Meta Flow Matching: Integrating Vector Fields on the Wasserstein Manifold

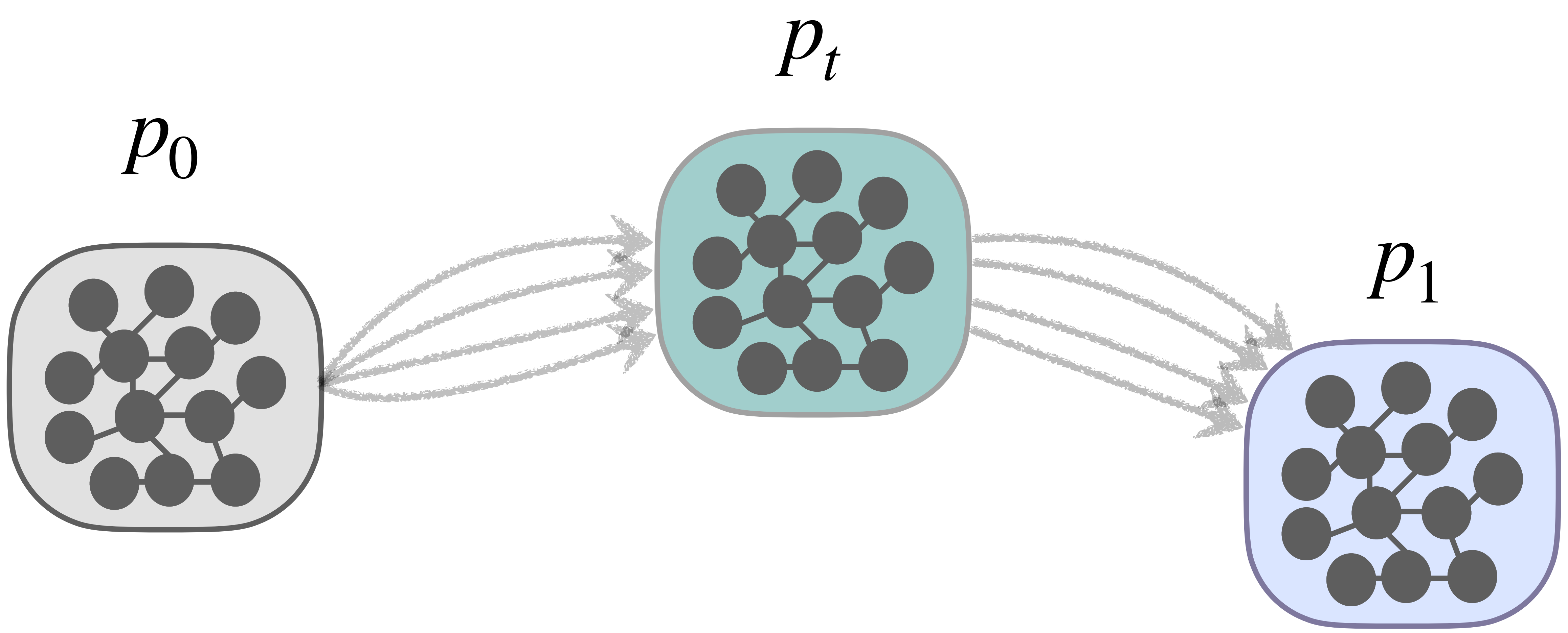
Lazar Atanackovic*, Xi Zhang*, Brandon Amos, Mathieu Blanchette, Leo J Lee,
Yoshua Bengio, Alexander Tong, Kirill Neklyudov

UNIVERSITY OF TORONTO
VECTOR INSTITUTE
Mila
Meta

Background and Motivation

In many scientific problems, we want to understand the dynamics of many-body problems, or the dynamic evolution of interacting particles

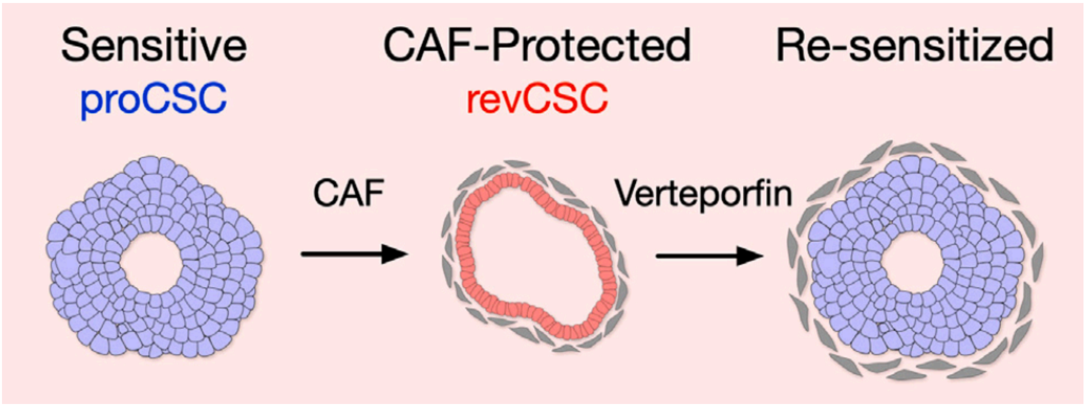
E.g. the dynamic processes ***cells*** undergo w.r.t. their environment and interactions with each other

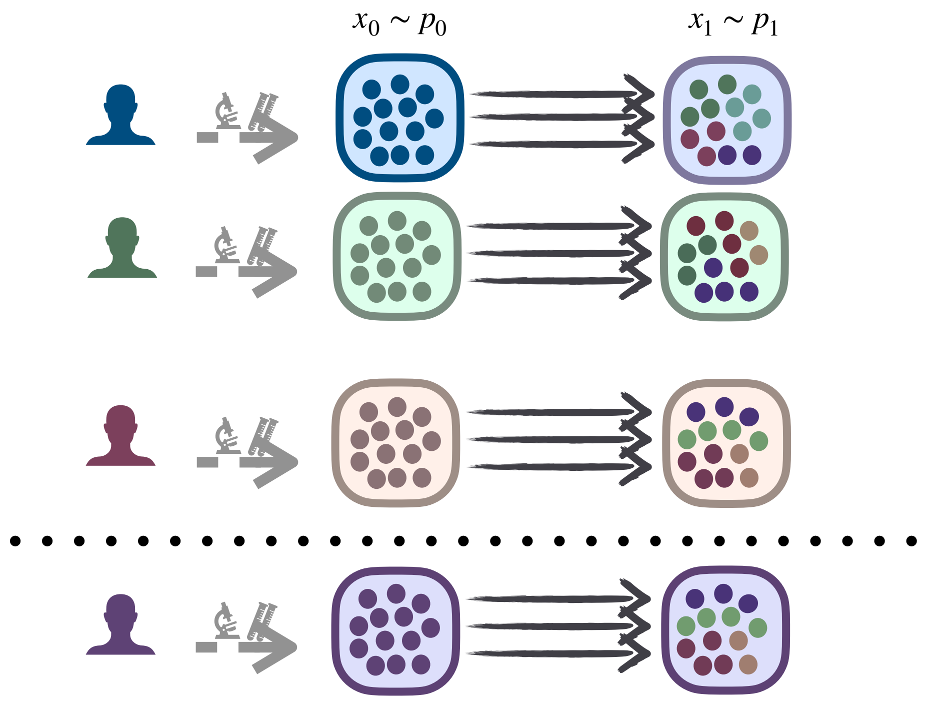


Background and Motivation

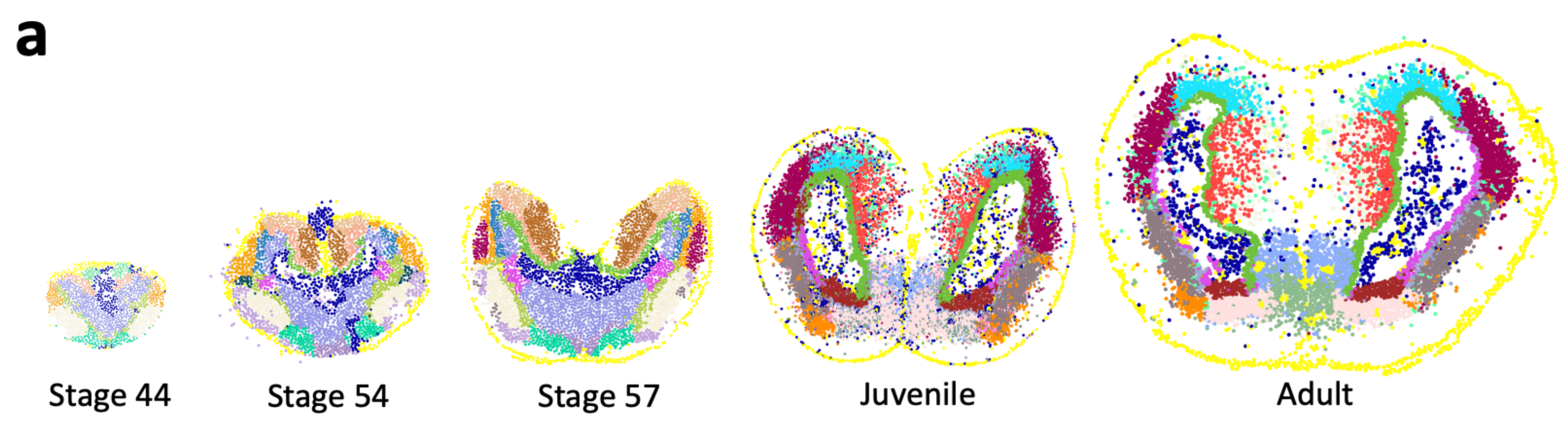
Big picture application/examples

Understanding treatment response on cancer cells (Zapatero et al, *Cell*, 2023) — ***we look at this***





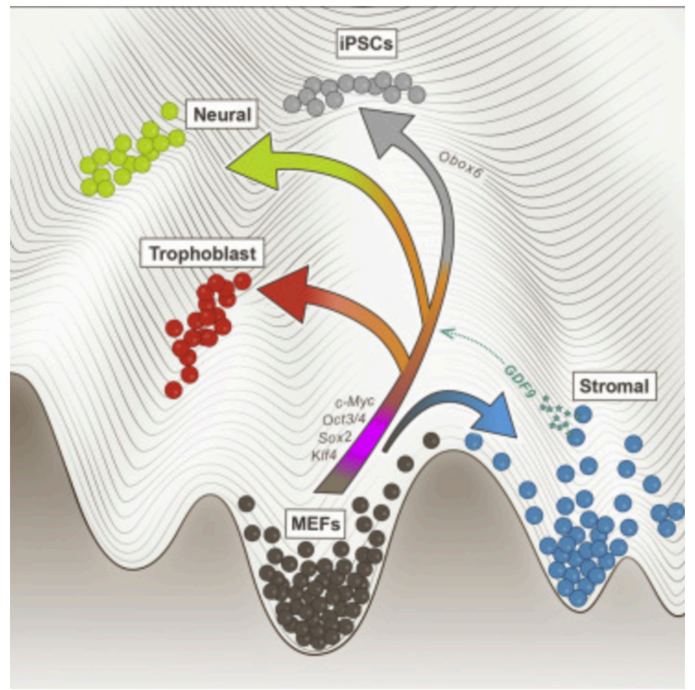
Understanding cell development in spatial transcriptome — e.g. *axolotl* brain (Halmost et al, 2024)



