based on the input data sampled from π, i.e.

$$v_t(\cdot, \varphi(\pi)) = v_t^*(\cdot, \pi) \,, \tag{14}$$

where $\varphi(\pi)$ is the embedding model of π and the joint density $\pi(\cdot \,|\, c)$ is generated using some unknown measure of the conditional variables $c \sim p(c)$.

3.1 Integrating Vector Fields on the Wasserstein Manifold via Meta Flow Matching

Consider the dataset of joint populations $\mathcal{D} = \{(\pi(x_0, x_1 \,|\, i))\}_i$, where, to simplify the notation, we associate every i-th population with its density $\pi(\cdot \,|\, i)$ and the conditioning variable here is the index of this population in the dataset. We make the following assumptions regarding the ground truth sampling process (i) we assume that the starting marginals $p_0(x_0 \,|\, i) = \int dx_1 \; \pi(x_0, x_1 \,|\, i)$ are sampled from some unknown distribution that can be parameterized with a large enough number of parameters (ii) the endpoint marginals $p_1(x_1 \,|\, i) = \int dx_0 \; \pi(x_0, x_1 \,|\, i)$ are obtained as push-forward densities solving the Cauchy problem in Eq. (12), (iii) there exists unique solution to this Cauchy problem.

One can learn a joint model of all the processes from the dataset $\mathcal{D}$ using the conditional version of the Flow Matching algorithm (see Section 2.2) where the population index i plays the role of the conditional variable. However, obviously, such a model will not generalize beyond the considered data $\mathcal{D}$ and unseen indices i. We illustrate this empirically in Section 5.

To be able to generalize to previously unseen populations, we propose learning the density-dependent vector field motivated by Eq. (12). That is, we propose to use an embedding function $\varphi : \mathcal{P}_2(\mathcal{X}) \to \mathbb{R}^m$ to embed the starting marginal density p_0, which we then input into the vector field model and minimize the following objective over ω

$$\mathcal{L}_{\text{MFM}}(\omega; \varphi) = \mathbb{E}_{i \sim \mathcal{D}} \mathbb{E}_{\pi(x_0, x_1 \,|\, i)} \int_0^1 dt \left\| \frac{\partial}{\partial t} f_t(x_0, x_1) - v_t(f_t(x_0, x_1) \,|\, \varphi(p_0); \omega) \right\|^2 . \tag{15}$$

Note that the initial density p_0 is enough to predict the push-forward density p_1 since the Cauchy problem for Eq. (12) has a unique solution. The embedding function $\varphi(p_0)$ can take different forms, e.g. it can be the density value $\varphi(p_0) = p_0(\cdot)$, which is then used inside the vector field model to evaluate at the current point (analogous to Example 2); a kernel density estimator (analogous to Example 1); or a parametric model taking the samples from this density as an input.

Proposition 1. *Meta Flow Matching recovers the Conditional Generation via Flow Matching when the conditional dependence of the marginals* $p_0(x_0 \,|\, c) = \int dx_1 \pi(x_0, x_1 \,|\, c)$ *and* $p_1(x_1 \,|\, c) = \int dx_0 \pi(x_0, x_1 \,|\, c)$ *and the distribution* $p(c)$ *are known, i.e. there exist* $\varphi : \mathcal{P}_2(\mathcal{X}) \to \mathbb{R}^m$ *such that* $\mathcal{L}_{MFM}(\omega) = \mathcal{L}_{CGFM}(\omega)$.

Proof. Indeed, sampling from the dataset $i \sim \mathcal{D}$ becomes sampling of the conditional variable $c \sim p(c)$ and the embedding function becomes $\varphi(p_0(\cdot \,|\, c)) = c$. □

Furthermore, for the parametric family of the embedding models $\varphi(p_t, \theta)$, we show that the parameters θ can be estimated by minimizing the objective in Eq. (15) in the joint optimization with the vector field parameters ω. We formalize this statement in the following theorem.

Algorithm 1: Meta Flow Matching (training)

Input : dataset of populations $\{(\pi(x_0, x_1 \mid i), c^i)\}_{i=1}^N$ and treatments c^i, and parametric models for the velocity, $v_t(\cdot;\omega)$, and population embedding $\varphi(\cdot;\theta)$.

for *training iterations* **do**

- $i \sim \mathcal{U}_{\{1,N\}}(i)$ // sample batch of n populations ids
- $(x_0^j, x_1^j, t^j) \sim \pi(x_0, x_1 \mid i)\mathcal{U}_{[0,1]}(t)$ // sample N_i particles for every population i
- $f_t(x_0^j, x_1^j) \leftarrow (1-t^j)x_0^j + t^j x_1^j$
- $h^i(\theta) \leftarrow \varphi\Big(\{x_0^j\}_{j=1}^{N_i};\theta\Big)$ // embed population $\{x_0^j\}_{j=1}^{N_i}$. For **CGFM** $h \leftarrow i$, **FM** $h \leftarrow \emptyset$.
- $\mathcal{L}_{\text{MFM}}(\omega,\theta) \leftarrow \frac{1}{n}\sum_i \frac{1}{n_i}\sum_j \left\| \frac{d}{dt} f_t(x_0^j, x_1^j) - v_{tj}\Big(f_t(x_0^j, x_1^j) \mid h^i(\theta), c^i; \omega\Big)\right\|^2$
- $\omega' \leftarrow \text{Update}(\omega, \nabla_\omega \mathcal{L}_{\text{MFM}}(\omega,\theta))$ // evaluate new parameters of the flow model
- $\theta' \leftarrow \text{Update}(\theta, \nabla_\theta \mathcal{L}_{\text{MFM}}(\omega,\theta))$ // evaluate new parameters of the embedding model
- $\omega \leftarrow \omega', \quad \theta \leftarrow \theta'$ // update both models

return $v_t(\cdot;\omega^*), \varphi(\cdot;\theta^*)$

Theorem 1. *Consider a dataset of populations $\mathcal{D} = \{(\pi(x_0, x_1 \mid i))\}_i$ generated from some unknown conditional model $\pi(x_0, x_1 \mid c)p(c)$. Then the following objective*

$$\mathcal{L}(\omega,\theta) = \mathbb{E}_{p(c)} \int_0^1 dt\, \mathbb{E}_{p_t(x_t \mid c)} \|v_t^*(x_t \mid c) - v_t(x_t \mid \varphi(p_0,\theta),\omega)\|^2 \tag{16}$$

is equivalent to the Meta Flow Matching objective

$$\mathcal{L}_{MFM}(\omega,\theta) = \mathbb{E}_{i\sim\mathcal{D}}\mathbb{E}_{\pi(x_0,x_1 \mid i)} \int_0^1 dt \left\| \frac{\partial}{\partial t} f_t(x_0,x_1) - v_t(f_t(x_0,x_1) \mid \varphi(p_0,\theta);\omega)\right\|^2 \tag{17}$$

up to an additive constant.

Proof. We postpone the proof to Appendix B. □

3.2 Learning Population Embeddings via Graph Neural Networks (GNNs)

In many applications, the populations $\mathcal{D} = \{(\pi(x_0, x_1 \mid i))\}_{i=1}^N$ are given as empirical distributions, i.e. they are represented as samples from some unknown density π

$$\{(x_0^j, x_1^j)\}_{j=1}^{N_i}, \quad (x_0^j, x_1^j) \sim \pi(x_0, x_1 \mid i)\,, \tag{18}$$

where N_i is the size of the i-th population. For instance, for the diffusion process considered in Example 2, the samples from $\pi(x_0, x_1 \mid i)$ can be generated by generating some marginal $p_1(x_1 \mid i)$ and then adding the Gaussian random variable to the samples x_1^j. We use this model in our synthetic experiments in Section 5.1.

Algorithm 2: Meta Flow Matching (sampling)

Input : initial population $\{x_0^j\}_{j=1}^{N'}$, treatment condition c^i, models $v_t(\cdot;\omega^*)$ and $\varphi(\cdot;\theta^*)$.

$h = \varphi\Big(\{x_0^j\}_{j=1}^{N'};\theta\Big)$ // embed the population

$x_1^j = \int_0^1 v_t(x_t^j \mid h, c^i;\omega)dt + x_0^j$ // ODE solver

return *predicted population* $\{x_1^j\}_{j=1}^{N'}$

Since the only available information about the populations is samples, we propose learning the embedding of populations via a parametric model $\varphi(p_0,\theta)$, i.e.

$$\varphi(p_0,\theta) = \varphi\Big(\{x_0^j\}_{j=1}^{N_i},\theta\Big), \quad (x_0^j, x_1^j) \sim \pi(x_0, x_1 \mid i)\,. \tag{19}$$

For this purpose, we employ GNNs, which recently have been successfully applied for simulation of complicated many-body problems in physics (Sanchez-Gonzalez et al., 2020). To embed a population $\{x_0^j\}_{j=1}^{N_i}$, we create a k-nearest neighbour graph G_i based on the metric in the state-space $\mathcal{X}$, input it into a GNN, which consists of several message-passing iterations (Gilmer et al., 2017) and the final average-pooling across nodes to produce the embedding vector. Finally, we update the parameters of

the GNN jointly with the parameters of the vector field to minimize the loss function in Eq. (17). We show pseudo-code for training in Algorithm 1 and sampling in Algorithm 2.

4 Related Work

***Meta* – amortizing learning over distributions** The meta-learning of probability measures was previously studied by Amos et al. (2022) where they demonstrated that the prediction of the optimal transport paths can be efficiently amortized over the input marginal measures. The main difference with our approach is that we are trying to learn the push-forward map without embedding the second marginal and without restricting ourselves to Input-Convex Neural Network (ICNN) by Amos et al. (2017) and optimal transport maps. MFM is *meta* in the same sense of amortizing optimal transport (or in our case flow) problems over multiple input distributions. Hence, we follow the naming convention of *Meta Optimal Transport* (Amos et al., 2022). This is different but related to how *meta* is used in the meta learning setting, which amortizes over learning problems (Hospedales et al., 2021; Achille et al., 2019).

Generative modeling for single cells Single cell data has expanded to encompass multiple modalities of data profiling cell state and activities (Frangieh et al., 2021; Bunne et al., 2023b). Single-cell data presents multiple challenges in terms of noise, non-time resolved, and high dimension, and generative models have been used to counter those problems. Autoencoder has been used to embed and extrapolate data Out Of Distribution (OOD) with its latent state dimension (Lotfollahi et al., 2019; Lopez et al., 2018; Hetzel et al., 2022). Orthogonal non-negative matrix factorization (oNMF) has also been used for dimensionality reduction combined with mixture models for cell state prediction (Chen et al., 2020). Other approaches have tried to use Flow Matching (FM) (Tong et al., 2024b;a; Neklyudov et al., 2023) or similar approaches such as the Monge gap (Uscidda and Cuturi, 2023) to predict cell trajectories. Currently, the state of the art method uses the principle of Optimal Transport (OT) to predict cell trajectories (Makkuva et al., 2020; Bunne et al., 2023b). These methods are based on input convex neural network (ICNN) architectures and can generalize out of distribution to a new cells. As of this time, our method is the only method that takes inter-cellular interactions into account and learns embeddings of entire cell populations to generalize across unseen distributions.

Generative modeling for physical processes The closest approach to ours is the prediction of the many-body interactions in physics (Sanchez-Gonzalez et al., 2020) via GNNs. However, the problem there is very different since these models use the information about the individual trajectories of samples, which are not available for the single-cell prediction. Liu et al. (2022a; 2024) consider a generalized form of Schrödinger bridges also with interacting terms, but do not consider the generalization to unseen distributions. Neklyudov et al. (2022) consider learning the vector field for any continuous time-evolution of a probability measure, however, their method is restricted to single curves and do not consider generalization to unseen data. Campbell et al. (2024) use input and space independent conditions for applications in co-protein design, but similarly, their method cannot generalize to novel distributions. Finally, the weather/climate forecast models generating the next state conditioned on the previous one (Price et al., 2023; Verma et al., 2024) are similar approaches to ours but operating on a much finer time resolution.

5 Experiments

To show the effectiveness of MFM to generalize under previously unseen populations for the population prediction task, we consider two experimental settings. (i) A synthetic experiment with well defined coupled populations, and (ii) experiments on a publicly available single-cell dataset consisting of populations from patient dependent treatment response trials. We parameterize all vector field models $v_t(\cdot \mid \varphi(p_0); \omega)$ using a Multi-Layer Perceptron (MLP). For MFM, we additionally parameterize $\varphi(p_t; \theta, k)$ using a Graph Convolutional Network (GCN) with a k-nearest neighbor graph edge pooling layer. We include details regarding model hyperparameters, training/optimization, and implementation in Appendix D and Appendix D.2. The results for all the models are averaged over 3 random seeds.

5.1 Synthetic Experiment

Synthetic data. We curate a synthetic dataset of the joint distributions $\{(p_0(x_0, \mid i), p_1(x_1 \mid i))\}_{i=1}^N$ by simulating a diffusion process applied to a set of pre-defined target distributions $p_1(x_1 \mid i)$ for $i = 1, \ldots, N$. To get a paired population $p_0(x_0 \mid i)$ we simulate the forward diffusion process without

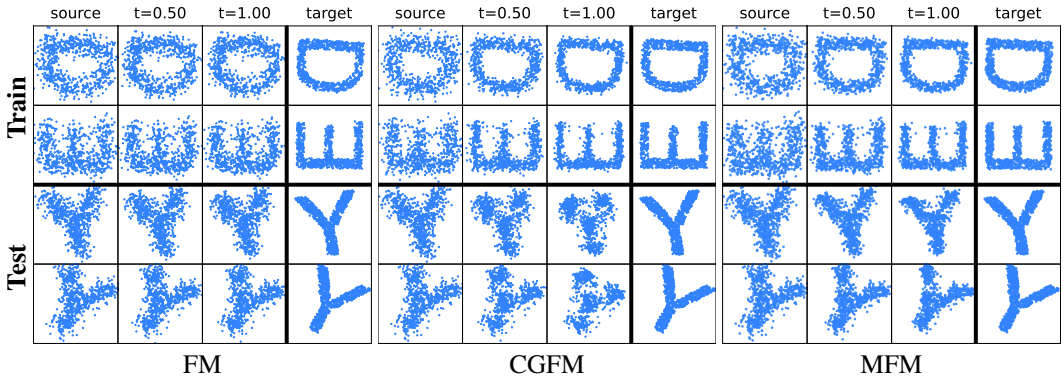

Figure 3: **Synthetic letters experiment visualizations.** Examples of model-generated samples from the source distribution ($t = 0$) to predicted target distribution ($t = 1$). See Fig. 6 Appendix F for further examples.