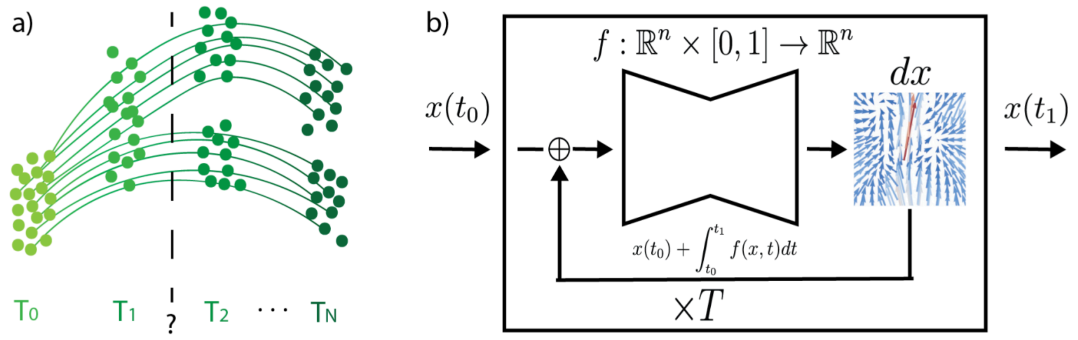
How can we model the evolution of cells while considering their interactions and population-specific responses?

Background and Motivation

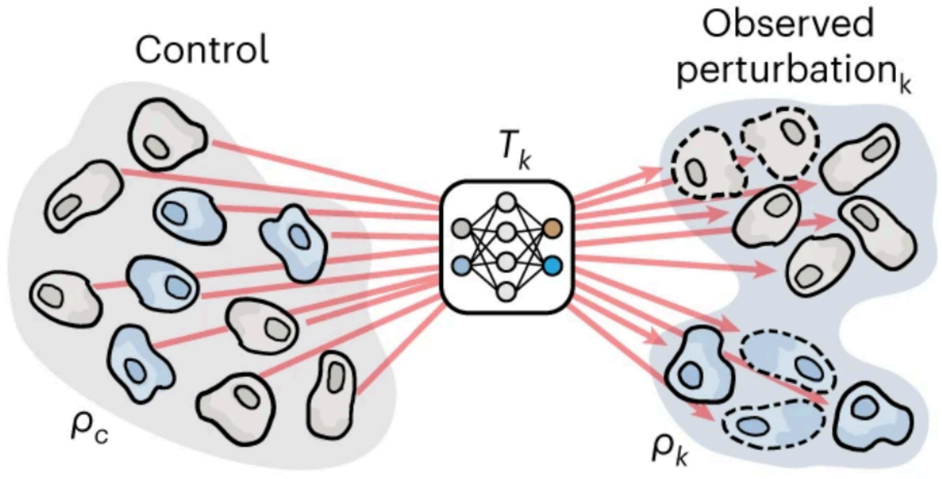
We want to model the dynamics of particles (or cells) at the population level. We have many methods that can do this.



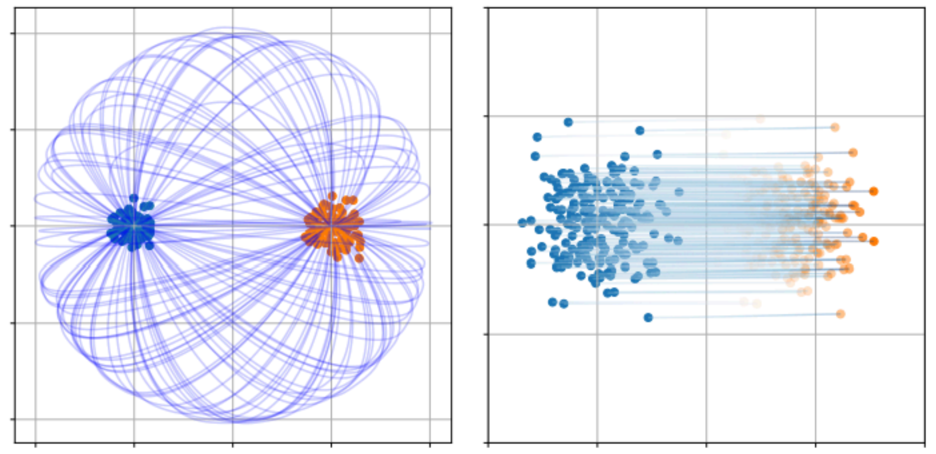
(Schiebinger et al, *Cell*, 2019)



(Tong et al, *ICML*, 2020)



(Bunne et al, *Nature Methods*, 2023)

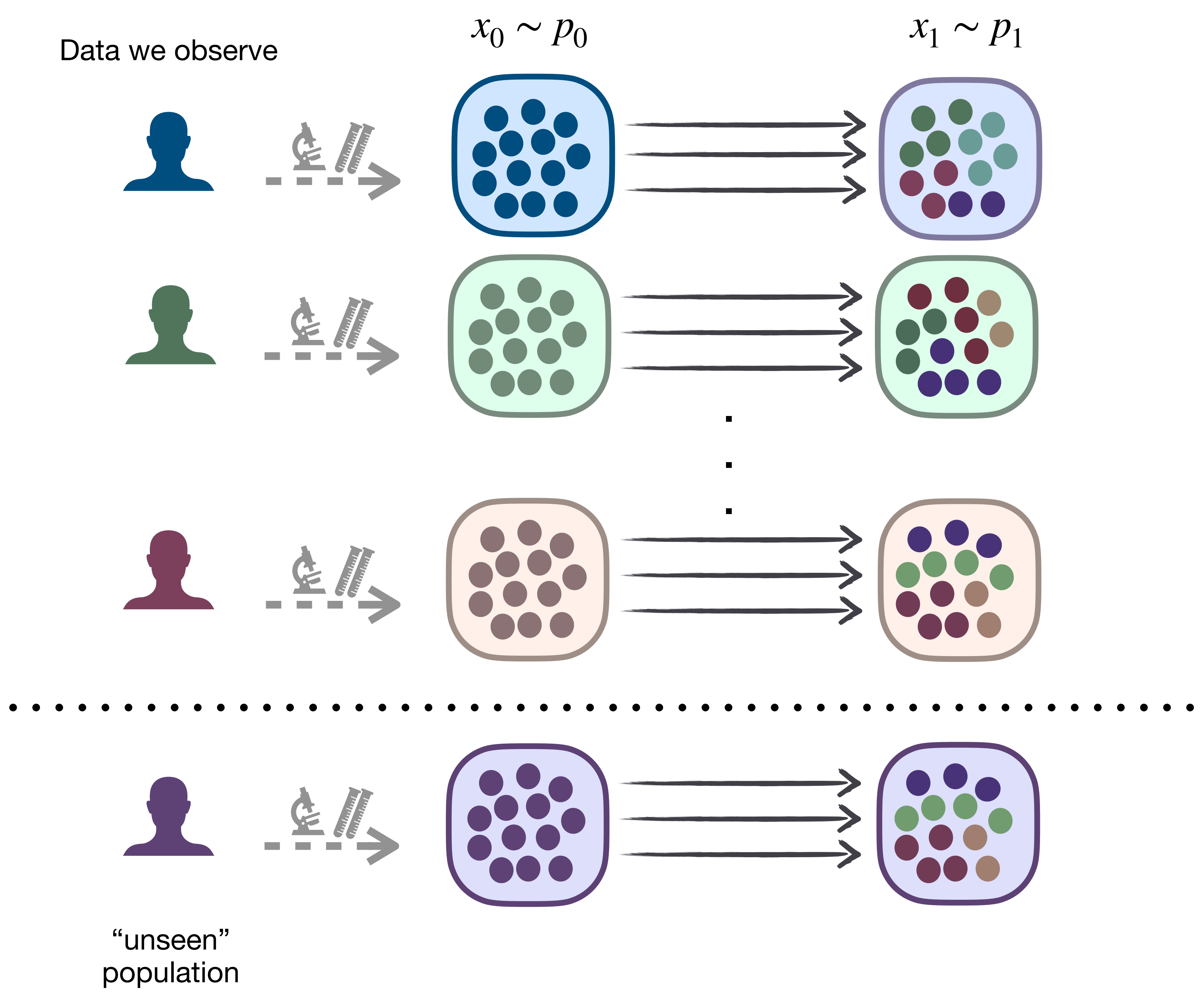


(Neklydov et al, *ICML*, 2024)

Existing methods typically only model the evolution of cells as independent particles.

Background and Motivation

We would also like a model that can generalize across measures (populations)



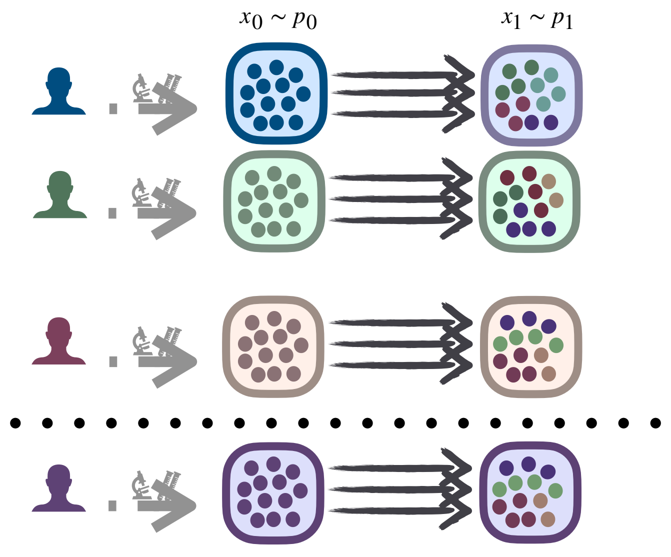
Existing methods are typically restricted to a single measure (population, patient). At best can condition on different dynamics.