Table 1: **Results of the synthetic letters experiment for population prediction on seen train populations and unseen test populations.** We report the 1-Wasserstein ($\mathcal{W}_1$), 2-Wasserstein ($\mathcal{W}_2$), and the maximum-mean-discrepancy (MMD) distributional distances. We consider 4 settings for MFM with varying k. We use $^{\text{w/}}\mathcal{N}$ to denote models that use Gaussian source distributions sampled via $x_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$, but maintains the noisy letters source populations p_0 as input to the population embedding model $\varphi(p_0; \theta, k)$. Populations that comprise test sets (X and Y) are entirely unseen during training. We include an ablation on the effect of changing the number of training populations (Fig. 5) and results when using OT couplings between samples (Table 4) in Appendix F.

	Train			**Test** (X's)			**Test** (Y's)		
	$\mathcal{W}_1(\downarrow)$	$\mathcal{W}_2(\downarrow)$	MMD $(\times 10^{-3})$ $(\downarrow)$	$\mathcal{W}_1(\downarrow)$	$\mathcal{W}_2(\downarrow)$	MMD $(\times 10^{-3})$ $(\downarrow)$	$\mathcal{W}_1(\downarrow)$	$\mathcal{W}_2(\downarrow)$	MMD $(\times 10^{-3})$ $(\downarrow)$
FM	0.209 ± 0.000	0.277 ± 0.000	2.54 ± 0.00	0.234 ± 0.000	0.309 ± 0.000	2.45 ± 0.00	0.238 ± 0.000	0.316 ± 0.000	3.32 ± 0.01
FM$^{\text{w/}}\mathcal{N}$	0.806 ± 0.000	0.960 ± 0.000	31.68 ± 0.00	0.764 ± 0.000	0.931 ± 0.000	25.04 ± 0.00	1.030 ± 0.000	1.228 ± 0.000	45.36 ± 0.00
CGFM	0.090 ± 0.000	0.113 ± 0.000	0.25 ± 0.00	0.334 ± 0.000	0.407 ± 0.000	5.55 ± 0.00	0.327 ± 0.000	0.405 ± 0.000	6.85 ± 0.00
CGFM$^{\text{w/}}\mathcal{N}$	0.156 ± 0.025	0.201 ± 0.027	1.02 ± 0.39	0.849 ± 0.004	0.993 ± 0.003	35.08 ± 0.75	1.062 ± 0.011	1.229 ± 0.010	55.66 ± 0.76
MFM$^{\text{w/}}\mathcal{N}$ ($k = 0$)	0.148 ± 0.003	0.195 ± 0.010	0.94 ± 0.11	0.347 ± 0.011	0.431 ± 0.012	6.47 ± 0.44	0.402 ± 0.011	0.485 ± 0.010	10.92 ± 0.18
MFM$^{\text{w/}}\mathcal{N}$ ($k = 1$)	0.154 ± 0.004	0.208 ± 0.010	0.91 ± 0.01	0.349 ± 0.023	0.433 ± 0.023	6.53 ± 0.52	0.391 ± 0.035	0.477 ± 0.041	10.71 ± 1.86
MFM$^{\text{w/}}\mathcal{N}$ ($k = 10$)	0.151 ± 0.013	0.197 ± 0.015	0.94 ± 0.15	0.343 ± 0.020	0.427 ± 0.019	6.38 ± 0.67	0.413 ± 0.018	0.502 ± 0.024	11.93 ± 1.14
MFM$^{\text{w/}}\mathcal{N}$ ($k = 50$)	0.174 ± 0.005	0.232 ± 0.006	1.40 ± 0.13	0.363 ± 0.010	0.449 ± 0.013	7.46 ± 0.44	0.446 ± 0.021	0.536 ± 0.028	13.40 ± 0.23
MFM ($k = 0$)	$\mathbf{0.081 \pm 0.003}$	$\mathbf{0.100 \pm 0.004}$	$\mathbf{0.16 \pm 0.06}$	0.202 ± 0.002	0.249 ± 0.003	2.29 ± 0.05	0.218 ± 0.001	0.262 ± 0.002	3.79 ± 0.11
MFM ($k = 1$)	0.082 ± 0.001	0.101 ± 0.002	$\mathbf{0.16 \pm 0.01}$	0.205 ± 0.008	0.251 ± 0.008	2.38 ± 0.22	0.215 ± 0.006	0.258 ± 0.007	3.78 ± 0.25
MFM ($k = 10$)	0.088 ± 0.002	0.109 ± 0.003	0.21 ± 0.01	$\mathbf{0.201 \pm 0.006}$	$\mathbf{0.248 \pm 0.006}$	2.20 ± 0.15	0.208 ± 0.003	0.252 ± 0.002	3.55 ± 0.06
MFM ($k = 50$)	0.092 ± 0.004	0.116 ± 0.004	0.25 ± 0.06	0.206 ± 0.008	0.257 ± 0.008	$\mathbf{2.18 \pm 0.25}$	$\mathbf{0.204 \pm 0.005}$	$\mathbf{0.249 \pm 0.006}$	$\mathbf{3.14 \pm 0.18}$

drift $x_0 \sim \mathcal{N}(x_1, \sigma)$. After this setup, for reasonable values of σ, we assume that one can reverse the diffusion process and learn the push-forward map from $p_0(x_0 \mid i)$ to $p_1(x_1 \mid i)$ for every index i. For this task, given the i-th population index we denote $p_0(x_0 \mid i)$ as the *source* population $p_1(x_1 \mid i)$ as the i-th *target* population.

To construct $p_1(x_1 \mid i)$, we discretize samples from a defined silhouette; e.g. an image of a character, where i indexes the respective character. We use upper case letters as the silhouette and generate the corresponding samples $x_1 \sim p_1(x_1 \mid i)$ from the uniform distribution over the silhouette and run the diffusion process for samples x_1 to acquire x_0. We construct the *training data* using 10 random orientations of 24 letters. We construct the *test data* by using 10 random orientations of "X" and "Y". Test data populations of "X" and "Y" are entirely unseen during training.

We train FM, CGFM and 4 variants of MFM of varying k for the GCN population embedding model $\varphi(p_0; \theta, k)$. When $k = 0$, $\varphi(p_t; \theta, k)$ becomes identical to the DeepSets model (Zaheer et al., 2017). We compare MFM to FM and CGFM. We repeat this experiment for models trained with source distributions sampled from a standard normal $x_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$. We label these models as FM$^{\text{w/}}\mathcal{N}$, CGFM$^{\text{w/}}\mathcal{N}$, and MFM$^{\text{w/}}\mathcal{N}$. Here, MFM$^{\text{w/}}\mathcal{N}$ still takes the original p_0 as input to the population embedding model $\varphi(p_0; \theta, k)$, while $v_t(\cdot; \omega)$ uses $x_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$ as the source distribution.

MFM generalizes to populations from unseen letter silhouettes. FM does not have access to conditional information; hence will only learn an aggregated lens of the distribution dynamics and will not be able to fit the training data, and consequently won't generalize to the test conditions. For the training data, the CGFM vector field model takes in the distribution index i as a one-hot input condition. On the test set, since none of these indices is present, we input the normalized constant vector, which averages the learned embeddings of the indices. Because of this, CGFM will fit the training data, however, will not be able to generalize to the unseen condition in the test dataset. CGFM indicates when the model can fit the training data and demonstrates that the test data is substantially different and cannot be generated by the same model. For MFM, we expect to both fit the training data and generalize to unseen distributional conditions.

We report results for the synthetic experiment in Fig. 3 and Table 1. We observe that indeed FM struggles to adequately learn to sample from $p_1(x_1 \mid i)$ in the training set, while CGFM is able to effectively sample from $p_1(x_1 \mid i)$ in the training set. As expected, CGFM fits the training data, but struggles to generalize beyond its set of training conditions. In contrast, we see that MFM is able to both fit the training data while also generalizing to the unseen test distributions. Interestingly, although MFM performs better for certain values of k versus others, performance does not vary significantly for the range considered. In Fig. 7 (Appendix F.1) we plot the letter population embeddings.

5.2 Experiments on Organoid Drug-screen Data

Organoid drug-screen data. For experiments on biological data, we use the organoid drug-screen dataset from Ramos Zapatero et al. (2023). This dataset is a single-cell mass-cytometry dataset collected over 10 patients. Somewhat unique to this dataset, unlike many prior perturbation-screen datasets which have a single control population, this dataset has matched controls to each experimental condition. Populations from each patient are treated with 11 different drug treatments of varying dose concentrations.[2] We use the term *replicate* to define control-treatment population pairs, $p_0(x_0 \mid c_i)$ and $p_1(x_1 \mid c_i)$, respectively (see Section 5.2-left). After pre-processing, we acquire a total of 927 population pairs. In each patient, cell population are categorized into 3 cell *cultures*: (i) cancer associated Fibroblasts, (ii) patient-derived organoid cancer cells (PDO), and (iii) patient-derived organoid cancer cells co-cultured fibroblasts (PDOF). We report results for the individual cultures, i.e. Fibroblast, PDO, and PDOF cultures, in Appendix F.

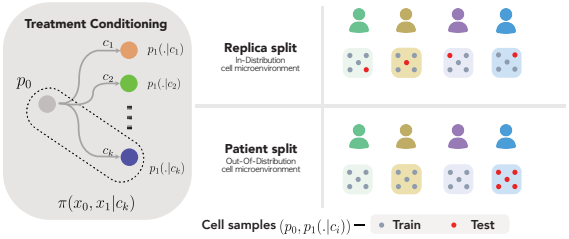

Figure 4: **Organoid drug-screen dataset overview.** (*Left*) a given replica consists of a control distribution p_0 and corresponding treatment response distribution p_1 for treatment condition c_i. (*Right*) train and test data splits for replicates (top) and patients (bottom). Experiments are conducted for 11 treatments, 10 patients, 3 culture conditions, and repeated (replicated) numerous times, resulting in a dataset of many control and treated population pairs.

Pre-processing and data splits. We filter each cell population to contain at least 1000 cells and consider 43 bio-markers. We consider two data splits for the organoid drug-screen dataset (see Section 5.2-right). ***Replicate split***; here we consider 2 settings, (i) leave-out replicates evenly across all patients for testing (*replica-1*), and (ii) leaving out a batch of replicates from a single patient while including the remaining replicas from the same patient for training (*replica-2*). The second setting is the ***Patients split***; here we leave-out replicates fully in one patients – in this setting, we are testing the ability of the model to generalize population prediction of treatment response for unseen patients. We do this for 3 different patients and report results across these independent splits. Further details regarding the organoid drug-screen dataset, data pre-processing, and data splits are provided in Appendix D.2.

Table 2: **Experimental results on the organoid drug-screen dataset** for population prediction of treatment response across left-out ***replica*** populations. We use $^{w/}$OT to denote models that incorporate OT couplings between source and target distributions. Note, the ICNN model operates in the OT regime. We use $^{w/}\mathcal{N}$ to denote models that use Gaussian source distributions sampled via $x_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$, but maintain the control populations p_0 as input to the population embedding model $\varphi(p_0; \theta, k)$. For extended results see Appendix F.

	$\mathcal{W}_1(\downarrow)$	$\mathcal{W}_2(\downarrow)$	MMD $(\times 10^{-3})$ $(\downarrow)$	$r^2(\uparrow)$
$d(p_0, p_1)$	4.513	4.695	19.14	0.876
$d(p_0 - \mu_0 + \tilde{\mu}_1, p_1)$	6.222	6.346	74.90	0.876
FM	4.340 ± 0.078	4.564 ± 0.111	13.00 ± 0.67	0.865 ± 0.034
FM$^{w/}$OT	4.210 ± 0.006	4.397 ± 0.001	12.16 ± 0.72	0.910 ± 0.005
FM$^{w/}\mathcal{N}$	7.114 ± 0.100	7.404 ± 0.086	64.97 ± 3.79	0.613 ± 0.008
CGFM	4.443 ± 0.033	4.621 ± 0.041	17.00 ± 1.03	0.899 ± 0.008
CGFM$^{w/}$OT	4.356 ± 0.027	4.531 ± 0.025	15.82 ± 0.19	0.909 ± 0.003
CGFM$^{w/}\mathcal{N}$	7.135 ± 0.045	7.390 ± 0.037	79.78 ± 4.67	0.637 ± 0.010
ICNN	4.488 ± 0.035	4.665 ± 0.038	17.60 ± 0.55	0.884 ± 0.002
MFM$^{w/}\mathcal{N}$ $(k = 0)$	4.177 ± 0.042	4.355 ± 0.048	10.53 ± 0.59	0.911 ± 0.001
MFM$^{w/}\mathcal{N}$ $(k = 10)$	4.156 ± 0.065	4.324 ± 0.067	9.58 ± 1.63	0.912 ± 0.003
MFM$^{w/}\mathcal{N}$ $(k = 50)$	4.153 ± 0.069	4.324 ± 0.070	9.63 ± 1.45	0.912 ± 0.002
MFM$^{w/}\mathcal{N}$ $(k = 100)$	4.166 ± 0.001	4.341 ± 0.003	9.52 ± 0.33	0.915 ± 0.005
MFM $(k = 0)$	4.209 ± 0.007	4.380 ± 0.012	12.34 ± 0.50	0.918 ± 0.002
MFM $(k = 10)$	4.216 ± 0.090	4.395 ± 0.098	11.99 ± 2.36	0.917 ± 0.005
MFM $(k = 50)$	4.214 ± 0.017	4.396 ± 0.020	12.09 ± 0.75	0.916 ± 0.002
MFM $(k = 100)$	4.100 ± 0.093	4.269 ± 0.104	$\mathbf{8.96 \pm 1.88}$	0.917 ± 0.004
MFM$^{w/}$OT $(k = 100)$	$\mathbf{4.098 \pm 0.017}$	$\mathbf{4.263 \pm 0.019}$	10.37 ± 0.29	$\mathbf{0.919 \pm 0.002}$

Baselines. For the organoid drug-screen experiments, we consider an ICNN model as an additional baseline alongside FM and CGFM. This is equivalent to the architecture used in CellOT (Bunne et al., 2023a); a method for learning cell specific response to treatments. The ICNN (and likewise CellOT) counterparts our FM$^{w/}$OT model in that it does not take the population index i as a condition. CGFM takes in the population identity index i as a one-hot input condition. Unlike CGFM, MFM does not

[2]We consider only the highest dosage and leave exploration of dose-dependent response to future work.

require knowledge of population identities. We additionally include two model-agnostic baselines: (i) the distributional distances between the target distribution and source distribution $d(p_0, p_1)$, and (ii) the distributional distances between the target distribution and the source distribution shifted by the average mean of target distributions ($\tilde{\mu}_1$) in the training data $d(p_0 - \mu_0 + \tilde{\mu}_1, p_1)$.