Problem setup

We want a model that can:

i) model the evolution of particles while taking into account their interactions

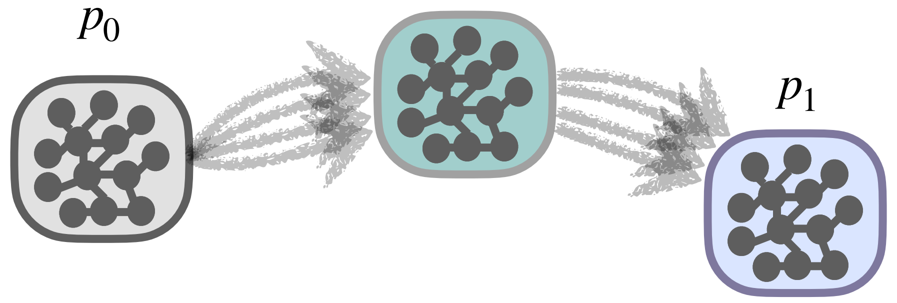
ii) generalize across *unseen* populations



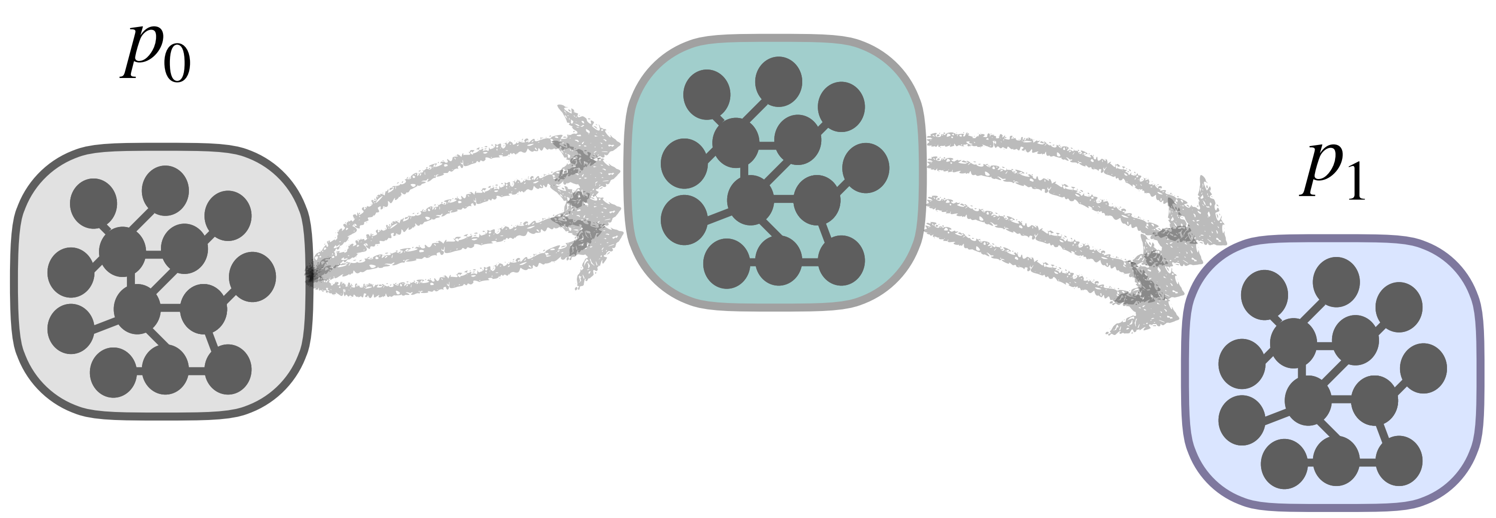
Main assumptions:

- Coupled distribution/population pairs $\{(p_0(x_0\,|\,i), p_1(x_1\,|\,i))\}_{i=1}^N$

- 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



Example: Mean-field limit of interacting particles



Consider a system of interacting particles

We can define a ***velocity*** $k(x, y) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ of the particle at point x interacting with the particle at point y

In the limit of the infinite number of particles ***one can describe their state using the density function*** $p_t(x)$

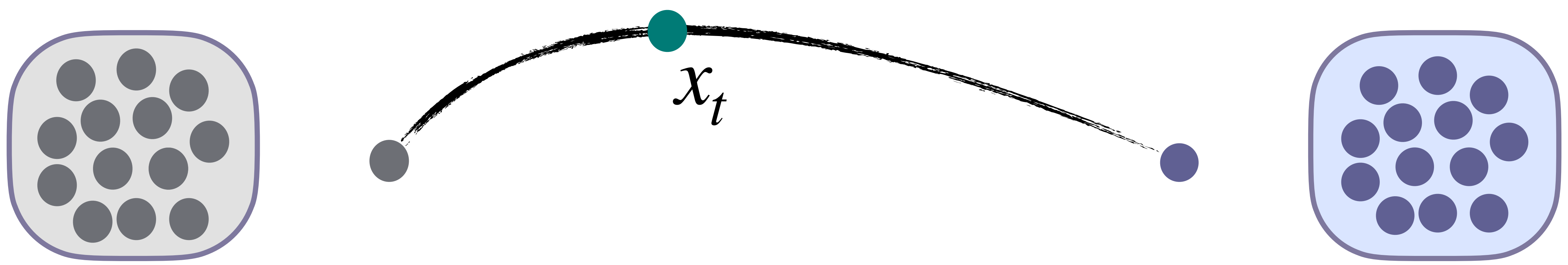
$$\frac{dx}{dt} = \mathbb{E}_{p_t(y)} k(x, y), \quad \frac{\partial p_t(x)}{\partial t} = -\langle \nabla_x, p_t(x) \mathbb{E}_{p_t(y)} k(x, y) \rangle$$

How do we train a model to learn this?

Generative Modelling via Flow Matching

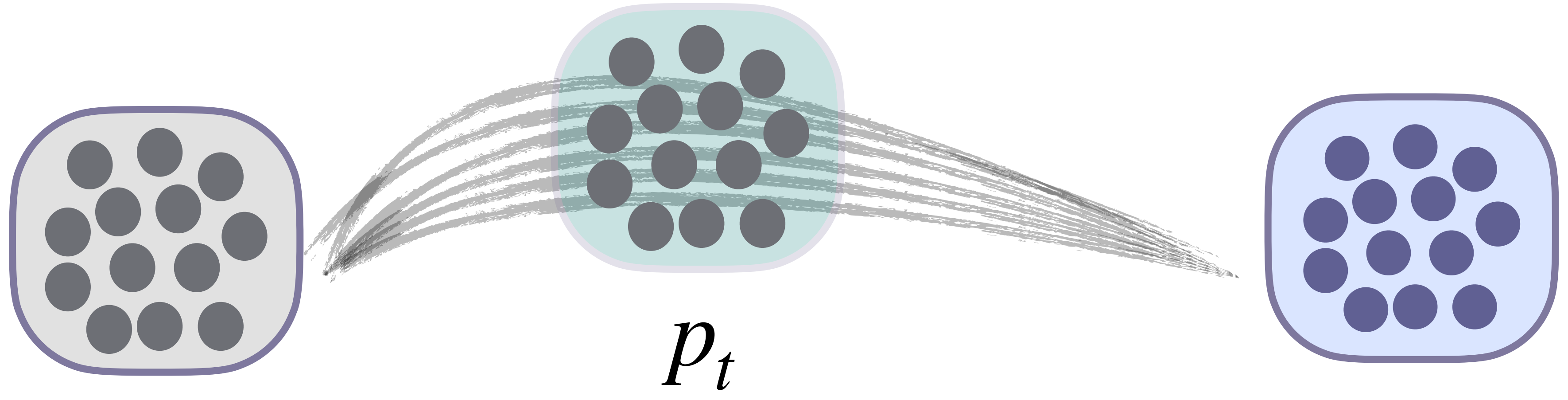
Consider a **continuous interpolation** between densities $p_0(x_0)$ and $p_1(x_1)$ — i.e. a sample x_t from the intermediate density $p_t(x_t)$ is produced as follows

$$x_t = f_t(x_0, x_1), \;\; (x_0, x_1) \sim \pi(x_0, x_1)$$



The corresponding density at t can be defined as follows

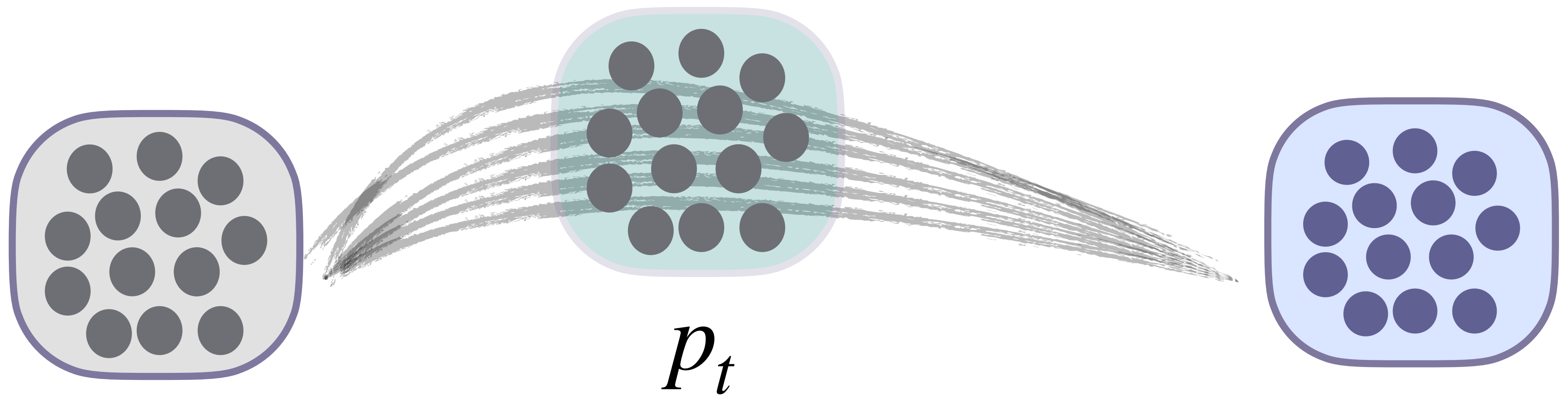
$$p_t(x) = \int dx_0 dx_1 \;\pi(x_0, x_1)\delta(x - f_t(x_0, x_1))$$



Generative Modelling via Flow Matching

An essential component of flow matching is the continuity equation, which describes the density change through a vector field $v_t^*(x)$

$$\frac{\partial p_t(x)}{\partial t} = - \langle \nabla_x, p_t(x) v_t^*(x) \rangle$$



We can approximate $v_t^*(x)$ via a parameterized function $v_t(x;\omega)$. Likewise, derive an objective/loss for learning:

$$\mathscr{L}_{FM}(\omega) = \mathbb{E}_{\pi(x_0,x_1)} \int_0^1 dt \| v_t^*(x) - v_t(x;\omega) \|^2$$

FM conditioned on entire populations

We consider a vector field model ***conditioned on an entire distribution*** p_t. We can write the continuity equation as

we focus on the setting of conditioning on p_0

$$\frac{\partial p_t(x)}{\partial t} = -\langle \nabla_x, p_t(x) v_t^*(x, p_t) \rangle$$

Then, we can aim to learn a vector field model:

$$v_t(\,\cdot\,, \varphi(\pi)) = v_t^*(\,\cdot\,, \pi)$$

where $\varphi(\pi)$ is an ***embedding model*** of π. The joint density $\pi(x_0, x_1 \mid i)$ is generated using some ***unknown measure of the conditional variables*** $i \sim p(i)$.

Intuition (dynamical systems)

Differential equation

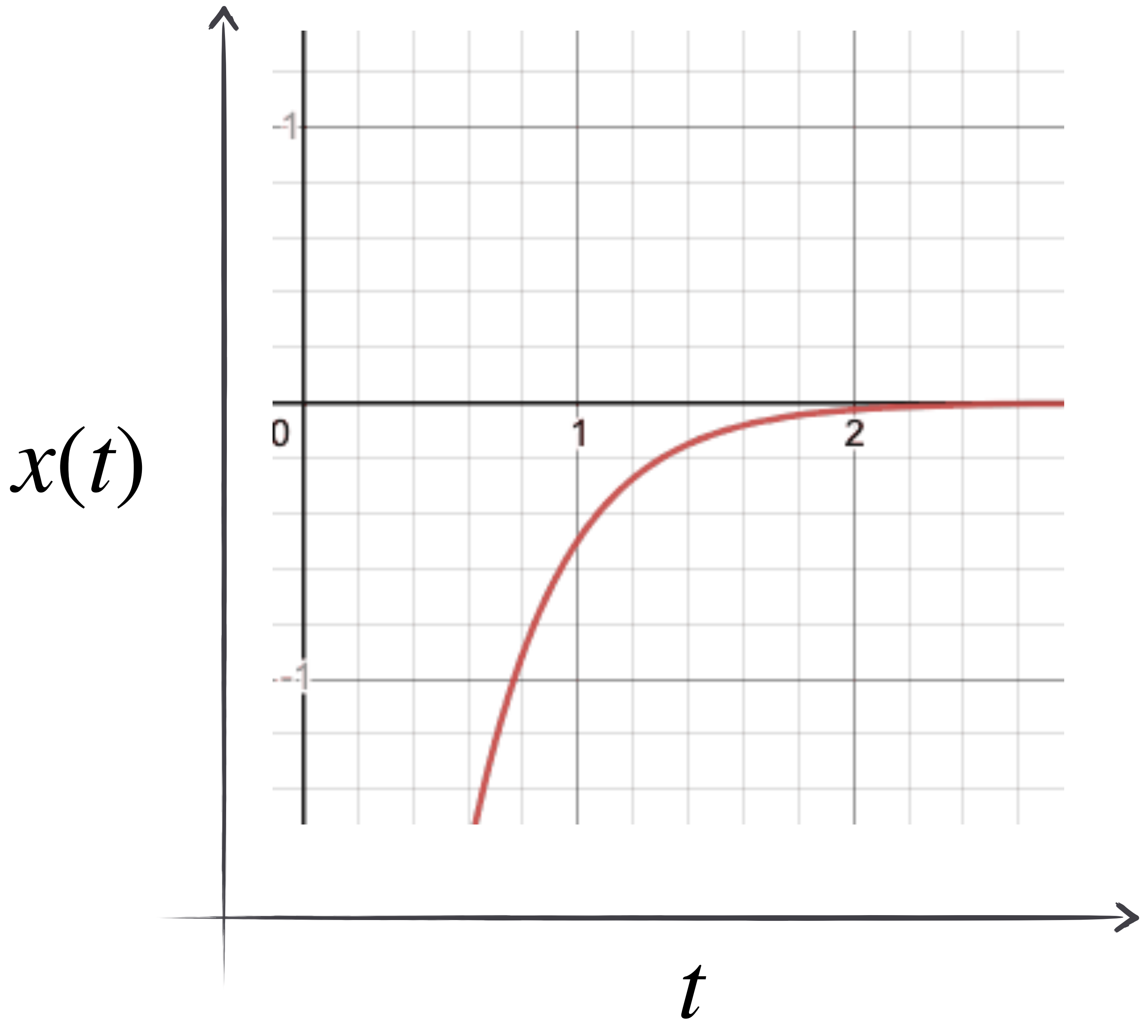
$$\frac{dx}{dt} = \mathbf{A}x$$

$\xrightarrow{\text{solution}}$

Dynamic system

$$x(t) = C \cdot e^{At}$$

For different C's we get different dynamic responses for the same system.



VECTOR INSTITUTE

Intuition (dynamical systems)

Differential equation

$$\frac{dx}{dt} = \mathbf{A}x \xrightarrow{\text{solution}} x(t) = C \cdot e^{At}$$