Predicting treatment response across replicates. We show results for generalization across replicates for the replica-2 split in Table 2. In Appendix F, we include an extended evaluation on a second replica split as well as report results for separate cell cultures in Table 8 and Table 9. We observe that MFM outperforms all baseline methods on left-out replicas. Although this is in general a simpler generalization problem compared to the left-out patient populations setting, this result suggests that there is arguably sufficient biological heterogeneity across replicas for MFM to learn meaningful embeddings. We observe that OT generally increases performance across all methods.

Predicting treatment response across patients. We show results for generalization across patients in Table 3. We observe that MFM (without OT) outperforms baseline methods, including the ICNN model which learns OT couplings between source and target distributions. All methods yield generally equal degrees of variation across 3 patient splits, with comparable variation relative to the trivial baseline. See Appendix F.1 for an analysis of population embeddings.

Table 3: **Experimental results on the organoid drug-screen dataset** for population prediction of treatment response across left-out ***patient*** populations. We report the mean and standard deviations across metrics computed over 3 different patient splits.

	$\mathcal{W}_1(\downarrow)$	$\mathcal{W}_2(\downarrow)$	MMD $(\times 10^{-3})$ $(\downarrow)$	$r^2(\uparrow)$
$d(p_0, p_1)$	4.175 ± 0.135	4.303 ± 0.174	12.11 ± 2.07	0.902 ± 0.006
$d(p_0 - \mu_0 + \tilde{\mu}_1, p_1)$	6.158 ± 0.239	6.235 ± 0.229	77.74 ± 10.72	0.902 ± 0.006
FM	4.171 ± 0.107	4.315 ± 0.142	10.95 ± 1.98	0.897 ± 0.023
CGFM	4.189 ± 0.088	4.321 ± 0.119	11.57 ± 0.96	0.914 ± 0.008
ICNN	4.157 ± 0.168	4.282 ± 0.213	11.18 ± 2.51	0.904 ± 0.005
MFM ($k = 0$)	4.135 ± 0.094	4.268 ± 0.128	10.18 ± 1.28	0.918 ± 0.007
MFM ($k = 10$)	4.112 ± 0.086	4.243 ± 0.121	9.90 ± 0.99	0.925 ± 0.008
MFM ($k = 50$)	$\mathbf{4.087 \pm 0.122}$	$\mathbf{4.218 \pm 0.160}$	$\mathbf{9.26 \pm 1.56}$	0.926 ± 0.007
MFM ($k = 100$)	4.112 ± 0.148	4.244 ± 0.186	9.63 ± 2.08	$\mathbf{0.931 \pm 0.002}$

Through the biological and synthetic experiments, we have shown that MFM is able to generalize to unseen distributions/populations. The implication of our results suggest that MFM can learn population dynamics in unseen environments. In biological contexts, like the one we have shown in this work, this result indicates that we can improve learning of population dynamics, of treatment response or any arbitrary perturbation, in new/unseen patients. This works towards a model where it is possible to predict and design an individualized treatment regimen for each patient based on their individual characteristics and tumor microenvironment.

6 Conclusion

Our paper highlights the significance of modeling dynamics based on the entire distribution. While flow-based models offer a promising avenue for learning dynamics at the population level, they were previously restricted to a single initial population and predefined (known) conditions. In this paper, we introduce *Meta Flow Matching* (MFM) as a practical solution to address these limitations. By integrating along vector fields of the Wasserstein manifold, MFM allows for a more comprehensive model of dynamical systems with interacting particles. Crucially, MFM leverages graph neural networks to embed the initial population, enabling the model to generalize over various initial distributions. MFM opens up new possibilities for understanding complex phenomena that emerge from interacting systems in biological and physical systems. In practice, we demonstrate that MFM learns meaningful embeddings of single-cell populations along with the developmental model of these populations. Moreover, our empirical study demonstrates the possibility of modeling patient-specific response to treatments via the meta-learning. We discuss broader impacts of this work in Appendix C.

Limitations. In this work we explored empirically the effect of conditioning the learned flow on the initial distribution. We argue this is a more natural model for many biological systems. However, there are many other aspects of modeling biological systems that we did not consider. In particular we did not consider extensions to the manifold setting (Huguet et al., 2022; 2023), unbalanced optimal transport (Benamou, 2003; Yang and Uhler, 2019; Chizat et al., 2018), aligned (Somnath et al., 2023; Liu et al., 2023), or stochastic settings (Bunne et al., 2023a; Koshizuka and Sato, 2023) in this work.

Future work. In this work, we focused on exploring the amortization of learning over the space of distributions using flow matching. However, our framework can be extended beyond flow matching. One of the central innovations in this work is conditioning vector fields on representations of entire distributions. Hence, this can be easily extended to other training regimes. Furthermore, we proposed the joint training of parameters ω and θ for the vector field $v_t(\cdot; \omega)$ and population embedding model $\varphi(\cdot; \theta)$ using the loss defined in Eq. (17). Although this is a novel approach to $v_t(\cdot|\varphi(\cdot; \theta); \omega)$ training, there remains room to explore the pre-training and/or use of different losses for the population embedding model.

REPRODUCIBILITY STATEMENT

To ensure reproducibility of our findings and results we submitted our source code as supplementary materials. In addition, we describe the mathematical details of our method throughout Section 3 and provide pseudo-code to reproduce training and inference in Algorithm 1 and Algorithm 2. We provide details regarding parameterization of models, optimization, and implementation in Appendix D.4 and Appendix E. For our datasets, we include details regarding the synthetic letters dataset and how to construct in it Section 5.1 and Appendix D.1, and details for data download and data processing for the Organoid drug-screen data in Section 5.2 and Appendix D.2. Information regarding the datasets that we consider in this work can also be found in the source code. For our theoretical contributions, we state our assumptions throughout the text in Section 2 and Section 3 and provide detailed proofs for Proposition 1 (in the main text) and Theorem 1 (in Appendix B).

REFERENCES

Achille, A., Lam, M., Tewari, R., Ravichandran, A., Maji, S., Fowlkes, C. C., Soatto, S., and Perona, P. (2019). Task2vec: Task embedding for meta-learning. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*, pages 6430–6439.

Albergo, M. S. and Vanden-Eijnden, E. (2022). Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*.

Ambrosio, L., Gigli, N., and Savaré, G. (2008). *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media.

Amos, B., Cohen, S., Luise, G., and Redko, I. (2022). Meta optimal transport. *arXiv preprint arXiv:2206.05262*.

Amos, B. et al. (2023). Tutorial on amortized optimization. *Foundations and Trends® in Machine Learning*, 16(5):592–732.

Amos, B., Xu, L., and Kolter, J. Z. (2017). Input convex neural networks. In *International conference on machine learning*, pages 146–155. PMLR.

Armingol, E., Officer, A., Harismendy, O., and Lewis, N. E. (2020). Deciphering cell–cell interactions and communication from gene expression. *Nature Reviews Genetics*, 22(2):71–88.

Benamou, J.-D. (2003). Numerical resolution of an "unbalanced" mass transport problem. *ESAIM: Mathematical Modelling and Numerical Analysis*, 37(5):851–868.

Binnewies, M., Roberts, E. W., Kersten, K., Chan, V., Fearon, D. F., Merad, M., Coussens, L. M., Gabrilovich, D. I., Ostrand-Rosenberg, S., Hedrick, C. C., Vonderheide, R. H., Pittet, M. J., Jain, R. K., Zou, W., Howcroft, T. K., Woodhouse, E. C., Weinberg, R. A., and Krummel, M. F. (2018). Understanding the tumor immune microenvironment (time) for effective therapy. *Nature Medicine*, 24(5):541–550.

Bunne, C., Stark, S. G., Gut, G., Del Castillo, J. S., Levesque, M., Lehmann, K.-V., Pelkmans, L., Krause, A., and Rätsch, G. (2023a). Learning single-cell perturbation responses using neural optimal transport. *Nature Methods*, 20(11):1759–1768.

Bunne, C., Stark, S. G., Gut, G., del Castillo, J. S., Levesque, M., Lehmann, K.-V., Pelkmans, L., Krause, A., and Rätsch, G. (2023b). Learning single-cell perturbation responses using neural optimal transport. *Nature Methods*, 20(11):1759–1768.

Campbell, A., Yim, J., Barzilay, R., Rainforth, T., and Jaakkola, T. (2024). Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. *arXiv preprint arXiv:2402.04997*.

Chen, S., Rivaud, P., Park, J. H., Tsou, T., Charles, E., Haliburton, J. R., Pichiorri, F., and Thomson, M. (2020). Dissecting heterogeneous cell populations across drug and disease conditions with popalign. *Proceedings of the National Academy of Sciences*, 117(46):28784–28794.

Chen, T., Chen, X., Chen, W., Heaton, H., Liu, J., Wang, Z., and Yin, W. (2022). Learning to optimize: A primer and a benchmark. *Journal of Machine Learning Research*, 23(189):1–59.

Chizat, L., Peyré, G., Schmitzer, B., and Vialard, F.-X. (2018). Unbalanced optimal transport: Dynamic and kantorovich formulations. *Journal of Functional Analysis*, 274(11):3090–3123.

Chung, W., Eum, H. H., Lee, H.-O., Lee, K.-M., Lee, H.-B., Kim, K.-T., Ryu, H. S., Kim, S., Lee, J. E., Park, Y. H., Kan, Z., Han, W., and Park, W.-Y. (2017). Single-cell rna-seq enables comprehensive tumour and immune cell profiling in primary breast cancer. *Nature Communications*, 8(1).

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

De Bortoli, V., Thornton, J., Heng, J., and Doucet, A. (2021). Diffusion schrödinger bridge with applications to score-based generative modeling. *Advances in Neural Information Processing Systems*, 34:17695–17709.

Frangieh, C. J., Melms, J. C., Thakore, P. I., Geiger-Schuller, K. R., Ho, P., Luoma, A. M., Cleary, B., Jerby-Arnon, L., Malu, S., Cuoco, M. S., Zhao, M., Ager, C. R., Rogava, M., Hovey, L., Rotem, A., Bernatchez, C., Wucherpfennig, K. W., Johnson, B. E., Rozenblatt-Rosen, O., Schadendorf, D., Regev, A., and Izar, B. (2021). Multimodal pooled perturb-cite-seq screens in patient models define mechanisms of cancer immune evasion. *Nature Genetics*, 53(3):332–341.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. (2017). Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR.

Goodenough, D. A. and Paul, D. L. (2009). Gap junctions. *Cold Spring Harb Perspect Biol*, 1(1):a002576.

Gulati, G. S., Sikandar, S. S., Wesche, D. J., Manjunath, A., Bharadwaj, A., Berger, M. J., Ilagan, F., Kuo, A. H., Hsieh, R. W., Cai, S., Zabala, M., Scheeren, F. A., Lobo, N. A., Qian, D., Yu, F. B., Dirbas, F. M., Clarke, M. F., and Newman, A. M. (2020). Single-cell transcriptional diversity is a hallmark of developmental potential. *Science*, 367(6476):405–411.

Hashimoto, T. B., Gifford, D. K., and Jaakkola, T. S. (2016). Learning population-level diffusions with generative recurrent networks. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 2417–2426.

Hetzel, L., Boehm, S., Kilbertus, N., Günnemann, S., Lotfollahi, M., and Theis, F. (2022). Predicting cellular responses to novel drug perturbations at a single-cell resolution. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A., editors, *Advances in Neural Information Processing Systems*, volume 35, pages 26711–26722. Curran Associates, Inc.

Hospedales, T., Antoniou, A., Micaelli, P., and Storkey, A. (2021). Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):5149–5169.

Huguet, G., Magruder, D. S., Tong, A., Fasina, O., Kuchroo, M., Wolf, G., and Krishnaswamy, S. (2022). Manifold interpolating optimal-transport flows for trajectory inference.

Huguet, G., Tong, A., Zapatero, M. R., Wolf, G., and Krishnaswamy, S. (2023). Geodesic sinkhorn: Optimal transport for high-dimensional datasets. In *IEEE MLSP*.

Isobe, N., Koyama, M., Hayashi, K., and Fukumizu, K. (2024). Extended flow matching: a method of conditional generation with generalized continuity equation. *arXiv preprint arXiv:2402.18839*.

Jabin, P.-E. and Wang, Z. (2016). Mean field limit and propagation of chaos for vlasov systems with bounded forces. *Journal of Functional Analysis*, 271(12):3588–3627.

Ji, Y., Lotfollahi, M., Wolf, F. A., and Theis, F. J. (2021). Machine learning for perturbational single-cell omics. *Cell Systems*, 12(6):522–537.

Koshizuka, T. and Sato, I. (2023). Neural lagrangian schr\"odinger bridge. In *ICLR*.

Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M. (2023). Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*.

Liu, G.-H., Chen, T. Q., So, O., and Theodorou, E. A. (2022a). Deep generalized schrödinger bridge. *NeurIPS*.

Liu, G.-H., Lipman, Y., Nickel, M., Karrer, B., Theodorou, E. A., and Chen, R. T. Q. (2024). Generalized schrödinger bridge matching. *International Conference on Learning Representations*.

Liu, G.-H., Vahdat, A., Huang, D.-A., Theodorou, E. A., Nie, W., and Anandkumar, A. (2023). I^2sb: Image-to-image schrödinger bridge. In *ICML*.

Liu, X., Gong, C., and Liu, Q. (2022b). Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*.

Lopez, R., Regier, J., Cole, M. B., Jordan, M. I., and Yosef, N. (2018). Deep generative modeling for single-cell transcriptomics. *Nature Methods*, 15(12):1053–1058.

Lotfollahi, M., Wolf, F. A., and Theis, F. J. (2019). scgen predicts single-cell perturbation responses. *Nature Methods*, 16(8):715–721.

Makkuva, A. V., Taghvaei, A., Oh, S., and Lee, J. D. (2020). Optimal transport mapping via input convex neural networks. In *ICML*.

Molè, M. A., Coorens, T. H. H., Shahbazi, M. N., Weberling, A., Weatherbee, B. A. T., Gantner, C. W., Sancho-Serra, C., Richardson, L., Drinkwater, A., Syed, N., Engley, S., Snell, P., Christie, L., Elder, K., Campbell, A., Fishel, S., Behjati, S., Vento-Tormo, R., and Zernicka-Goetz, M. (2021). A single cell characterisation of human embryogenesis identifies pluripotency transitions and putative anterior hypoblast centre. *Nature Communications*, 12(1).

Neklyudov, K., Brekelmans, R., Tong, A., Atanackovic, L., Liu, Q., and Makhzani, A. (2023). A computational framework for solving wasserstein lagrangian flows. *arXiv preprint arXiv:2310.10649*.