Dynamic system

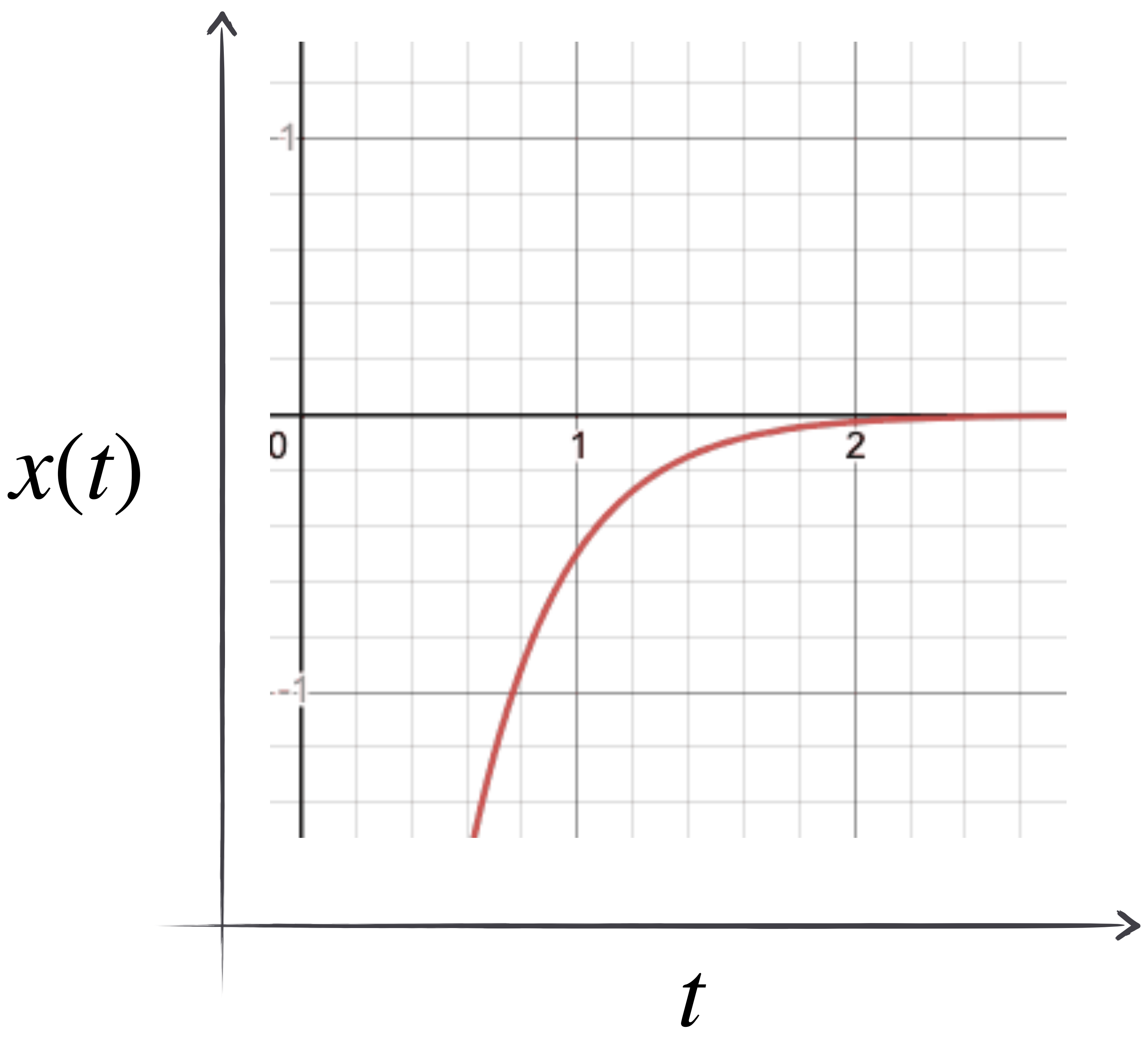
Given initial conditions of the system

$$\frac{dx}{dt} = \mathbf{A}x, \quad (x(t_0), t_0) = (1,0)$$

$$x(t) = C \cdot e^{At}$$

$$x(0) = 1 = Ce^{0 \cdot t} = C \cdot 1$$

$$C = 1$$



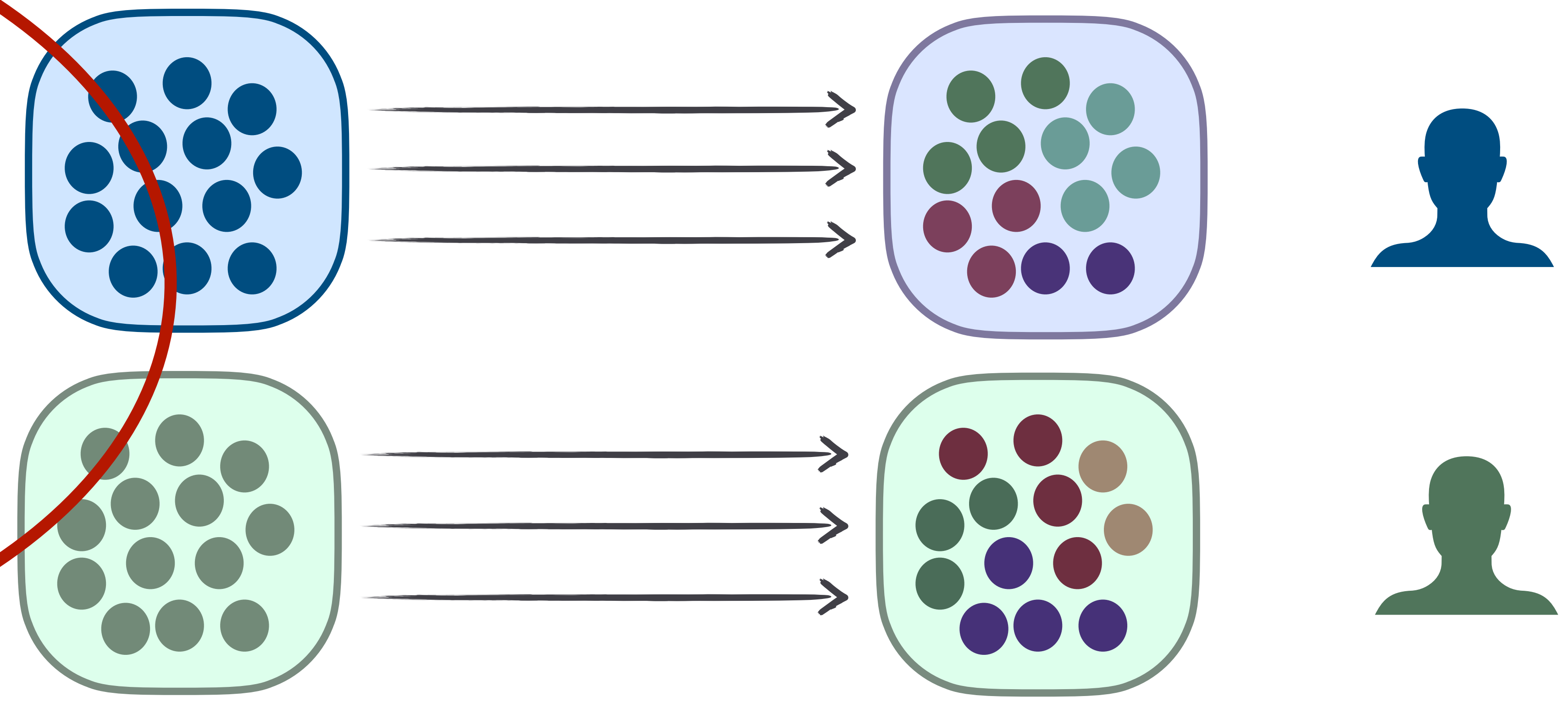
Can we design a model that conditions on the the entire environment?

Initial conditions

$$(x(t_0), t_0) = (1,0), \quad C = 1$$

$$(x(t_0), t_0) = (2,0), \quad C = 2$$

Control **Treated**



Meta Flow Matching (MFM)

Consider the data set $D = \{(\pi(x_0, x_1 \,|\, i))\}_{i=1}^{N}$.

The only information we have is samples from the i-th population ($p(i)$ is unknown). We learn embeddings for the population using a parameterized model

$$\varphi(p_0, \theta) = \varphi\left(\{x_0^j\}_{j=1}^{N_i}, \theta\right), \;\; (x_0^j, x_1^j) \sim \pi(x_0, x_1 \,|\, i)$$

Then we can write our **MFM** objective as:

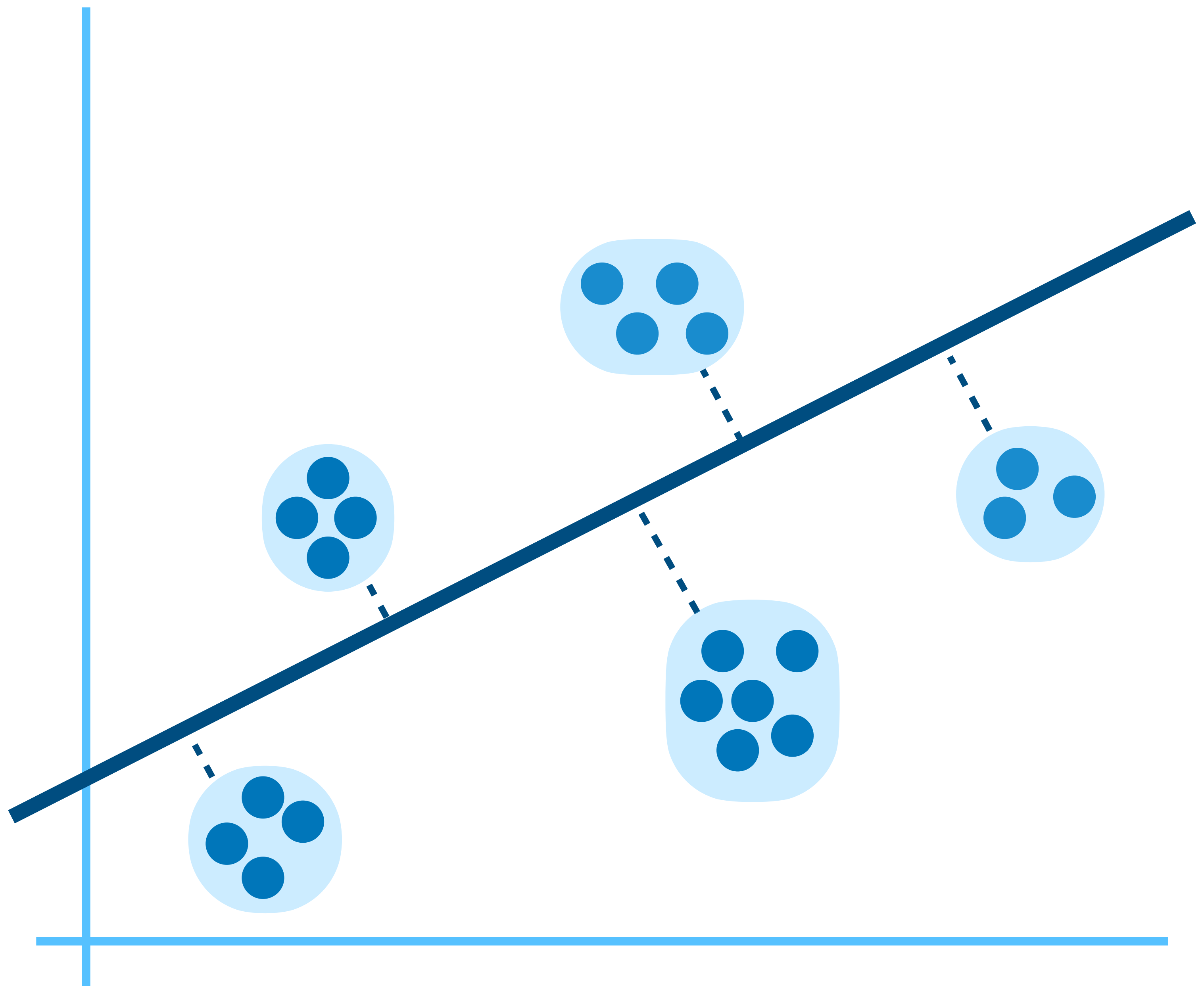
$$\mathscr{L}_{MFM}(\omega, \theta) = \mathbb{E}_{i \sim D} \mathbb{E}_{\pi(x_0, x_1 | i)} \int_0^1 dt \| \frac{\partial}{\partial t} f_t(x_0, x_1) - v_t(f_t(x_0, x_1) \,|\, \varphi(p_0; \theta); \omega) \|^2$$

Meta Flow Matching (MFM)

MFM objective

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

Intuition — you can think of this as similar to "**Wasserstein/distributional regression**" — data points are distributions rather than just single samples



MFM Training

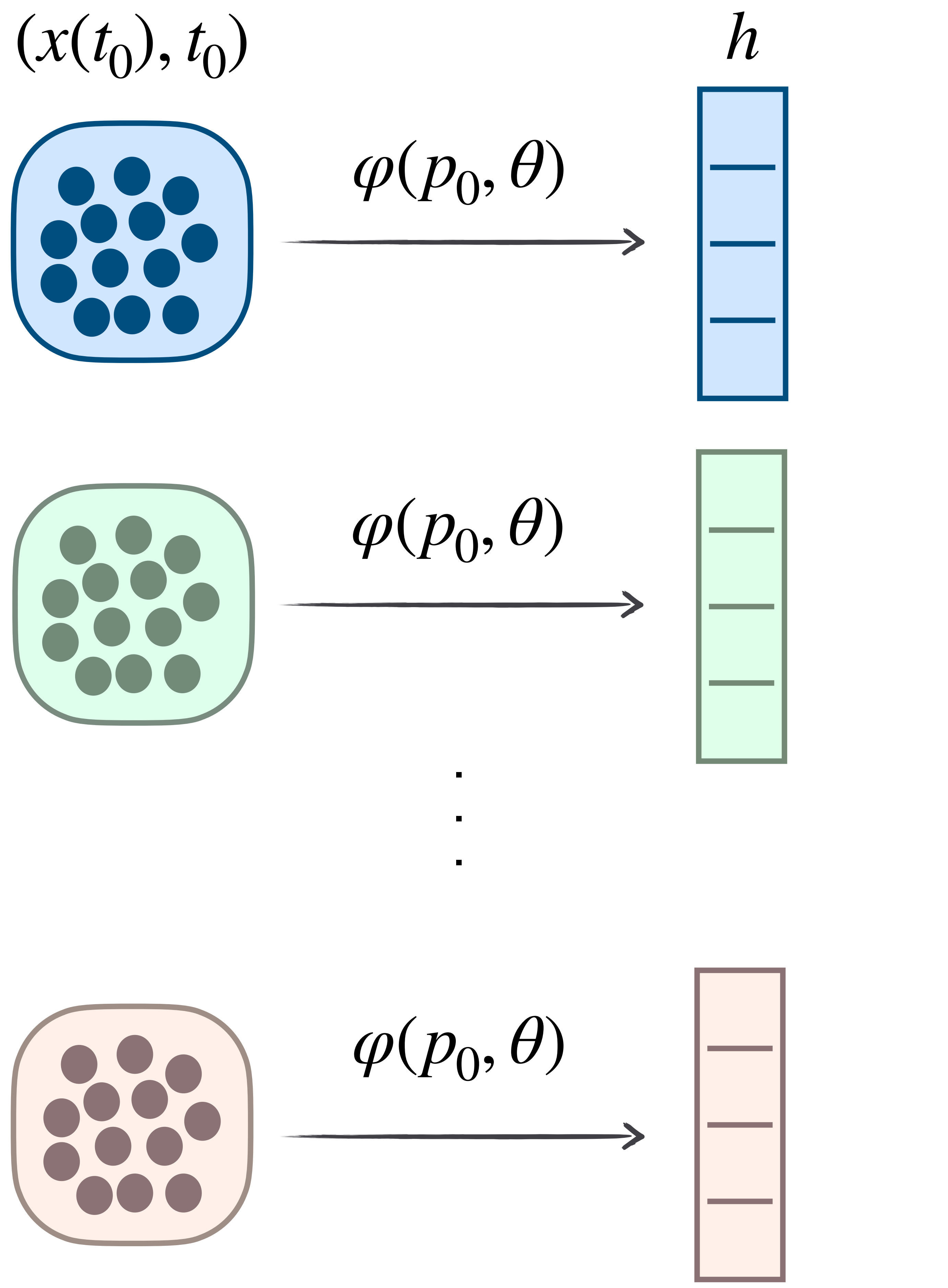
Pseudocode

Dataset $\{(\pi(x_0, x_1 \mid i), c^i)\}_{i=1}^N$, velocity $v_t(\,\cdot\,;\omega)$, and population embedding model $\varphi(\,\cdot\,;\theta)$