Neklyudov, K., Severo, D., and Makhzani, A. (2022). Action matching: A variational method for learning stochastic dynamics from samples.

Otto, F. (2001). The geometry of dissipative evolution equations: the porous medium equation.

Peidli, S., Green, T. D., Shen, C., Gross, T., Min, J., Garda, S., Yuan, B., Schumacher, L. J., Taylor-King, J. P., Marks, D. S., et al. (2024). scperturb: harmonized single-cell perturbation data. *Nature Methods*, pages 1–10.

Peyré, G. and Cuturi, M. (2019). *Computational Optimal Transport*. arXiv:1803.00567.

Pooladian, A.-A., Ben-Hamu, H., Domingo-Enrich, C., Amos, B., Lipman, Y., and Chen, R. T. (2023). Multisample flow matching: Straightening flows with minibatch couplings. *arXiv preprint arXiv:2304.14772*.

Price, I., Sanchez-Gonzalez, A., Alet, F., Ewalds, T., El-Kadi, A., Stott, J., Mohamed, S., Battaglia, P., Lam, R., and Willson, M. (2023). Gencast: Diffusion-based ensemble forecasting for medium-range weather. *arXiv preprint arXiv:2312.15796*.

Ramos Zapatero, M., Tong, A., Opzoomer, J. W., O'Sullivan, R., Cardoso Rodriguez, F., Sufi, J., Vlckova, P., Nattress, C., Qin, X., Claus, J., Hochhauser, D., Krishnaswamy, S., and Tape, C. J. (2023). Trellis tree-based analysis reveals stromal regulation of patient-derived organoid drug responses. *Cell*, 186(25):5606–5619.e24.

Rizvi, A. H., Camara, P. G., Kandror, E. K., Roberts, T. J., Schieren, I., Maniatis, T., and Rabadan, R. (2017). Single-cell topological rna-seq analysis reveals insights into cellular differentiation and development. *Nature Biotechnology*, 35(6):551–560.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695.

Saharia, C., Chan, W., Chang, H., Lee, C., Ho, J., Salimans, T., Fleet, D., and Norouzi, M. (2022a). Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 conference proceedings*, pages 1–10.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E. L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al. (2022b). Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494.

Sanchez-Gonzalez, A., Godwin, J., Pfaff, T., Ying, R., Leskovec, J., and Battaglia, P. (2020). Learning to simulate complex physics with graph networks. In *International conference on machine learning*, pages 8459–8468. PMLR.

Schiebinger, G., Shu, J., Tabaka, M., Cleary, B., Subramanian, V., Solomon, A., Gould, J., Liu, S., Lin, S., Berube, P., et al. (2019). Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming. *Cell*, 176(4):928–943.

Somnath, V. R., Pariset, M., Hsieh, Y.-P., Martinez, M. R., Krause, A., and Bunne, C. (2023). Aligned diffusion schröodinger bridges. In *UAI*.

Svensson, V., Vento-Tormo, R., and Teichmann, S. A. (2018). Exponential scaling of single-cell rna-seq in the past decade. *Nature protocols*, 13(4):599–604.

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

Tong, A., Huang, J., Wolf, G., Van Dijk, D., and Krishnaswamy, S. (2020). Trajectorynet: A dynamic optimal transport network for modeling cellular dynamics. In *International conference on machine learning*, pages 9526–9536. PMLR.

Tong, A. Y., Malkin, N., Fatras, K., Atanackovic, L., Zhang, Y., Huguet, G., Wolf, G., and Bengio, Y. (2024b). Simulation-free schrödinger bridges via score and flow matching. In *International Conference on Artificial Intelligence and Statistics*, pages 1279–1287. PMLR.

Uscidda, T. and Cuturi, M. (2023). The monge gap: A regularizer to learn all transport maps. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J., editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 34709–34733. PMLR.

Verma, Y., Heinonen, M., and Garg, V. (2024). Climode: Climate and weather forecasting with physics-informed neural odes. *arXiv preprint arXiv:2404.10024*.

Villani, C. (2009). *Optimal transport: old and new*, volume 338. Springer.

Weinreb, C., Wolock, S., Tusi, B. K., Socolovsky, M., and Klein, A. M. (2018). Fundamental limits on dynamic inference from single-cell snapshots. 115(10):E2467–E2476.

Yang, K. D. and Uhler, C. (2019). Scalable unbalanced optimal transport using generative adversarial networks. In *7th International Conference on Learning Representations*, page 20.

Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R. R., and Smola, A. J. (2017). Deep sets. *Advances in neural information processing systems*, 30.

Zeng, T. and Dai, H. (2019). Single-cell rna sequencing-based computational analysis to describe disease heterogeneity. *Frontiers in Genetics*, 10.

Zheng, Q., Le, M., Shaul, N., Lipman, Y., Grover, A., and Chen, R. T. (2023). Guided flows for generative modeling and decision making. *arXiv preprint arXiv:2311.13443*.

A DEFINITIONS

All the densities we consider are the densities of the probability measures from the following set

$$\mathcal{P}_2(\mathcal{X}) := \left\{ \mu \in \mathcal{P}(\mathcal{X}) : \int_{\mathcal{X}} d\mu(x)\, d(x, \bar{x})^2 < +\infty \quad \text{for some} \quad \bar{x} \in \mathcal{X} \right\}, \tag{20}$$

where $\mathcal{P}(\mathcal{X})$ is the family of all Borel probability measures on $\mathcal{X}$.

Thus, one can introduce 2-Wasserstein distance between measures in $\mathcal{P}_2(\mathcal{X})$, i.e.

$$W_2^2(\mu_0, \mu_1) := \min \left\{ \int_0^1 dt\, \|v_t\|^2_{L^2(p_t;\mathcal{X})} : \frac{\partial p_t}{\partial t} + \langle \nabla, p_t v_t \rangle = 0 \right\}, \tag{21}$$

where we used the Benamou-Brenier formulation of optimal transport (Benamou, 2003). That is, the possible changes of the density are restricted by the continuity equation

$$\frac{\partial p_t(x)}{\partial t} + \langle \nabla_x, p_t(x) v_t(x) \rangle = 0\,, \tag{22}$$

where the define the divergence as

$$\langle \nabla_x, v(x) \rangle = \sum_{i=1}^{d} \frac{\partial v_i(x)}{\partial x_i}\,. \tag{23}$$

Note that $W_2^2(\cdot,\cdot)$ define a metric on $\mathcal{P}_2(\mathcal{X})$, which allows to introduce absolutely-continuous curves and the following tangent space

$$\mathrm{Tan}_\mu \mathcal{P}_2(\mathcal{X}) := \overline{\{\nabla s : s \in C^\infty(\mathcal{X})\}}^{L^2(\mu;\mathcal{X})} \quad \forall \mu \in \mathcal{P}_2(\mathcal{X})\,. \tag{24}$$

These developments lead to the celebrated Riemannian interpretation of the 2-Wasserstein distance $W_2^2(\cdot,\cdot)$ on $\mathcal{P}_2(\mathcal{X})$ proposed by (Otto, 2001) and rigorously studied and presented in (Ambrosio et al., 2008).

B PROOF OF THEOREM 1

Theorem 1. *Consider a dataset of populations $\mathcal{D} = \{(\pi(x_0, x_1 \,|\, i))\}_i$ generated from some unknown conditional model $\pi(x_0, x_1 \,|\, c)p(c)$. Then the following objective*

$$\mathcal{L}(\omega, \theta) = \mathbb{E}_{p(c)} \int_0^1 dt\, \mathbb{E}_{p_t(x_t \,|\, c)} \|v_t^*(x_t \,|\, c) - v_t(x_t \,|\, \varphi(p_0, \theta), \omega)\|^2 \tag{16}$$

is equivalent to the Meta Flow Matching objective

$$\mathcal{L}_{MFM}(\omega, \theta) = \mathbb{E}_{i \sim \mathcal{D}} \mathbb{E}_{\pi(x_0, x_1 \,|\, i)} \int_0^1 dt \left\| \frac{\partial}{\partial t} f_t(x_0, x_1) - v_t(f_t(x_0, x_1) \,|\, \varphi(p_0, \theta); \omega) \right\|^2 \tag{17}$$

up to an additive constant.

Proof. The loss function

$$\begin{aligned}
\mathcal{L}(\omega, \theta) &= \mathbb{E}_{p(c)} \int_0^1 dt\, \mathbb{E}_{p_t(x_t \,|\, c)} \|v_t^*(x_t \,|\, c) - v_t(x_t \,|\, \varphi(p_t, \theta); \omega)\|^2 && (25)\\
&= -2\mathbb{E}_{p(c)} \int dt dx\, \langle p_t(x \,|\, c) v_t^*(x \,|\, c), v_t(x \,|\, \varphi(p_t, \theta); \omega) \rangle + && (26)\\
&\quad + \mathbb{E}_{p(c)} \int_0^1 dt\, \mathbb{E}_{p_t(x_t \,|\, c)} \|v_t(x_t \,|\, \varphi(p_t, \theta), \omega)\|^2 + && (27)\\
&\quad + \mathbb{E}_{p(c)} \int_0^1 dt\, \mathbb{E}_{p_t(x_t \,|\, c)} \|v_t^*(x_t \,|\, c)\|^2\,. && (28)
\end{aligned}$$

The last term does not depend on θ, the second term we can estimate, for the first term, we use the formula for the (from Eq. (8))

$$p_t(\xi \,|\, c) v_t^*(\xi \,|\, c) = \mathbb{E}_{\pi(x_0, x_1)} \delta(f_t(x_0, x_1) - \xi) \frac{\partial f_t(x_0, x_1)}{\partial t}\,. \tag{29}$$

Thus, the loss is equivalent (up to a constant) to

$$\mathcal{L}(\omega,\theta) = -2\mathbb{E}_{p(c)}\mathbb{E}_{\pi(x_0,x_1\,|\,c)}\int dt\,\left\langle \frac{\partial f_t(x_0,x_1)}{\partial t}, v_t(f_t(x_0,x_1)\,|\,\varphi(p_t,\theta);\omega)\right\rangle + \tag{30}$$

$$+\,\mathbb{E}_{p(c)}\mathbb{E}_{\pi(x_0,x_1\,|\,c)}\int_0^1 dt\,\|v_t(f_t(x_0,x_1)\,|\,\varphi(p_t,\theta),\omega)\|^2 \pm \tag{31}$$

$$\pm\,\mathbb{E}_{p(c)}\mathbb{E}_{\pi(x_0,x_1\,|\,c)}\int_0^1 dt\,\left\|\frac{\partial f_t(x_0,x_1)}{\partial t}\right\|^2 \tag{32}$$

$$= \mathbb{E}_{c\sim p(c)}\mathbb{E}_{\pi(x_0,x_1\,|\,c)}\int_0^1 dt\,\left\|\frac{\partial}{\partial t}f_t(x_0,x_1) - v_t(f_t(x_0,x_1)\,|\,\varphi(p_t,\theta);\omega)\right\|^2. \tag{33}$$

Note that in the final expression we do not need access to the probabilistic model of $p(c)$ if the joints $\pi(x_0,x_1\,|\,c)$ are already sampled in the data $\mathcal{D}$. Thus, we have

$$\mathcal{L}(\omega,\theta) = \mathbb{E}_{c\sim p(c)}\mathbb{E}_{\pi(x_0,x_1\,|\,c)}\int_0^1 dt\,\left\|\frac{\partial}{\partial t}f_t(x_0,x_1) - v_t(f_t(x_0,x_1)\,|\,\varphi(p_t,\theta);\omega)\right\|^2 \tag{34}$$

$$= \mathbb{E}_{i\sim\mathcal{D}}\mathbb{E}_{\pi(x_0,x_1\,|\,i)}\int_0^1 dt\,\left\|\frac{\partial}{\partial t}f_t(x_0,x_1) - v_t(f_t(x_0,x_1)\,|\,\varphi(p_t,\theta);\omega)\right\|^2 \tag{35}$$

$$= \mathcal{L}_{\text{MFM}}(\omega,\theta)\,. \tag{36}$$

□

C Broader Impacts

This paper is primarily a theoretical and methodological contribution with little societal impact. MFM can be used to better model dynamical systems of interacting particles and in particular cellular systems. Better modeling of cellular systems can potentially be used for the development of malicious biological agents. However, we do not see this as a significant risk at this time.

D Experimental Details

D.1 Synthetic Letters Data

The synthetic letters dataset (used for Fig. 3 and Table 1) contains 240 train populations a 10 test populations. Each population contains roughly between 750 and 2700 samples in this dataset.

D.2 Organoid Drug-Screen Data

The *organoid drug-screen* dataset consists of patient derived organoids (PDOs) from 10 different patients (Ramos Zapatero et al., 2023).[3] For each patient, experiments are replicated and performed over varrying conditions. This results in a significant amount of coupled populations pairs $(p_0(x_0|c_i), p_1(x_1|c_i))$, where we use c_i to denote the treatment condition corresponding to respective population pair. After pre-processing, we acquire a total of 927 replicates (or coupled population pairs). In the *replica-1 split*, we use 713 populations for training, 111 left-out population for validation, and 103 left-out populations for testing. In the *replica-2 split*, we use 861 populations for training, 33 left-out populations for validation, and 33 left-out populations for testing. We use the replica splits to set to select reasonable hyperparameters for MFM and the baseline models.

For the patients split, we consider 3 different patient splits where we independently leave out all populations from either patient 21, patient 27, and patients 75. Organoid PDO-21 was found to present an interesting chemoprotective response (High CAF Protection) to cancer treatments, and hence was selected by Ramos Zapatero et al. (2023) for further study and data collection. For the same reason we select PDO-21 as the unseen test patient for the *patients split* experiments. We additionally consider populations from PDO-75 as a third left-out patient split. In the patients splits, we only split data into train and test sets, resulting in 839 population pairs for training and 88 population pairs for test evaluation for PDO-21 and PDO-27 splits. For PDO-75, the train split contain 830 train population pairs and 97 test population pairs.

[3] PDOs are cultures of cell populations which are derived from patient cells.

D.3 METRICS

We use four metrics to evaluate the quality of a generated empirical distribution $\hat{p}$ as compared to an empirical ground truth distribution p. In this section we detail the calculation and interpretation of each metric. For letters, we evaluate over all samples, while for the orgranoid drug-screen dataset we evaluate over 5000 cells. If a target population contains less than 5000 cells, we evaluate using sample size corresponding to the size of the entire target population.

1-Wasserstein ($\mathcal{W}_1$) and 2-Wasserstein ($\mathcal{W}_2$). We evaluate the Wasserstein distance for $\alpha = 1$ and $\alpha = 2$. Specifically for $\{\hat{x}_i\}_{i=0}^n = \hat{p}$ and $\{x_i\}_{i=0}^n = p$ we calculate the α-Wasserstein distance with respect to the Euclidean distance as

$$\mathcal{W}_\alpha = \left(\inf_{\pi \in \Pi(\hat{p}, p)} \int \|x - y\|_2^\alpha d\pi(x, y) \right)^{1/\alpha} \tag{37}$$

where $\Pi(\hat{p}, p)$ denotes the set of all joint probability measures with marginals $\hat{p}$, p.

Maximum mean discrepancy (MMD). We evaluate an estimate for MMD between empirical distributions $\{\hat{x}_i\}_{i=0}^n = \hat{p}$ and $\{x_i\}_{i=0}^n = p$ using the radial basis function (RBF) kernel $k(\cdot, \cdot; \gamma)$ as:

$$\text{MMD}(p, \hat{p}) = \frac{1}{n^2} \sum_i^n \sum_j^n k(\hat{x}_i, \hat{x}_j; \gamma) + \frac{1}{n^2} \sum_i^n \sum_j^n k(x_i, x_j; \gamma) - \frac{2}{n^2} \sum_i^n \sum_j^n k(\hat{x}_i, x_j; \gamma). \tag{38}$$

We estimate MMD for $\gamma = 2, 1, 0.5, 0.1, 0.01, 0.005$ and report the average.

Coefficient of determination (r^2). We adopt the r^2 metric, overall average correlation coefficient, from Bunne et al. (2023a) used to compare predictions and observations. This metric is computed as follows. First, given two arrays $X \in \mathbb{R}^{n \times d}$ and $Y \in \mathbb{R}^{n \times d}$, where n is the number of samples and d is the number of features, we can compute the correlation as

$$\text{corr}(X, Y) = \frac{\sum_{i=1}^n (X_i - \mu_X)(Y_i - \mu_Y)}{\sqrt{\sum_{i=1}^n (X_i - \mu_X)^2} \sqrt{\sum_{i=1}^n (Y_i - \mu_Y)^2}}, \tag{39}$$

where μ_X and μ_Y are the empirical means of X and Y. We compute correlation across all points for X (predicted samples) and Y (target samples) individually

$$\Sigma_{ij}^X = \text{corr}(X_i, X_j), \quad \Sigma_{ij}^Y = \text{corr}(Y_i, Y_j). \tag{40}$$

Then, we consider the two arrays

$$S^X = \Sigma_{ij}^X \forall i < j, S^Y = \Sigma_{ij}^Y \forall i < j. \tag{41}$$

The overall average correlation coefficient is reported as

$$r^2 = \text{corr}(S^X, S^Y). \tag{42}$$

We compute a r^2 for every population in the evaluation sets and report the average r^2 over this set. We take this evaluation directly from Bunne et al. (2023a), which is a typical evaluation of single-cell prediction methods.

D.4 MODEL ARCHITECTURES AND HYPERPARAMETERS

ICNN The ICNN baseline was constructed with two networks ICNN network $f(x)$ and $g(x)$, with non-negative leaky ReLU activation layers. $f(x)$ is used to minimize the transport distance and $g(x)$ is used to transport from source to target. It has four hidden units with width of 64, and a latent dimension of 50. Both networks uses Adam optimizer ($lr = 10^{-4}$, $\beta_1 = 0.5$, $\beta_2 = 0.9$). $g(x)$ is trained with an inner iteration of 10 for every iteration $f(x)$ is trained.

Vector Field Models All vector field models v_t are parameterized with linear layers of 512 hidden units and SELU activation functions. For the synthetic experiments, we use 4 hidden layers. For the biological experiments, we use 7 hidden layers and skip connections across every layer. We found that this setup worked well for the biological experiments. The FM vector field model additionally takes a conditional input for the one-hot treatment encoding. CGFM takes the conditional input for the one-hot treatment conditions as well as a one-hot encoding for the population index condition i. The MFM vector field model takes population embedding conditions, that is output from the GCN, as input, as well as the treatment one-hot encoding. All vector field models use temporal embeddings for time and positional embeddings for the input samples. We did not sweep the size of this embeddings space and found that a temporal embedding and positional embeddings sizes of 128 worked sufficiently well.

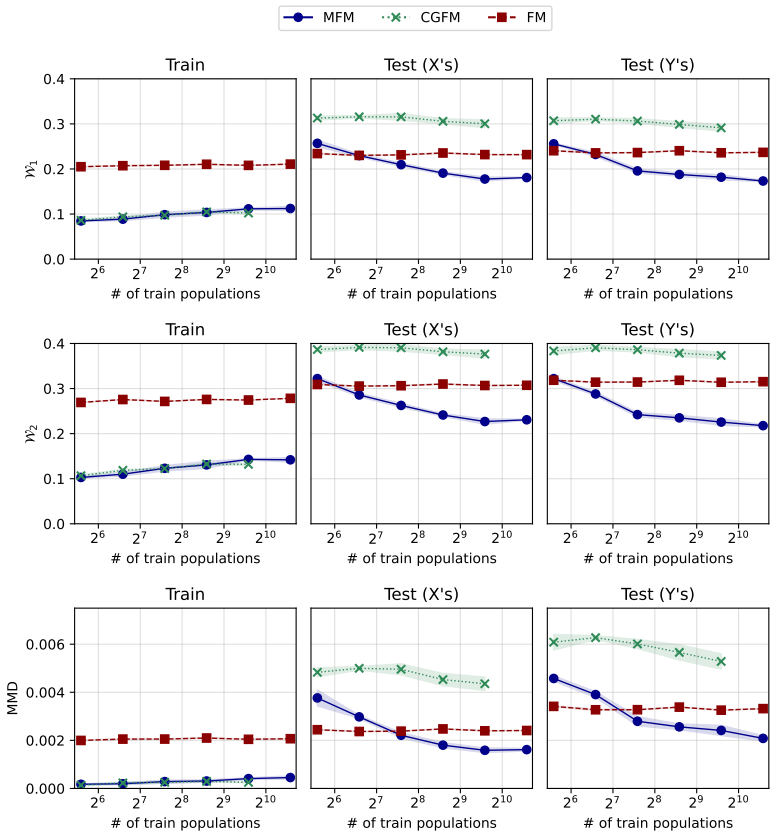

Figure 5: **Synthetic letters ablation over number of training populations.** Here we fix the test sets (X's and Y's) with 10 random rotations (same across each experiment). We then ablate the number of populations used for training FM (red), CGFM (green), and MFM (blue), by changing the number of random rotations/orientations used for each letter silhouette. We observe that for MFM the distributional errors on the test sets consistently decrease as we increase the number of training populations. In contrast, since FM and CGFM cannot generalize across novel populations, this increase in training populations does not lead to an overall improved performance on the test populations. For a large number of training populations, CGFM exhibits exhaustive memory requirements since it requires one-hot encodings as input conditions to denote the population index.

Graph Neural Network We considered a GCN model that consists of a k-nearest neighbour graph edge pooling layer and graph convolution layers with 512 hidden units. For the synthetic experiment we found that 3 GCN layers to work well, while for the biological experiments we found 2 GCN layers to perform well. The final GCN model layer outputs an embedding representation $e \in \mathbb{R}^d$. For the Synthetic experiment, we found that $d = 64$ performed well, and $d = 128$ performed well for the biological experiments. We normalize and project embeddings onto a hyper-sphere, and find that this normalization helps improve training. Additionally, the GCN takes a one-hot cell-type encoding (encoding for Fibroblast cells or PDO cells) for the control populations p_0. This may be beneficial for PDOF populations where both Fibroblast cells and PDO cells are present. However, it is important to note that labeling which cells are Fibroblasts versus PDOs withing the PDOF cultures is difficult and noisy in itself, hence such a cell-type condition may yield no additive information/performance gain.

Optimization We use the Adam optimizer with a learning rate of 0.0001 for all Flow-matching models (FM, CGFM, MFM). We also used the Adam optimizer with a learning rate of 0.0001 for the GCN model. To train the MFM (FM+GCN) models, we alternate between updating the vector field model parameters ω and the GCN model parameters θ. We alternate between updating the respective model parameters every gradient step. For the syntehtic experiment, FM and CGFM model were trained for 12000 epochs, while MFM models were trained for 24000 epochs, with a population batch size of 10 and a sample batch size of 700. Due to the alternating optimization, the MFM vector field model receives half as many updates compared to its counterparts (FM and CGFM). Therefore, training for the double the epochs is necessary for fair comparison.

Influence of GCN parameter k on performance. We observe that no clear single selection for k yields the best performance across all tasks on the single-cell experiments. For the *replicates* split, $k = 0$ on average performs better than $k > 0$, whereas on the *patients* split, the opposite is true, $k = 100$ performs best on average. This is possibly since the *patients* split forms a more difficult generalization problem (more diversity between training populations and test populations), and hence it is more difficult to over-fit during training with higher k.

The hyperparameters stated in this section were selected from brief and small grid search sweeps. We did not conduct any thorough hyperparameter optimization.

E Implementation Details

We implement all our experiments using PyTorch and PyTorch Geometric. All experiments were conducted on a HPC cluster primarily on NVIDIA Tesla T4 16GB and A40 48GB GPUs. Each individual seed experiment run required only 1 GPU. Depending on the model (FM vs CGFM vs MFM) Each synthetic experiment ran between 6-24 hours and each real-data experiment ran between 12-36 hours. All experiments took approximately 500 GPU hours in aggregate.

F EXTENDED RESULTS

Figure 6: Model-generated samples for synthetic letters from the source ($t = 0$) to target ($t = 1$) distributions.

Table 4: **Comparison of models trained using independent coupling versus OT couplings on the synthetic letters setting.** We report the 1-Wasserstein ($\mathcal{W}_1$), 2-Wasserstein ($\mathcal{W}_2$), and the maximum-mean-discrepancy (MMD) distributional distances. We use $^{w/}$OT to denote models that incorporate OT couplings between source and target distributions (Tong et al., 2024a; Pooladian et al., 2023). We report results for MFM ($k = 50$). Here we report results for models trained with a population batch size of 1 and sample batch size of 700. We observe using OT couplings on this setting does not provide a significant change in performance.

	Train			Test (X's)			Test (Y's)		
	$\mathcal{W}_1(\downarrow)$	$\mathcal{W}_2(\downarrow)$	MMD $_{(\times 10^{-3})}$ $(\downarrow)$	$\mathcal{W}_1(\downarrow)$	$\mathcal{W}_2(\downarrow)$	MMD $_{(\times 10^{-3})}$ $(\downarrow)$	$\mathcal{W}_1(\downarrow)$	$\mathcal{W}_2(\downarrow)$	MMD $_{(\times 10^{-3})}$ $(\downarrow)$
FM	0.210 ± 0.000	0.279 ± 0.000	2.548 ± 0.000	0.231 ± 0.000	0.307 ± 0.000	2.422 ± 0.000	0.236 ± 0.000	0.315 ± 0.000	3.305 ± 0.000
FM$^{w/}$OT	0.210 ± 0.001	0.277 ± 0.001	2.555 ± 0.034	0.234 ± 0.001	0.309 ± 0.001	2.458 ± 0.030	0.239 ± 0.001	0.316 ± 0.001	3.368 ± 0.057
CGFM	$\mathbf{0.093 \pm 0.000}$	$\mathbf{0.118 \pm 0.000}$	0.229 ± 0.000	0.321 ± 0.000	0.395 ± 0.000	5.125 ± 0.000	0.309 ± 0.000	0.388 ± 0.000	6.184 ± 0.000
CGFM$^{w/}$OT	$\mathbf{0.093 \pm 0.000}$	0.119 ± 0.000	$\mathbf{0.215 \pm 0.036}$	0.314 ± 0.002	0.390 ± 0.002	4.871 ± 0.040	0.307 ± 0.001	0.388 ± 0.002	6.021 ± 0.084
MFM	0.100 ± 0.002	0.127 ± 0.002	0.322 ± 0.032	$\mathbf{0.189 \pm 0.003}$	$\mathbf{0.238 \pm 0.005}$	1.742 ± 0.086	$\mathbf{0.190 \pm 0.002}$	$\mathbf{0.237 \pm 0.003}$	$\mathbf{2.473 \pm 0.120}$
MFM$^{w/}$OT	0.103 ± 0.007	0.133 ± 0.010	0.450 ± 0.234	$\mathbf{0.189 \pm 0.006}$	0.239 ± 0.007	$\mathbf{1.706 \pm 0.139}$	0.193 ± 0.006	0.240 ± 0.006	2.522 ± 0.220

Table 5: **Experimental results on the organoid drug-screen dataset** for population prediction of treatment response across ***replica-1*** populations. Results shown in this table are for **dim** $= \mathbf{43}$. We use $^{w/}\mathcal{N}$ to denote models that use Gaussian source distributions sampled via $x_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$, but maintain the control populations p_0 as input to the population embedding model $\varphi(p_0; \theta, k)$. We further report $d(p_0 - \mu_0 + \mu_1, p_1)$, a baseline that uses the means from each true treated populations p_1.