1. In every iteration, we sample $i \sim \mathcal{U}_{\{1,N\}}(i), \quad (x_0^j, x_1^j, t^j) \sim \pi(x_0, x_1 \mid i)\mathcal{U}_{[0,1]}(t)$
2. Embed population $h^i(\theta) \leftarrow \varphi\left(\{x_0^j\}_{j=1}^{N_i}; \theta\right)$ (GNN / permutation invariant)
3. Compute loss: $\mathcal{L}_{\text{MFM}}(\omega, \theta) \leftarrow \frac{1}{n}\sum_i \frac{1}{n_i}\sum_j \|\frac{d}{dt} f_t(x_0^j, x_1^j) - v_{t^j}\left(f_t(x_0^j, x_1^j) \,|\, h^i(\theta), c^i; \omega\right)\|^2$
4. We jointly update ω, θ using $\mathcal{L}_{MFM}(\omega, \theta)$ (alternating updates every iteration)

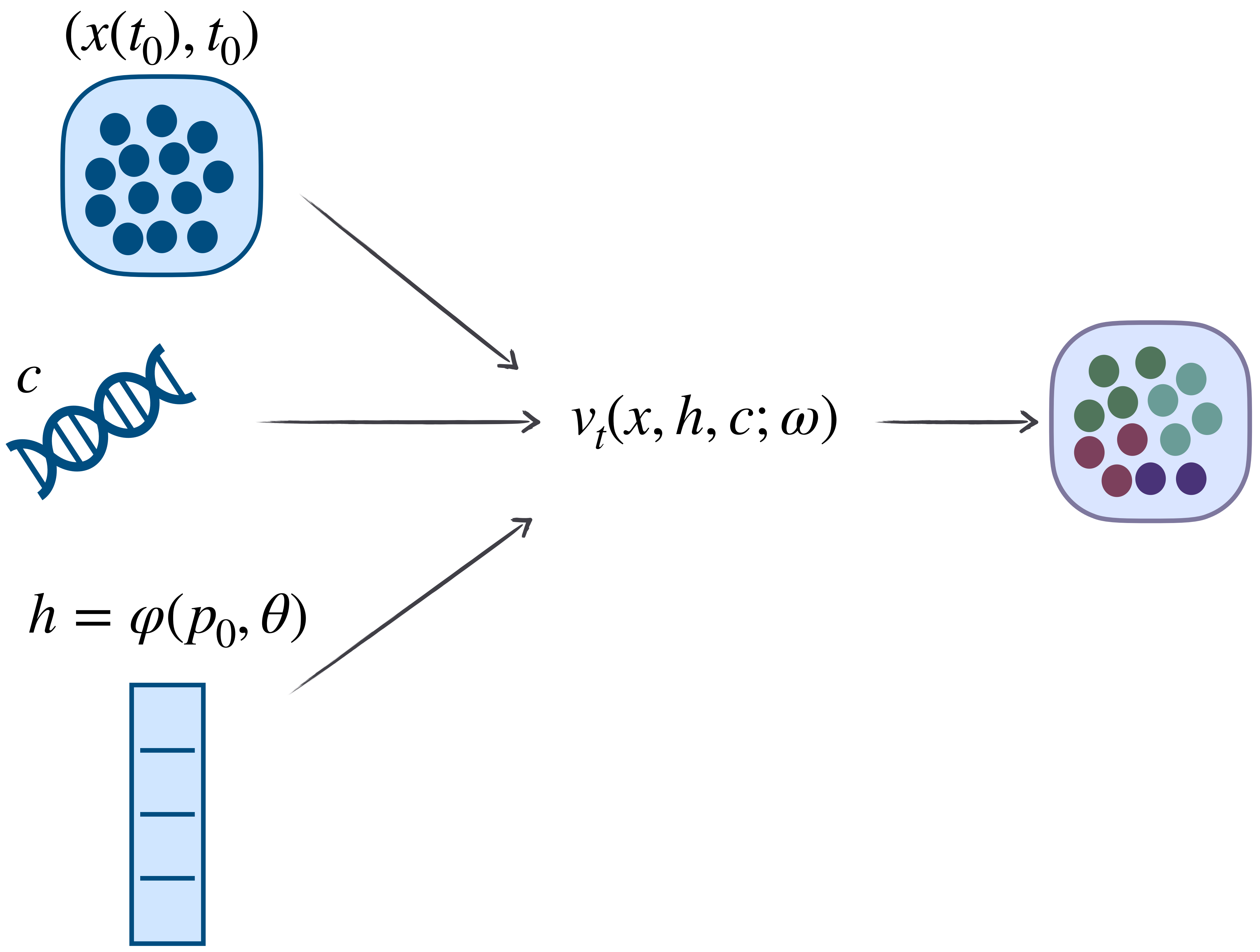
MFM Overview

A ***model*** to learn to represent the population (GCN w/ *knn* edge pooling)



$\varphi(p_0, \theta)$ (GCN) captures interactions between particles

v approximates the population dynamics given representation of the population, and additional *seen* conditions c (e.g. treatments applied to population), as input



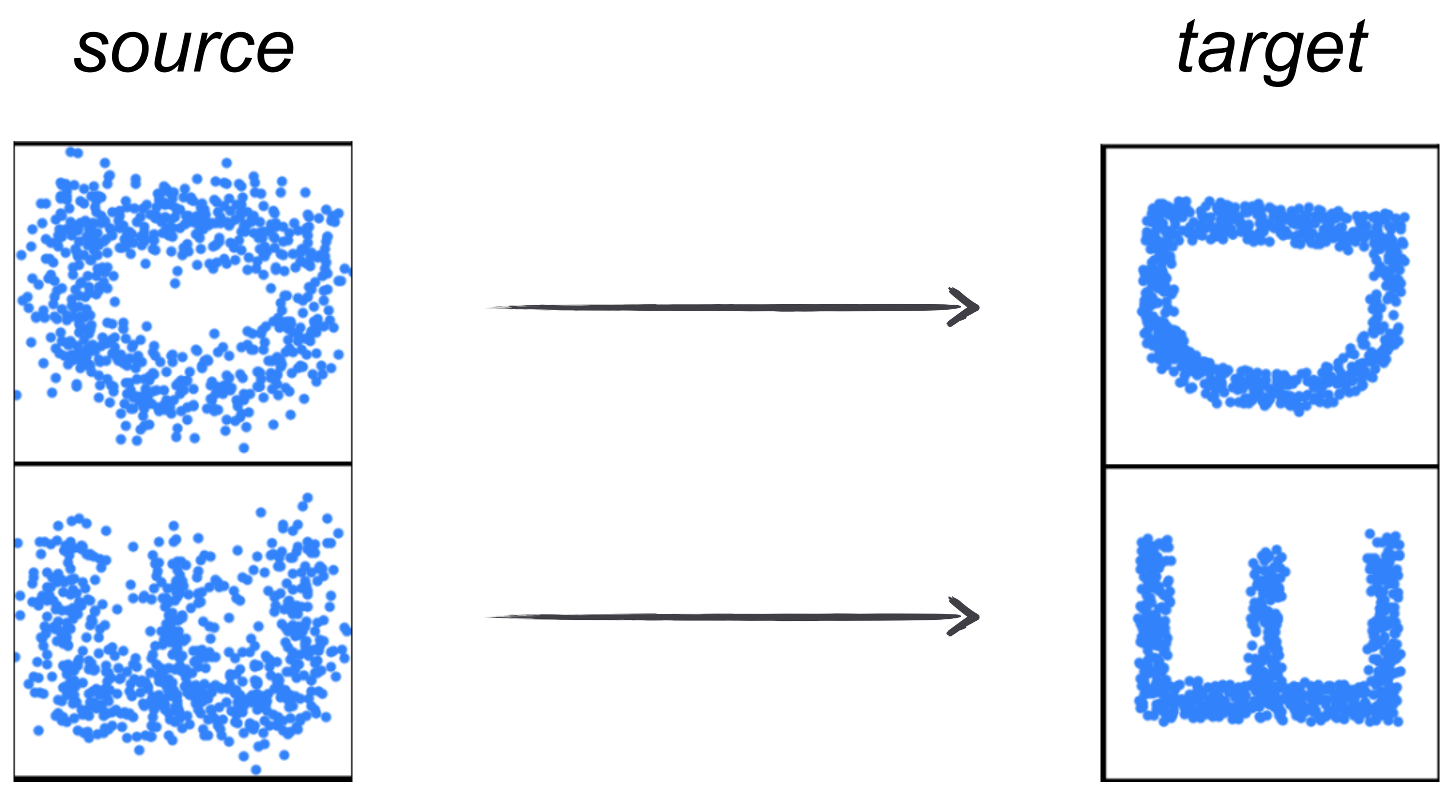
How does MFM perform?

Synthetic Example

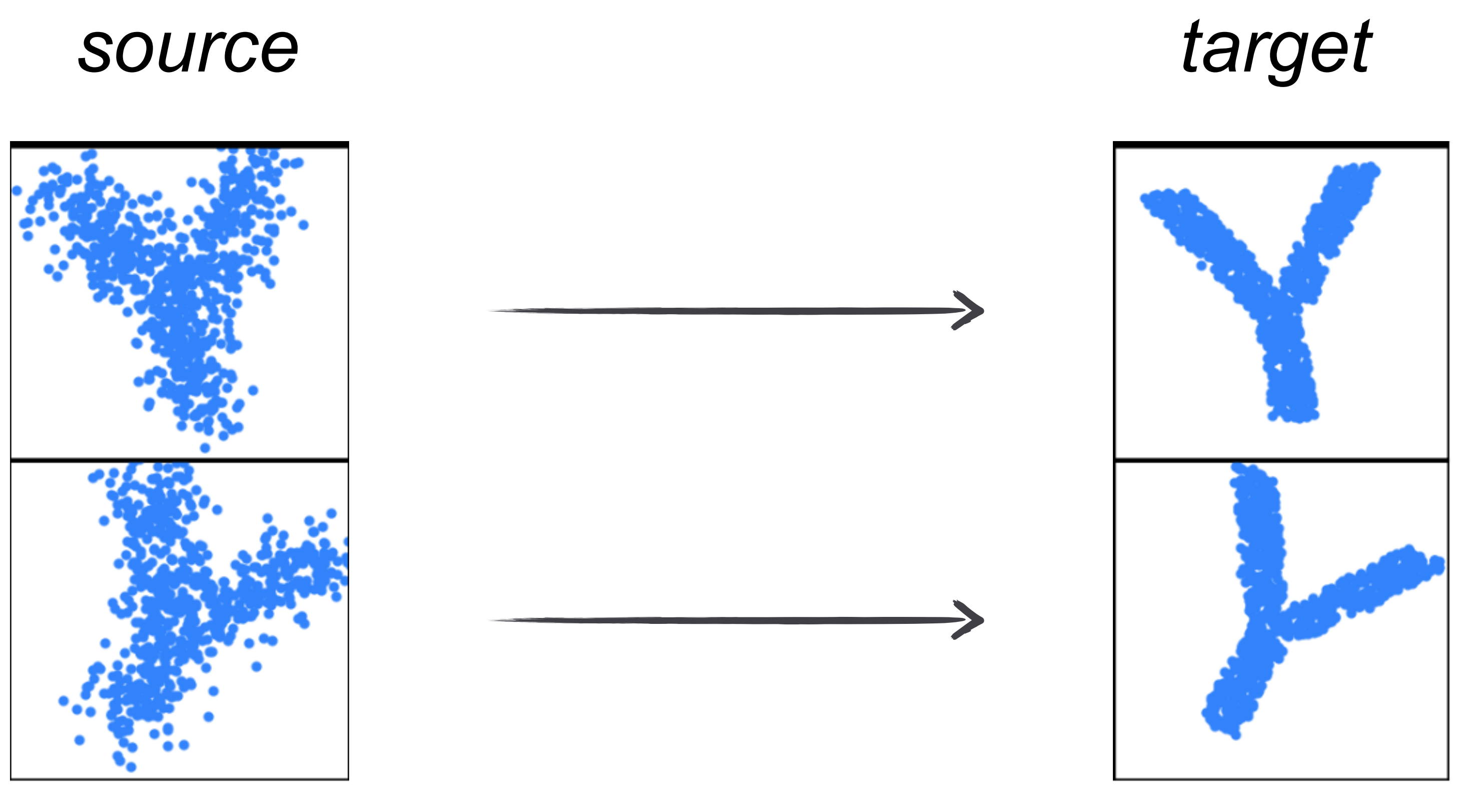
We create a synthetic dataset of paired joint distributions $\{(p_0(x_0 \,|\, i), p_1(x_1 \,|\, i))\}_{i=1}^N$

- We define a set of pre-defined target distributions $p_1(x_1 \,|\, i)$ for $i = 1, \ldots, N$ (letter silhouettes)
- To get paired $p_0(x_0 \,|\, i)$ we simulate the forward diffusion process without drift $x_0 \sim \mathcal{N}(x_1, \sigma)$
- We assume that one can reverse the diffusion process and learn the push-forward map from $p_0(x_0 \,|\, i)$ (***source***) to $p_1(x_1 \,|\, i)$ (***target***) for every index i

Train: 24 letters (excluding ‘Y’ and ‘X’), each in 10 different orientations

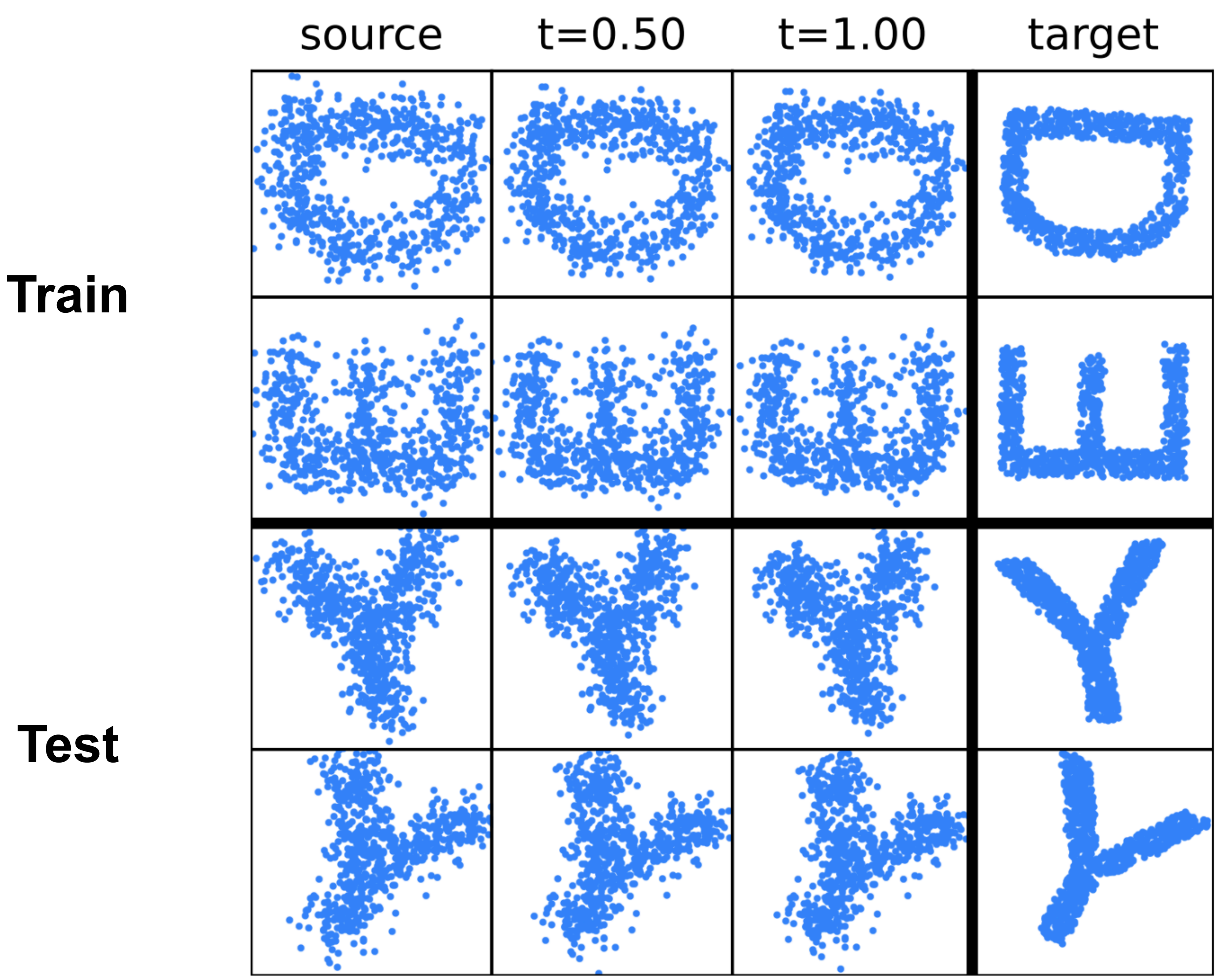


Test: ’Y’ and ‘X’, each in 10 different orientations



Synthetic Example (FM)