	Train				Test			
	$\mathcal{W}_1(\downarrow)$	$\mathcal{W}_2(\downarrow)$	MMD $_{(\times 10^{-3})}$ $(\downarrow)$	$r^2(\uparrow)$	$\mathcal{W}_1(\downarrow)$	$\mathcal{W}_2(\downarrow)$	MMD $_{(\times 10^{-3})}$ $(\downarrow)$	$r^2(\uparrow)$
$d(p_0, p_1)$	4.182	4.294	10.84	0.925	4.100	4.222	10.21	0.931
$d(p_0 - \mu_0 + \tilde{\mu}_1, p_1)$	6.009	6.074	68.70	0.925	5.985	6.058	68.00	0.931
$d(p_0 - \mu_0 + \mu_1, p_1)$	3.913	3.992	1.97	0.925	–	–	–	–
FM	3.925 ± 0.019	4.041 ± 0.023	3.76 ± 0.26	0.952 ± 0.007	3.961 ± 0.036	4.089 ± 0.042	5.90 ± 0.25	0.941 ± 0.010
FM$^{w/}\mathcal{N}$	6.908 ± 0.037	7.181 ± 0.033	57.70 ± 0.75	0.639 ± 0.005	6.972 ± 0.022	7.244 ± 0.022	60.39 ± 0.98	0.642 ± 0.007
CGFM	$\mathbf{3.864 \pm 0.064}$	3.975 ± 0.069	$\mathbf{3.16 \pm 0.89}$	0.964 ± 0.006	4.087 ± 0.063	4.211 ± 0.066	8.84 ± 0.75	0.938 ± 0.006
CGFM$^{w/}\mathcal{N}$	4.187 ± 0.008	4.340 ± 0.009	8.69 ± 0.50	0.936 ± 0.002	6.852 ± 0.045	7.114 ± 0.044	71.24 ± 3.71	0.666 ± 0.016
ICNN	4.286 ± 0.018	4.313 ± 0.112	38.60 ± 0.21	0.897 ± 0.031	4.194 ± 0.110	4.313 ± 0.112	37.90 ± 2.84	0.897 ± 0.008
MFM$^{w/}\mathcal{N}$ ($k = 0$)	3.940 ± 0.022	4.047 ± 0.023	3.91 ± 0.18	0.959 ± 0.006	3.896 ± 0.026	4.002 ± 0.030	$\mathbf{4.35 \pm 0.18}$	0.950 ± 0.005
MFM$^{w/}\mathcal{N}$ ($k = 10$)	3.976 ± 0.044	4.086 ± 0.049	4.52 ± 0.42	0.961 ± 0.002	3.943 ± 0.032	4.051 ± 0.034	5.28 ± 0.25	0.952 ± 0.001
MFM$^{w/}\mathcal{N}$ ($k = 50$)	3.968 ± 0.013	4.075 ± 0.014	4.36 ± 0.44	0.961 ± 0.002	3.934 ± 0.007	4.041 ± 0.008	4.99 ± 0.35	0.954 ± 0.000
MFM$^{w/}\mathcal{N}$ ($k = 100$)	3.937 ± 0.014	4.040 ± 0.015	3.94 ± 0.00	0.963 ± 0.001	3.908 ± 0.030	4.011 ± 0.033	4.68 ± 0.52	0.953 ± 0.002
MFM ($k = 0$)	3.874 ± 0.015	$\mathbf{3.973 \pm 0.020}$	3.37 ± 0.14	$\mathbf{0.967 \pm 0.003}$	$\mathbf{3.880 \pm 0.009}$	$\mathbf{3.990 \pm 0.011}$	4.68 ± 0.16	0.955 ± 0.002
MFM ($k = 10$)	3.896 ± 0.021	4.000 ± 0.021	3.82 ± 0.12	0.964 ± 0.001	3.899 ± 0.013	4.012 ± 0.011	5.13 ± 0.48	0.955 ± 0.001
MFM ($k = 50$)	3.888 ± 0.038	3.991 ± 0.030	3.59 ± 0.41	0.963 ± 0.001	3.900 ± 0.038	4.013 ± 0.034	5.06 ± 0.22	0.954 ± 0.003
MFM ($k = 100$)	3.906 ± 0.010	4.008 ± 0.005	4.05 ± 0.38	0.964 ± 0.002	3.898 ± 0.008	4.009 ± 0.009	5.19 ± 0.05	$\mathbf{0.957 \pm 0.000}$

Table 6: **Experimental results on the organoid drug-screen dataset** for population prediction of treatment response across ***replica-2*** populations. Results shown in this table are for **dim** $= \mathbf{43}$.

	Train				Test			
	$\mathcal{W}_1(\downarrow)$	$\mathcal{W}_2(\downarrow)$	MMD $_{(\times 10^{-3})}$ $(\downarrow)$	$r^2(\uparrow)$	$\mathcal{W}_1(\downarrow)$	$\mathcal{W}_2(\downarrow)$	MMD $_{(\times 10^{-3})}$ $(\downarrow)$	$r^2(\uparrow)$
$d(p_0, p_1)$	4.096	4.213	8.66	0.923	4.513	4.695	19.14	0.876
$d(p_0 - \mu_0 + \tilde{\mu}_1, p_1)$	6.002	6.083	68.50	0.929	6.222	6.346	74.90	0.876
$d(p_0 - \mu_0 + \mu_1, p_1)$	3.887	3.962	1.76	0.929	–	–	–	–
FM	3.985 ± 0.054	4.115 ± 0.067	4.64 ± 0.43	0.938 ± 0.014	4.340 ± 0.078	4.564 ± 0.111	13.00 ± 0.67	0.865 ± 0.034
FM$^{w/}\mathcal{N}$	6.892 ± 0.027	7.164 ± 0.033	57.03 ± 1.00	0.655 ± 0.003	7.114 ± 0.100	7.404 ± 0.086	64.97 ± 3.79	0.613 ± 0.008
CGFM	$\mathbf{3.882 \pm 0.019}$	3.999 ± 0.020	$\mathbf{3.16 \pm 0.59}$	0.952 ± 0.004	4.443 ± 0.033	4.621 ± 0.041	17.00 ± 1.03	0.899 ± 0.008
CGFM$^{w/}\mathcal{N}$	4.313 ± 0.077	4.480 ± 0.081	11.51 ± 1.96	0.918 ± 0.004	7.135 ± 0.045	7.390 ± 0.037	79.78 ± 4.67	0.637 ± 0.010
ICNN	4.308 ± 0.034	4.413 ± 0.036	7.07 ± 0.13	0.929 ± 0.006	4.488 ± 0.035	4.665 ± 0.038	17.60 ± 0.55	0.884 ± 0.002
MFM$^{w/}\mathcal{N}$ ($k = 0$)	3.982 ± 0.014	4.095 ± 0.015	5.04 ± 0.36	0.951 ± 0.002	4.177 ± 0.042	4.355 ± 0.048	10.53 ± 0.59	0.911 ± 0.001
MFM$^{w/}\mathcal{N}$ ($k = 10$)	4.006 ± 0.008	4.119 ± 0.012	5.13 ± 0.30	0.948 ± 0.001	4.156 ± 0.065	4.324 ± 0.067	9.58 ± 1.63	0.912 ± 0.003
MFM$^{w/}\mathcal{N}$ ($k = 50$)	3.982 ± 0.018	4.095 ± 0.016	4.74 ± 0.21	0.951 ± 0.002	4.153 ± 0.069	4.324 ± 0.070	9.63 ± 1.45	0.912 ± 0.002
MFM$^{w/}\mathcal{N}$ ($k = 100$)	4.004 ± 0.012	4.119 ± 0.014	5.19 ± 0.43	0.949 ± 0.002	4.166 ± 0.001	4.341 ± 0.003	9.52 ± 0.33	0.915 ± 0.005
MFM ($k = 0$)	3.905 ± 0.005	4.012 ± 0.006	4.18 ± 0.25	$\mathbf{0.958 \pm 0.001}$	4.209 ± 0.007	4.380 ± 0.012	12.34 ± 0.50	$\mathbf{0.918 \pm 0.002}$
MFM ($k = 10$)	3.896 ± 0.033	4.005 ± 0.036	3.89 ± 0.44	0.957 ± 0.005	4.216 ± 0.090	4.395 ± 0.098	11.99 ± 2.36	0.917 ± 0.005
MFM ($k = 50$)	3.902 ± 0.018	4.008 ± 0.022	4.20 ± 0.17	$\mathbf{0.958 \pm 0.000}$	4.214 ± 0.017	4.396 ± 0.020	12.09 ± 0.75	0.916 ± 0.002
MFM ($k = 100$)	3.884 ± 0.039	$\mathbf{3.986 \pm 0.044}$	3.77 ± 0.49	0.955 ± 0.001	$\mathbf{4.100 \pm 0.093}$	$\mathbf{4.269 \pm 0.104}$	$\mathbf{8.96 \pm 1.88}$	0.917 ± 0.004

Table 7: **Experimental results on the organoid drug-screen dataset** for population prediction of treatment response across left-out ***patient*** populations. Results shown in this table are for **dim** $= \mathbf{43}$. We report the mean and standard deviations across metrics computed over 3 patient splits.

	Train				Test			
	$\mathcal{W}_1(\downarrow)$	$\mathcal{W}_2(\downarrow)$	MMD $_{(\times 10^{-3})}$ $(\downarrow)$	$r^2(\uparrow)$	$\mathcal{W}_1(\downarrow)$	$\mathcal{W}_2(\downarrow)$	MMD $_{(\times 10^{-3})}$ $(\downarrow)$	$r^2(\uparrow)$
$d(p_0, p_1)$	4.128 ± 0.069	4.254 ± 0.076	9.437 ± 1.209	0.937 ± 0.007	4.175 ± 0.135	4.303 ± 0.174	12.107 ± 2.065	0.902 ± 0.006
$d(p_0 - \mu_0 + \tilde{\mu}_1, p_1)$	6.007 ± 0.027	6.085 ± 0.034	68.394 ± 0.993	0.937 ± 0.007	6.158 ± 0.239	6.235 ± 0.229	77.741 ± 10.723	0.902 ± 0.006
$d(p_0 - \mu_0 + \mu_1, p_1)$	3.905 ± 0.042	3.987 ± 0.042	2.009 ± 0.223	0.937 ± 0.007	–	–	–	–
FM	3.950 ± 0.055	4.087 ± 0.056	4.617 ± 1.440	0.936 ± 0.006	4.171 ± 0.107	4.315 ± 0.142	10.950 ± 1.980	0.897 ± 0.023
CGFM	3.868 ± 0.034	3.994 ± 0.037	3.612 ± 0.398	0.955 ± 0.009	4.189 ± 0.088	4.321 ± 0.119	11.568 ± 0.955	0.914 ± 0.008
ICNN	4.394 ± 0.477	4.508 ± 0.518	7.802 ± 3.373	0.914 ± 0.092	4.157 ± 0.168	4.282 ± 0.213	11.183 ± 2.511	0.904 ± 0.005
MFM ($k = 0$)	3.860 ± 0.064	3.973 ± 0.076	$\mathbf{3.496 \pm 0.869}$	0.966 ± 0.005	4.135 ± 0.094	4.268 ± 0.128	10.182 ± 1.275	0.918 ± 0.007
MFM ($k = 10$)	$\mathbf{3.853 \pm 0.062}$	$\mathbf{3.963 \pm 0.067}$	3.551 ± 0.916	$\mathbf{0.968 \pm 0.005}$	4.112 ± 0.086	4.243 ± 0.121	9.896 ± 0.993	0.925 ± 0.008
MFM ($k = 50$)	3.863 ± 0.042	3.980 ± 0.053	3.640 ± 0.640	0.966 ± 0.006	$\mathbf{4.087 \pm 0.122}$	$\mathbf{4.218 \pm 0.160}$	$\mathbf{9.262 \pm 1.558}$	0.926 ± 0.007
MFM ($k = 100$)	3.876 ± 0.055	3.990 ± 0.062	3.646 ± 0.910	0.965 ± 0.004	4.112 ± 0.148	4.244 ± 0.186	9.628 ± 2.078	$\mathbf{0.931 \pm 0.002}$

Table 8: **Extended experimental results on the organoid drug-screen dataset** for population prediction of treatment response across ***replicas-1*** populations. Results shown in this table are for **dim = 43**. Here we show results for the individual cell cultures: **Fibroblasts**, patient derived organoids (**PDOs**), and patient derived organoids with Fibroblasts (**PDOFs**), respectively. We consider 4 settings for MFM with varying nearest-neighbours parameter ($k = 0, 10, 50, 100$).

Fibroblasts

	Train				Test			
	$\mathcal{W}_1(\downarrow)$	$\mathcal{W}_2(\downarrow)$	MMD $_{(\times 10^{-3})}$ $(\downarrow)$	$r^2(\uparrow)$	$\mathcal{W}_1(\downarrow)$	$\mathcal{W}_2(\downarrow)$	MMD $_{(\times 10^{-3})}$ $(\downarrow)$	$r^2(\uparrow)$
FM	3.748 ± 0.030	3.838 ± 0.047	2.17 ± 0.19	0.937 ± 0.020	3.730 ± 0.018	3.816 ± 0.031	2.29 ± 0.11	0.948 ± 0.014
FM$^{w/}\mathcal{N}$	6.701 ± 0.093	7.045 ± 0.094	51.55 ± 3.16	0.621 ± 0.007	6.548 ± 0.115	6.885 ± 0.113	48.95 ± 3.69	0.647 ± 0.007
CGFM	$\mathbf{3.620 \pm 0.004}$	$\mathbf{3.674 \pm 0.004}$	$\mathbf{0.86 \pm 0.07}$	0.979 ± 0.001	3.732 ± 0.014	3.797 ± 0.015	3.26 ± 0.07	0.959 ± 0.004
CGFM$^{w/}\mathcal{N}$	3.671 ± 0.011	3.725 ± 0.011	0.96 ± 0.07	$\mathbf{0.980 \pm 0.001}$	5.844 ± 0.192	6.148 ± 0.204	39.80 ± 3.07	0.695 ± 0.007
ICNN	4.075 ± 0.009	4.145 ± 0.008	4.47 ± 0.04	0.925 ± 0.006	3.784 ± 0.007	3.848 ± 0.007	3.95 ± 0.07	0.952 ± 0.005
MFM$^{w/}\mathcal{N}$ ($k = 0$)	3.745 ± 0.034	3.808 ± 0.045	1.73 ± 0.27	0.968 ± 0.010	3.730 ± 0.036	3.792 ± 0.045	2.02 ± 0.29	0.967 ± 0.011
MFM$^{w/}\mathcal{N}$ ($k = 10$)	3.718 ± 0.023	3.774 ± 0.023	1.48 ± 0.22	0.976 ± 0.002	3.719 ± 0.023	3.777 ± 0.023	2.15 ± 0.25	0.973 ± 0.000
MFM$^{w/}\mathcal{N}$ ($k = 50$)	3.716 ± 0.027	3.771 ± 0.027	1.55 ± 0.32	0.976 ± 0.001	3.717 ± 0.032	3.773 ± 0.031	2.13 ± 0.45	0.976 ± 0.001
MFM$^{w/}\mathcal{N}$ ($k = 100$)	3.712 ± 0.013	3.767 ± 0.012	1.37 ± 0.17	0.977 ± 0.002	3.708 ± 0.011	3.764 ± 0.011	1.81 ± 0.29	0.975 ± 0.002
MFM ($k = 0$)	3.667 ± 0.027	3.722 ± 0.028	1.43 ± 0.18	0.975 ± 0.002	$\mathbf{3.661 \pm 0.027}$	3.719 ± 0.027	$\mathbf{1.74 \pm 0.08}$	$\mathbf{0.976 \pm 0.001}$
MFM ($k = 10$)	3.679 ± 0.032	3.734 ± 0.032	1.69 ± 0.34	0.974 ± 0.001	3.680 ± 0.030	3.738 ± 0.030	2.04 ± 0.28	0.975 ± 0.001
MFM ($k = 50$)	3.664 ± 0.042	3.720 ± 0.043	1.46 ± 0.25	0.974 ± 0.002	3.664 ± 0.039	$\mathbf{3.722 \pm 0.041}$	1.84 ± 0.23	0.975 ± 0.001
MFM ($k = 100$)	3.676 ± 0.005	3.732 ± 0.006	1.56 ± 0.27	0.972 ± 0.003	3.674 ± 0.001	3.731 ± 0.002	2.05 ± 0.20	$\mathbf{0.976 \pm 0.001}$

PDOs

	Train				Test			
	$\mathcal{W}_1(\downarrow)$	$\mathcal{W}_2(\downarrow)$	MMD $_{(\times 10^{-3})}$ $(\downarrow)$	$r^2(\uparrow)$	$\mathcal{W}_1(\downarrow)$	$\mathcal{W}_2(\downarrow)$	MMD $_{(\times 10^{-3})}$ $(\downarrow)$	$r^2(\uparrow)$
FM	3.910 ± 0.037	3.998 ± 0.042	3.37 ± 0.25	0.964 ± 0.009	3.825 ± 0.062	3.937 ± 0.072	3.31 ± 0.34	0.965 ± 0.012
FM$^{w/}\mathcal{N}$	7.334 ± 0.101	7.578 ± 0.103	69.40 ± 3.18	0.585 ± 0.009	7.461 ± 0.091	7.692 ± 0.094	72.49 ± 3.39	0.592 ± 0.009
CGFM	3.792 ± 0.061	3.866 ± 0.063	2.27 ± 0.59	0.979 ± 0.003	4.062 ± 0.096	4.181 ± 0.103	7.75 ± 1.45	0.950 ± 0.011
CGFM$^{w/}\mathcal{N}$	$\mathbf{3.746 \pm 0.032}$	$\mathbf{3.807 \pm 0.037}$	$\mathbf{1.19 \pm 0.29}$	$\mathbf{0.986 \pm 0.002}$	7.672 ± 0.149	7.909 ± 0.144	94.96 ± 7.33	0.602 ± 0.022
ICNN	4.533 ± 0.008	4.635 ± 0.007	11.53 ± 0.11	0.901 ± 0.005	4.152 ± 0.014	4.261 ± 0.013	8.66 ± 0.27	0.928 ± 0.004
MFM$^{w/}\mathcal{N}$ ($k = 0$)	3.788 ± 0.047	3.851 ± 0.053	1.59 ± 0.20	0.982 ± 0.002	3.741 ± 0.042	3.829 ± 0.047	$\mathbf{2.39 \pm 0.23}$	0.978 ± 0.002
MFM$^{w/}\mathcal{N}$ ($k = 10$)	3.822 ± 0.063	3.891 ± 0.071	1.76 ± 0.54	0.983 ± 0.001	3.785 ± 0.028	3.875 ± 0.029	2.82 ± 0.49	0.979 ± 0.002
MFM$^{w/}\mathcal{N}$ ($k = 50$)	3.794 ± 0.033	3.858 ± 0.034	1.71 ± 0.50	0.985 ± 0.001	3.775 ± 0.022	3.868 ± 0.020	2.92 ± 0.41	0.980 ± 0.002
MFM$^{w/}\mathcal{N}$ ($k = 100$)	3.783 ± 0.018	3.845 ± 0.018	1.53 ± 0.19	0.984 ± 0.002	$\mathbf{3.749 \pm 0.027}$	$\mathbf{3.835 \pm 0.028}$	2.59 ± 0.24	$\mathbf{0.981 \pm 0.002}$
MFM ($k = 0$)	3.843 ± 0.056	3.916 ± 0.066	2.65 ± 0.81	0.971 ± 0.002	3.794 ± 0.065	3.891 ± 0.074	3.34 ± 1.03	0.974 ± 0.002
MFM ($k = 10$)	3.852 ± 0.039	3.932 ± 0.045	2.80 ± 0.19	0.972 ± 0.003	3.781 ± 0.013	3.879 ± 0.015	3.45 ± 0.52	0.978 ± 0.002
MFM ($k = 50$)	3.844 ± 0.036	3.924 ± 0.033	2.51 ± 0.01	0.973 ± 0.006	3.791 ± 0.029	3.889 ± 0.026	3.33 ± 0.14	0.974 ± 0.006
MFM ($k = 100$)	3.905 ± 0.082	3.988 ± 0.088	3.87 ± 1.81	0.974 ± 0.001	3.783 ± 0.025	3.882 ± 0.029	3.33 ± 0.45	0.976 ± 0.003

PDOFs

	Train				Test			
	$\mathcal{W}_1(\downarrow)$	$\mathcal{W}_2(\downarrow)$	MMD $_{(\times 10^{-3})}$ $(\downarrow)$	$r^2(\uparrow)$	$\mathcal{W}_1(\downarrow)$	$\mathcal{W}_2(\downarrow)$	MMD $_{(\times 10^{-3})}$ $(\downarrow)$	$r^2(\uparrow)$
FM	4.117 ± 0.006	4.287 ± 0.007	$\mathbf{5.75 \pm 0.63}$	0.953 ± 0.007	4.328 ± 0.057	4.514 ± 0.064	12.11 ± 0.76	0.911 ± 0.017
FM$^{w/}\mathcal{N}$	6.686 ± 0.093	6.921 ± 0.095	52.12 ± 2.39	0.722 ± 0.006	6.906 ± 0.073	7.154 ± 0.073	59.73 ± 3.16	0.686 ± 0.009
CGFM	4.180 ± 0.130	4.385 ± 0.145	6.35 ± 2.13	0.933 ± 0.020	4.467 ± 0.080	4.653 ± 0.083	15.51 ± 0.93	0.906 ± 0.004
CGFM$^{w/}\mathcal{N}$	5.150 ± 0.046	5.499 ± 0.049	23.79 ± 1.21	0.843 ± 0.005	7.039 ± 0.147	7.285 ± 0.153	78.96 ± 6.83	0.700 ± 0.020
ICNN	4.434 ± 0.006	4.582 ± 0.006	5.69 ± 0.05	0.954 ± 0.002	4.552 ± 0.005	4.735 ± 0.006	17.02 ± 0.22	0.898 ± 0.005
MFM$^{w/}\mathcal{N}$ ($k = 0$)	4.287 ± 0.019	4.482 ± 0.015	8.41 ± 0.93	0.928 ± 0.011	4.218 ± 0.016	4.385 ± 0.016	$\mathbf{8.62 \pm 0.24}$	0.905 ± 0.002
MFM$^{w/}\mathcal{N}$ ($k = 10$)	4.372 ± 0.045	4.582 ± 0.048	9.70 ± 1.33	0.923 ± 0.007	4.326 ± 0.052	4.501 ± 0.057	10.86 ± 0.60	0.904 ± 0.004
MFM$^{w/}\mathcal{N}$ ($k = 50$)	4.356 ± 0.036	4.566 ± 0.040	9.72 ± 0.68	0.922 ± 0.006	4.291 ± 0.014	4.469 ± 0.018	10.14 ± 0.60	0.905 ± 0.001
MFM$^{w/}\mathcal{N}$ ($k = 100$)	4.295 ± 0.026	4.495 ± 0.033	8.55 ± 0.56	0.932 ± 0.007	4.251 ± 0.071	4.423 ± 0.080	9.44 ± 1.51	0.905 ± 0.003
MFM ($k = 0$)	$\mathbf{4.111 \pm 0.016}$	$\mathbf{4.281 \pm 0.023}$	6.04 ± 0.26	$\mathbf{0.954 \pm 0.009}$	$\mathbf{4.184 \pm 0.022}$	$\mathbf{4.359 \pm 0.022}$	8.96 ± 0.64	0.915 ± 0.003
MFM ($k = 10$)	4.158 ± 0.028	4.332 ± 0.029	6.97 ± 0.80	0.946 ± 0.006	4.235 ± 0.019	4.418 ± 0.022	9.89 ± 0.81	0.914 ± 0.002
MFM ($k = 50$)	4.155 ± 0.035	4.330 ± 0.015	6.80 ± 0.98	0.943 ± 0.002	4.245 ± 0.044	4.429 ± 0.035	10.00 ± 0.28	0.912 ± 0.001
MFM ($k = 100$)	4.137 ± 0.048	4.305 ± 0.068	6.73 ± 0.42	0.945 ± 0.002	4.236 ± 0.001	4.412 ± 0.004	10.18 ± 0.10	$\mathbf{0.918 \pm 0.005}$

Table 9: **Extended experimental results on the organoid drug-screen dataset** for population prediction of treatment response across ***replicas-2*** populations. Results shown in this table are for **dim = 43**. Here we show results for the individual cell cultures: **Fibroblasts**, patient derived organoids (**PDOs**), and patient derived organoids with Fibroblasts (**PDOFs**), respectively. We consider 4 settings for MFM with varying nearest-neighbours parameter ($k = 0, 10, 50, 100$).

Fibroblasts

	Train				Test			
	$\mathcal{W}_1(\downarrow)$	$\mathcal{W}_2(\downarrow)$	MMD $_{(\times 10^{-3})}$ $(\downarrow)$	$r^2(\uparrow)$	$\mathcal{W}_1(\downarrow)$	$\mathcal{W}_2(\downarrow)$	MMD $_{(\times 10^{-3})}$ $(\downarrow)$	$r^2(\uparrow)$
FM	3.694 ± 0.018	3.775 ± 0.023	1.74 ± 0.20	0.951 ± 0.007	3.646 ± 0.163	3.805 ± 0.231	3.67 ± 1.46	0.867 ± 0.070
FM$^{w/}\mathcal{N}$	6.790 ± 0.109	7.124 ± 0.120	54.97 ± 1.68	0.661 ± 0.005	6.905 ± 0.092	7.252 ± 0.100	64.40 ± 1.31	0.607 ± 0.012
CGFM	$\mathbf{3.616 \pm 0.031}$	$\mathbf{3.671 \pm 0.032}$	$\mathbf{0.90 \pm 0.13}$	0.980 ± 0.003	3.530 ± 0.011	3.584 ± 0.010	4.37 ± 0.55	0.975 ± 0.003
CGFM$^{w/}\mathcal{N}$	3.663 ± 0.011	3.718 ± 0.011	0.93 ± 0.13	$\mathbf{0.982 \pm 0.001}$	5.813 ± 0.372	6.136 ± 0.398	44.44 ± 8.18	0.688 ± 0.026
ICNN	4.054 ± 0.03	4.124 ± 0.033	4.43 ± 0.14	0.925 ± 0.007	3.477 ± 0.031	3.534 ± 0.033	3.67 ± 0.17	0.969 ± 0.004
MFM$^{w/}\mathcal{N}$ ($k = 0$)	3.671 ± 0.021	3.726 ± 0.022	1.17 ± 0.17	0.979 ± 0.001	3.503 ± 0.030	3.555 ± 0.031	$\mathbf{1.68 \pm 0.11}$	0.975 ± 0.001
MFM$^{w/}\mathcal{N}$ ($k = 10$)	3.691 ± 0.015	3.748 ± 0.014	1.18 ± 0.14	0.978 ± 0.000	3.526 ± 0.024	3.582 ± 0.022	1.86 ± 0.36	0.972 ± 0.001
MFM$^{w/}\mathcal{N}$ ($k = 50$)	3.678 ± 0.017	3.733 ± 0.017	1.15 ± 0.12	0.979 ± 0.001	3.515 ± 0.029	3.567 ± 0.029	2.07 ± 0.63	0.977 ± 0.002
MFM$^{w/}\mathcal{N}$ ($k = 100$)	3.706 ± 0.020	3.764 ± 0.022	1.44 ± 0.08	0.977 ± 0.001	3.536 ± 0.036	3.590 ± 0.037	1.90 ± 0.23	0.975 ± 0.003
MFM ($k = 0$)	3.626 ± 0.012	3.682 ± 0.013	1.20 ± 0.04	0.979 ± 0.000	3.451 ± 0.020	3.505 ± 0.021	2.55 ± 0.42	0.981 ± 0.002
MFM ($k = 10$)	3.624 ± 0.025	3.678 ± 0.026	1.04 ± 0.17	0.981 ± 0.000	3.451 ± 0.037	3.504 ± 0.038	2.47 ± 0.56	$\mathbf{0.982 \pm 0.001}$
MFM ($k = 50$)	3.639 ± 0.016	3.694 ± 0.016	1.71 ± 0.20	0.979 ± 0.001	$\mathbf{3.443 \pm 0.035}$	$\mathbf{3.497 \pm 0.037}$	2.89 ± 1.85	0.981 ± 0.001
MFM ($k = 100$)	3.654 ± 0.005	3.712 ± 0.005	1.56 ± 0.20	0.978 ± 0.000	3.480 ± 0.027	3.534 ± 0.029	3.32 ± 0.49	0.980 ± 0.001

PDOs

	Train				Test			
	$\mathcal{W}_1(\downarrow)$	$\mathcal{W}_2(\downarrow)$	MMD $_{(\times 10^{-3})}$ $(\downarrow)$	$r^2(\uparrow)$	$\mathcal{W}_1(\downarrow)$	$\mathcal{W}_2(\downarrow)$	MMD $_{(\times 10^{-3})}$ $(\downarrow)$	$r^2(\uparrow)$
FM	3.948 ± 0.092	4.050 ± 0.116	3.96 ± 0.68	0.942 ± 0.023	4.043 ± 0.100	4.174 ± 0.131	4.90 ± 0.50	0.931 ± 0.033
FM$^{w/}\mathcal{N}$	7.109 ± 0.028	7.351 ± 0.030	63.42 ± 1.97	0.626 ± 0.008	7.697 ± 0.212	7.976 ± 0.182	77.52 ± 7.33	0.519 ± 0.002
CGFM	3.750 ± 0.013	3.815 ± 0.017	1.77 ± 0.71	0.977 ± 0.004	4.225 ± 0.146	4.333 ± 0.171	9.50 ± 3.14	0.933 ± 0.016
CGFM$^{w/}\mathcal{N}$	$\mathbf{3.728 \pm 0.024}$	$\mathbf{3.787 \pm 0.026}$	$\mathbf{1.14 \pm 0.16}$	$\mathbf{0.985 \pm 0.000}$	8.333 ± 0.309	8.534 ± 0.298	114.45 ± 14.17	0.508 ± 0.016
ICNN	4.476 ± 0.039	4.58 ± 0.041	11.17 ± 0.05	0.906 ± 0.007	4.413 ± 0.048	4.519 ± 0.051	11.18 ± 0.05	0.886 ± 0.004
MFM$^{w/}\mathcal{N}$ ($k = 0$)	3.775 ± 0.012	3.837 ± 0.013	1.75 ± 0.05	0.979 ± 0.001	$\mathbf{3.855 \pm 0.075}$	3.944 ± 0.087	3.73 ± 0.79	0.978 ± 0.003
MFM$^{w/}\mathcal{N}$ ($k = 10$)	3.799 ± 0.009	3.860 ± 0.009	1.89 ± 0.26	0.978 ± 0.001	3.877 ± 0.042	3.961 ± 0.047	3.53 ± 0.30	0.978 ± 0.003
MFM$^{w/}\mathcal{N}$ ($k = 50$)	3.800 ± 0.023	3.863 ± 0.023	1.93 ± 0.27	0.980 ± 0.001	3.857 ± 0.059	$\mathbf{3.938 \pm 0.066}$	$\mathbf{3.40 \pm 0.77}$	$\mathbf{0.981 \pm 0.001}$
MFM$^{w/}\mathcal{N}$ ($k = 100$)	3.800 ± 0.008	3.863 ± 0.010	2.05 ± 0.12	0.979 ± 0.001	3.882 ± 0.008	3.973 ± 0.009	3.80 ± 0.26	0.979 ± 0.000
MFM ($k = 0$)	3.797 ± 0.020	3.864 ± 0.020	2.40 ± 0.32	0.971 ± 0.001	3.894 ± 0.026	3.982 ± 0.037	3.87 ± 0.83	0.976 ± 0.002
MFM ($k = 10$)	3.792 ± 0.041	3.863 ± 0.049	2.29 ± 0.47	0.975 ± 0.005	3.953 ± 0.095	4.055 ± 0.115	4.80 ± 2.00	0.978 ± 0.001
MFM ($k = 50$)	3.796 ± 0.014	3.864 ± 0.016	2.50 ± 0.22	0.972 ± 0.001	4.035 ± 0.099	4.153 ± 0.116	6.40 ± 2.19	0.971 ± 0.004
MFM ($k = 100$)	3.798 ± 0.037	3.866 ± 0.041	2.54 ± 0.28	0.970 ± 0.002	3.869 ± 0.094	3.953 ± 0.113	3.49 ± 0.76	0.975 ± 0.001

PDOFs

	Train				Test			
	$\mathcal{W}_1(\downarrow)$	$\mathcal{W}_2(\downarrow)$	MMD $_{(\times 10^{-3})}$ $(\downarrow)$	$r^2(\uparrow)$	$\mathcal{W}_1(\downarrow)$	$\mathcal{W}_2(\downarrow)$	MMD $_{(\times 10^{-3})}$ $(\downarrow)$	$r^2(\uparrow)$
FM	4.313 ± 0.062	4.521 ± 0.076	8.21 ± 0.55	0.919 ± 0.014	5.330 ± 0.112	5.712 ± 0.113	30.45 ± 3.30	$\mathbf{0.798 \pm 0.002}$
FM$^{w/}\mathcal{N}$	6.778 ± 0.046	7.015 ± 0.042	52.69 ± 2.63	0.708 ± 0.004	6.741 ± 0.177	6.985 ± 0.163	52.98 ± 5.35	0.713 ± 0.014
CGFM	4.281 ± 0.065	4.511 ± 0.078	$\mathbf{6.81 \pm 1.33}$	0.898 ± 0.011	5.574 ± 0.051	5.945 ± 0.052	37.13 ± 0.40	0.790 ± 0.009
CGFM$^{w/}\mathcal{N}$	5.549 ± 0.206	5.936 ± 0.214	32.44 ± 5.58	0.787 ± 0.014	7.258 ± 0.197	7.501 ± 0.212	80.45 ± 8.02	0.713 ± 0.005
ICNN	4.393 ± 0.034	4.534 ± 0.037	5.60 ± 0.20	$\mathbf{0.956 \pm 0.003}$	5.573 ± 0.049	5.943 ± 0.056	37.96 ± 1.58	$\mathbf{0.798 \pm 0.004}$
MFM$^{w/}\mathcal{N}$ ($k = 0$)	4.500 ± 0.039	4.721 ± 0.043	12.21 ± 1.07	0.896 ± 0.005	5.173 ± 0.078	5.567 ± 0.077	26.17 ± 2.29	0.779 ± 0.003
MFM$^{w/}\mathcal{N}$ ($k = 10$)	4.528 ± 0.033	4.751 ± 0.045	12.31 ± 0.73	0.888 ± 0.003	5.065 ± 0.169	5.431 ± 0.172	23.34 ± 4.74	0.786 ± 0.005
MFM$^{w/}\mathcal{N}$ ($k = 50$)	4.468 ± 0.026	4.690 ± 0.030	11.15 ± 0.34	0.894 ± 0.006	5.088 ± 0.121	5.468 ± 0.115	23.43 ± 3.15	0.778 ± 0.005
MFM$^{w/}\mathcal{N}$ ($k = 100$)	4.508 ± 0.042	4.731 ± 0.042	12.09 ± 1.34	0.891 ± 0.006	5.082 ± 0.037	5.461 ± 0.047	23.16 ± 0.90	0.792 ± 0.015
MFM ($k = 0$)	4.293 ± 0.005	4.491 ± 0.005	8.94 ± 0.54	0.925 ± 0.002	5.283 ± 0.017	5.653 ± 0.029	30.60 ± 0.77	0.797 ± 0.005
MFM ($k = 10$)	4.273 ± 0.039	4.474 ± 0.041	8.35 ± 0.83	0.915 ± 0.009	5.244 ± 0.168	5.626 ± 0.175	28.71 ± 5.10	0.790 ± 0.016
MFM ($k = 50$)	$\mathbf{4.271 \pm 0.038}$	4.466 ± 0.049	8.40 ± 0.40	0.924 ± 0.001	5.165 ± 0.132	5.538 ± 0.140	26.97 ± 2.46	0.796 ± 0.009
MFM ($k = 100$)	4.200 ± 0.079	$\mathbf{4.381 \pm 0.095}$	7.20 ± 1.13	0.918 ± 0.004	$\mathbf{4.950 \pm 0.199}$	$\mathbf{5.321 \pm 0.219}$	$\mathbf{20.08 \pm 5.10}$	$\mathbf{0.798 \pm 0.013}$

F.1 Analysis of Population Embeddings

Here, we analyze the embedding space of the population embedding model $\varphi(p_0;\theta)$ for the synthetic letters dataset and the organoid drug-screen dataset. To do this, we compute embeddings $h = \varphi(\{x_0^j\}_{j=1}^{N'};\theta)$ for each initial population. For letters we consider 200 random rotations per letter and for the organoid drug-screen dataset we consider all population pairs (p_0, p_1). We then project the embeddings into a 2-dimensional space using uniform manifold approximation and projection (UMAP) to visualize model embeddings. We compute pairwise Euclidean distances of the samples in the projected embedding space and report these distances visually using heat maps. See Fig. 7.

On the letters dataset, the population embedding model learns to embed similar letter close silhouettes together. For example, 'A' and 'V', 'T' and 'Y' (even though 'Y' silhouettes are never seen during training), and more. Note, that in this dataset, letter silhouette populations p_0 are both corrupted with noise and rotated randomly. We conduct the same analysis for the organoid drug-screen dataset. We observe that in some sense $\varphi(p_0;\theta)$ groups populations from particular patients in a generally consistent matter to that found in Ramos Zapatero et al. (2023). Specifically, we observe patient who are chemosensitive (PDO 21, 75, 23, 27) cluster together, away from patients who are chemorefractory (PDO 5, 11, 141, 216). This illustrates the embedding are able to capture patient drug response characteristics from untreated patient sample p_0.

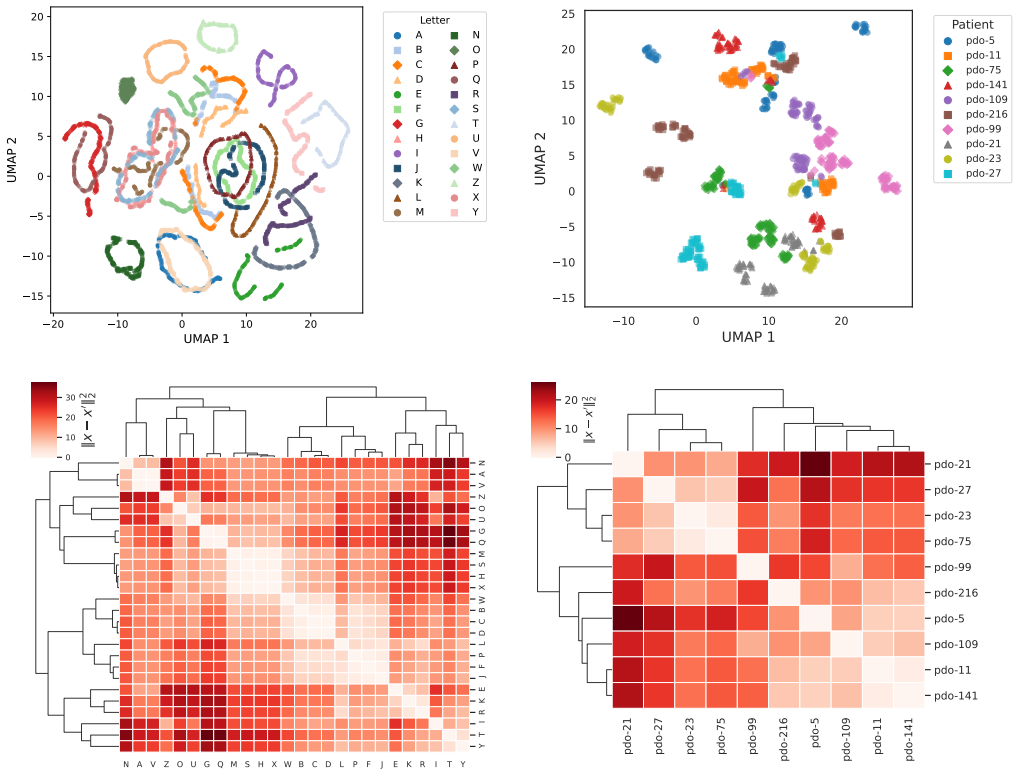

Figure 7: **Analysis of population embeddings from $\varphi(p_0;\theta)$.** We provide a UMAP visualization of population embeddings space (top) and pairwise distances of population embeddings computed in 2-dimensional space (bottom). (*Left*) analysis of letter population embeddings plotted for 200 random rotations of each noisy letter silhouette. (*Right*) analysis of patient population embeddings plotted for all control populations.

F.2 Analysis of Prediction Quality on Organoid Drug-screen Data

In this section, we analyze the predictions produced by vector field models trained with MFM. For this analysis, we look at model predictions for the three respective left-out test patient splits: PDO-21, PDO-27, PDO-75. For each control population in the patient split (specifically, for the test patient in the given split), we predict the respective treatment response. We then subset the data to look

populations in the PDOF culture and take the mean of observations in each population (for both predicted and target populations). We project the means of the predicted and target populations into 2D space using principle component analysis (PCA) across all. We fit PCA using means from source, target, and predicted samples. Lastly, we plot the target and predicted samples separately. We show the result of this analysis in Fig. 8.

We observe that for all three test patients the general structure is preserved, with the treatment, Oxaliplatin (Green), being the furthest away both for the target and ground truth datasets. This is because (as shown in the original paper by Ramos Zapatero et al. (2023)) Oxaliplatin has a large effect on these cancer cells for the PDOF subset of PDOs. We see from Fig. 8 that this is more pronounced for PDO-21 and PDO-27 than for PDO-75. Overall, this reflects the conclusions drawn from Figure 4 in the original dataset paper Ramos Zapatero et al. (2023). In this way, we are able to draw similar conclusions from the predicted populations as from the data.

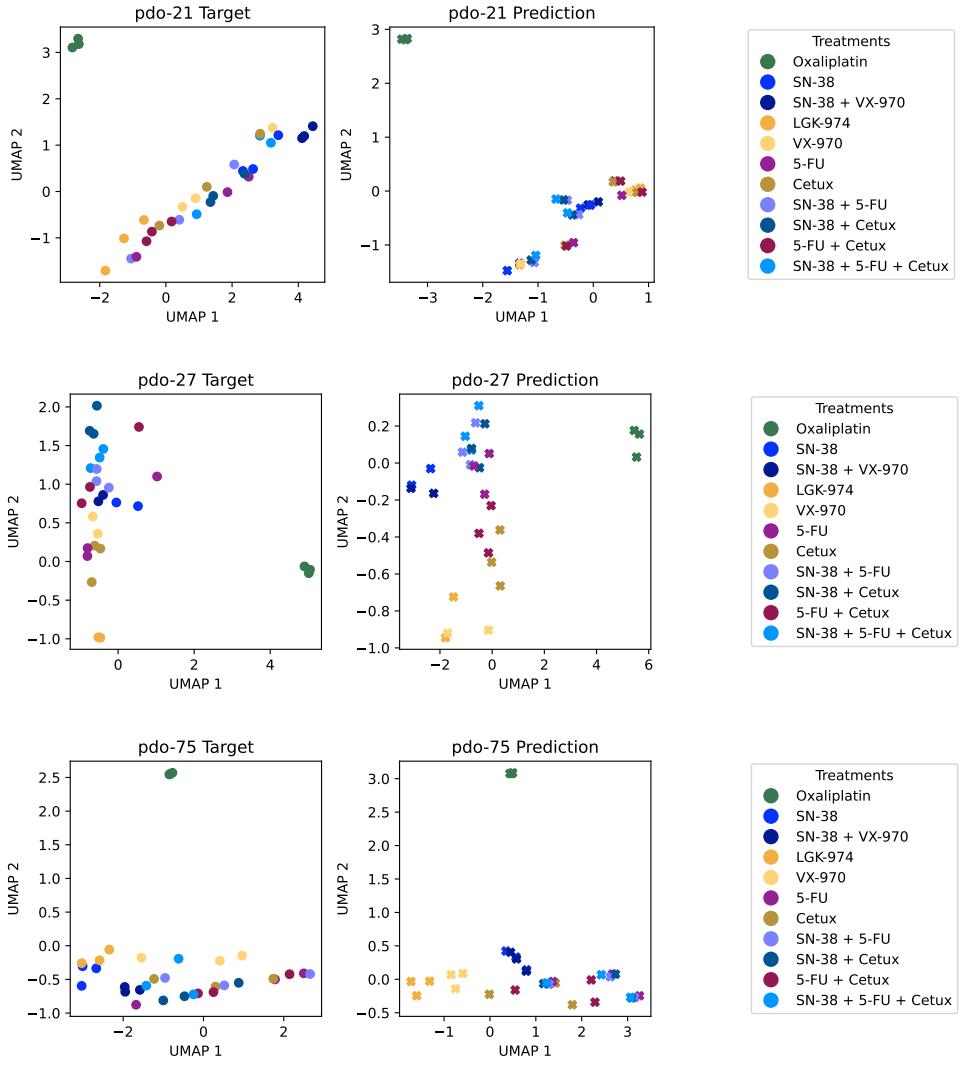

Figure 8: **Analysis of treatment-specific response prediction.** We plot population means in 2D PCA space for target populations (*Left*), predicted populations (*Middle*), and the respective treatment identifier legend (*Right*).