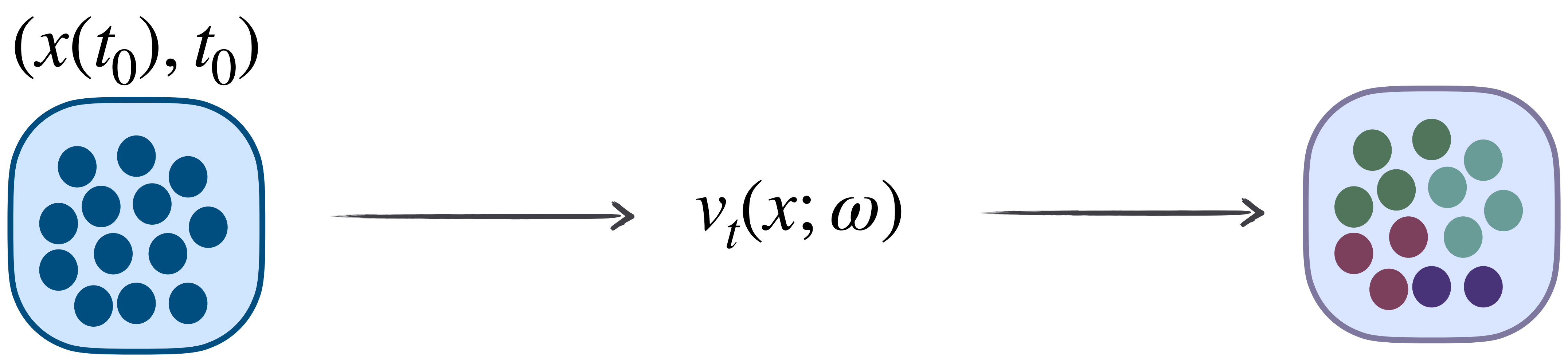
No population information



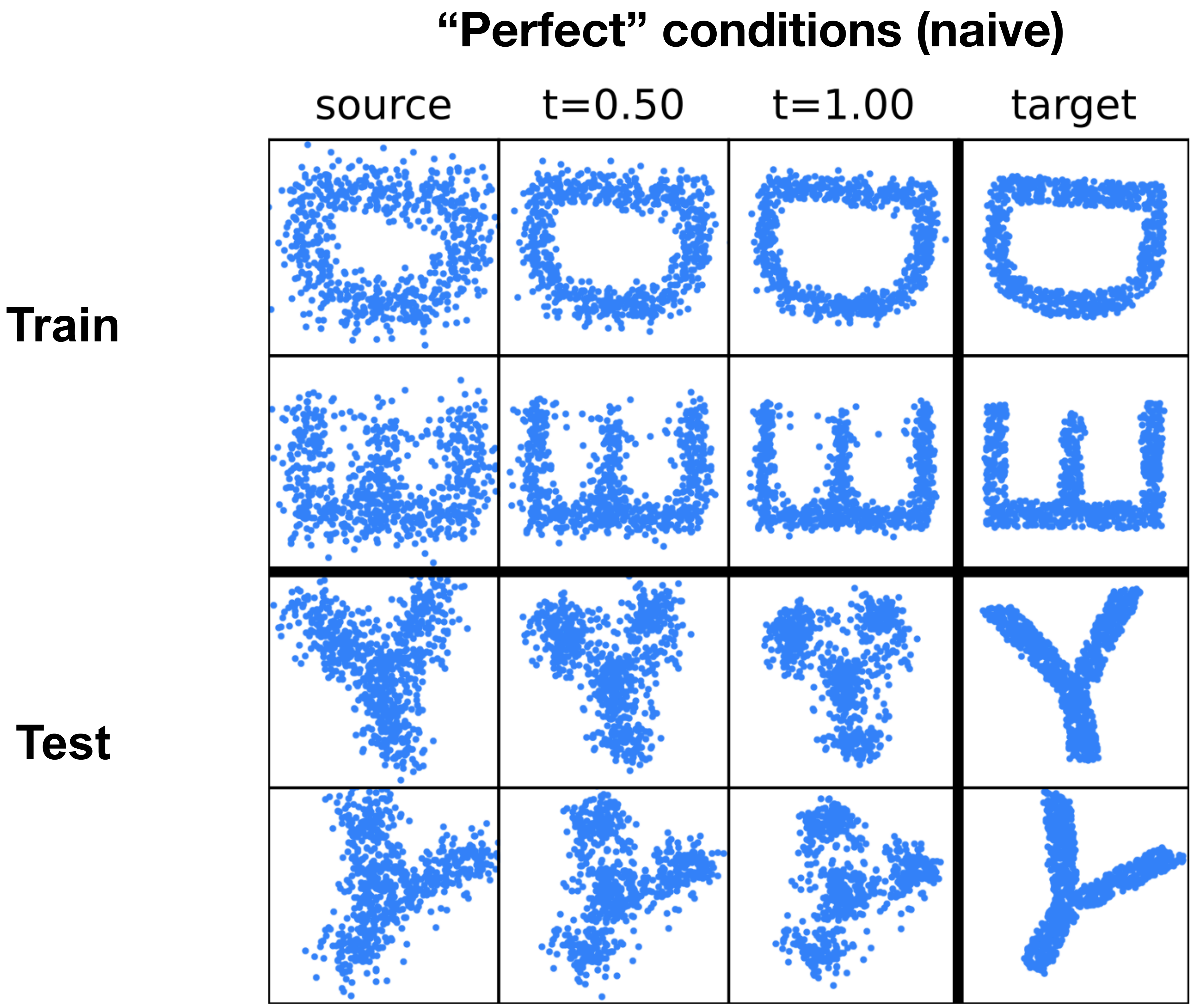
FM

FM cannot fit the training data and cannot generalize to *unseen* populations

Predicts aggregate response over populations



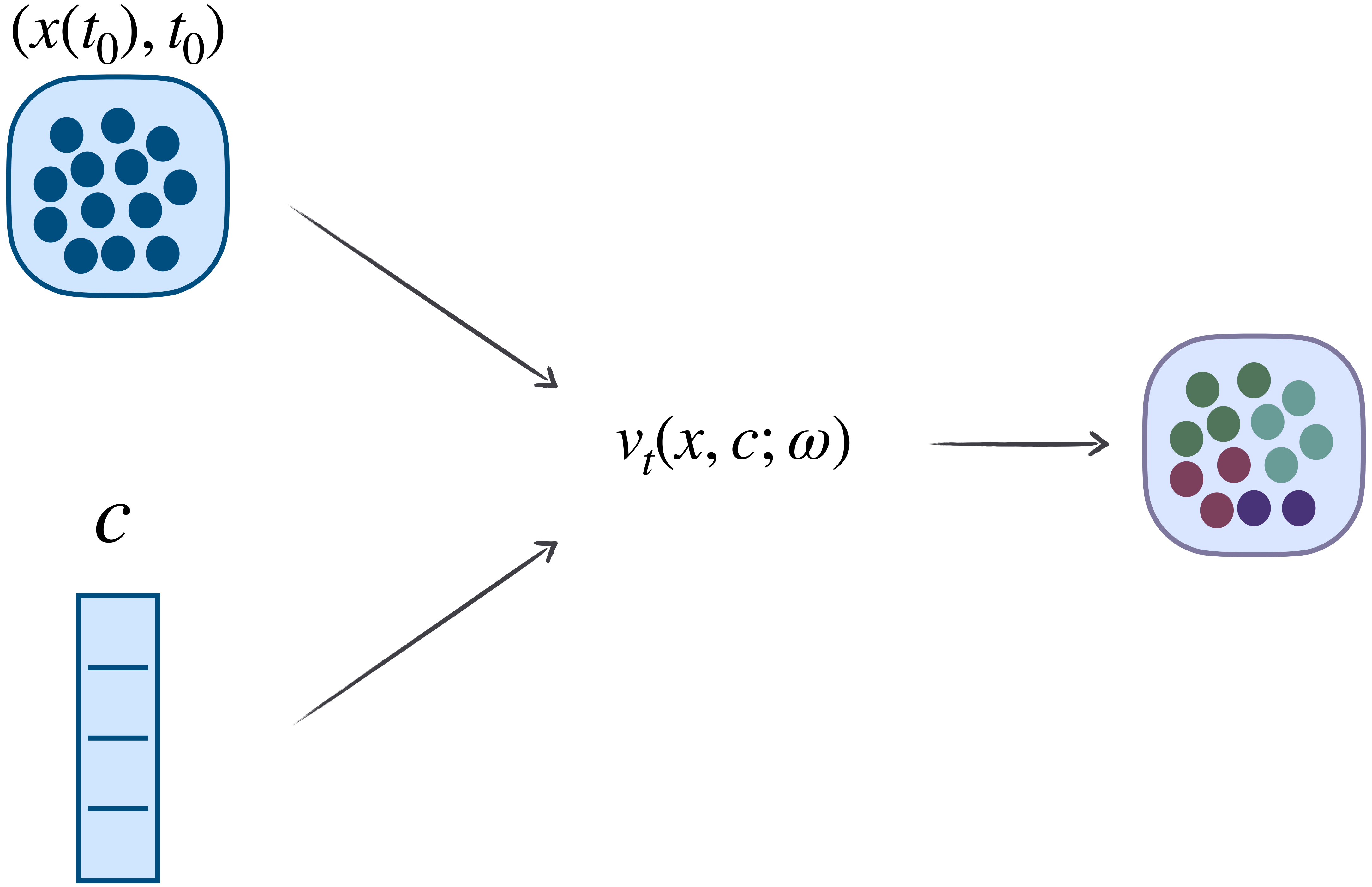
Synthetic Example (CGFM)



CGFM

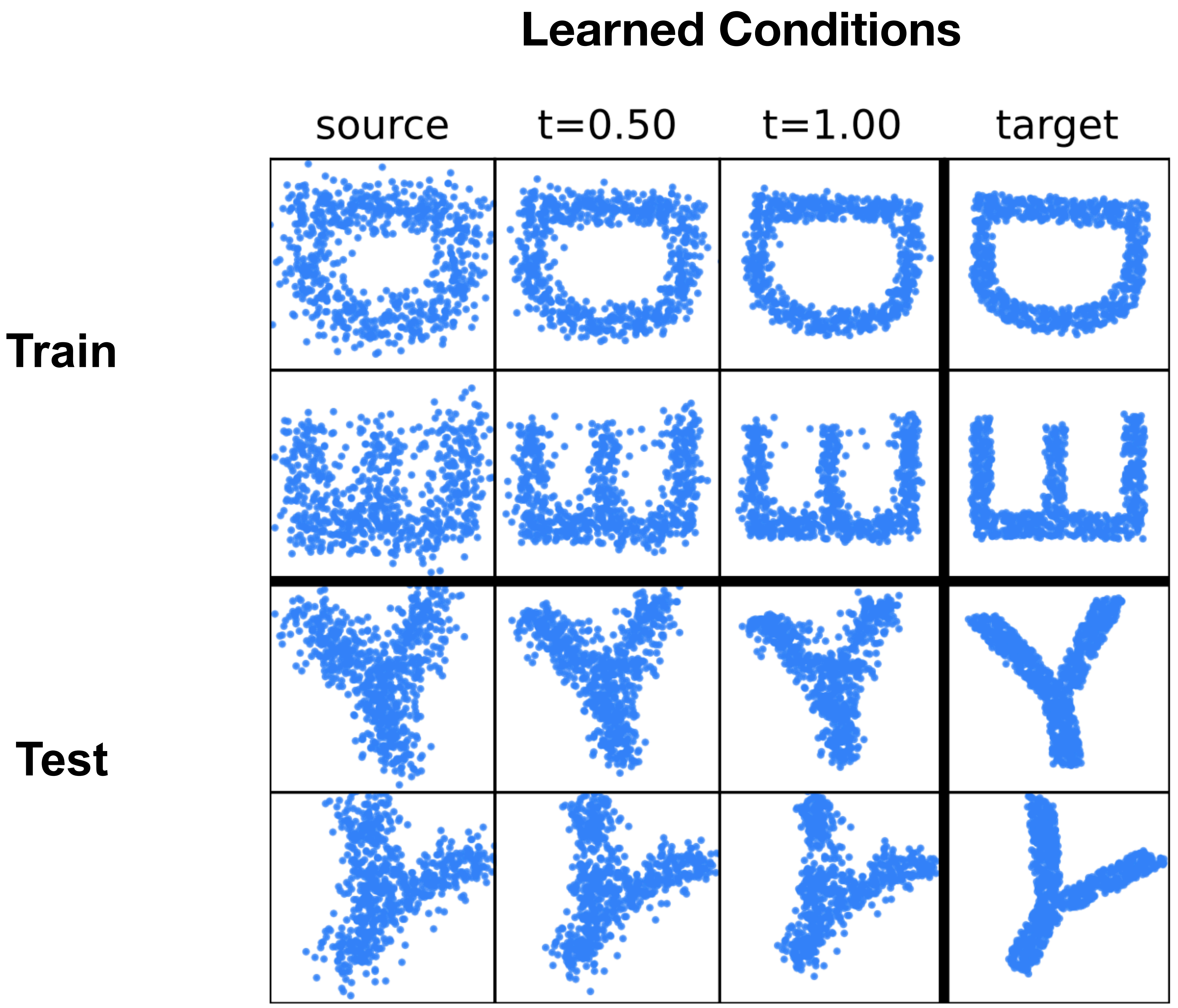
CGFM cannot generalize to the conditions of *unseen* populations

A model v to approximate the dynamical response given the population index/condition c



Perfect information on which environment the model is working with

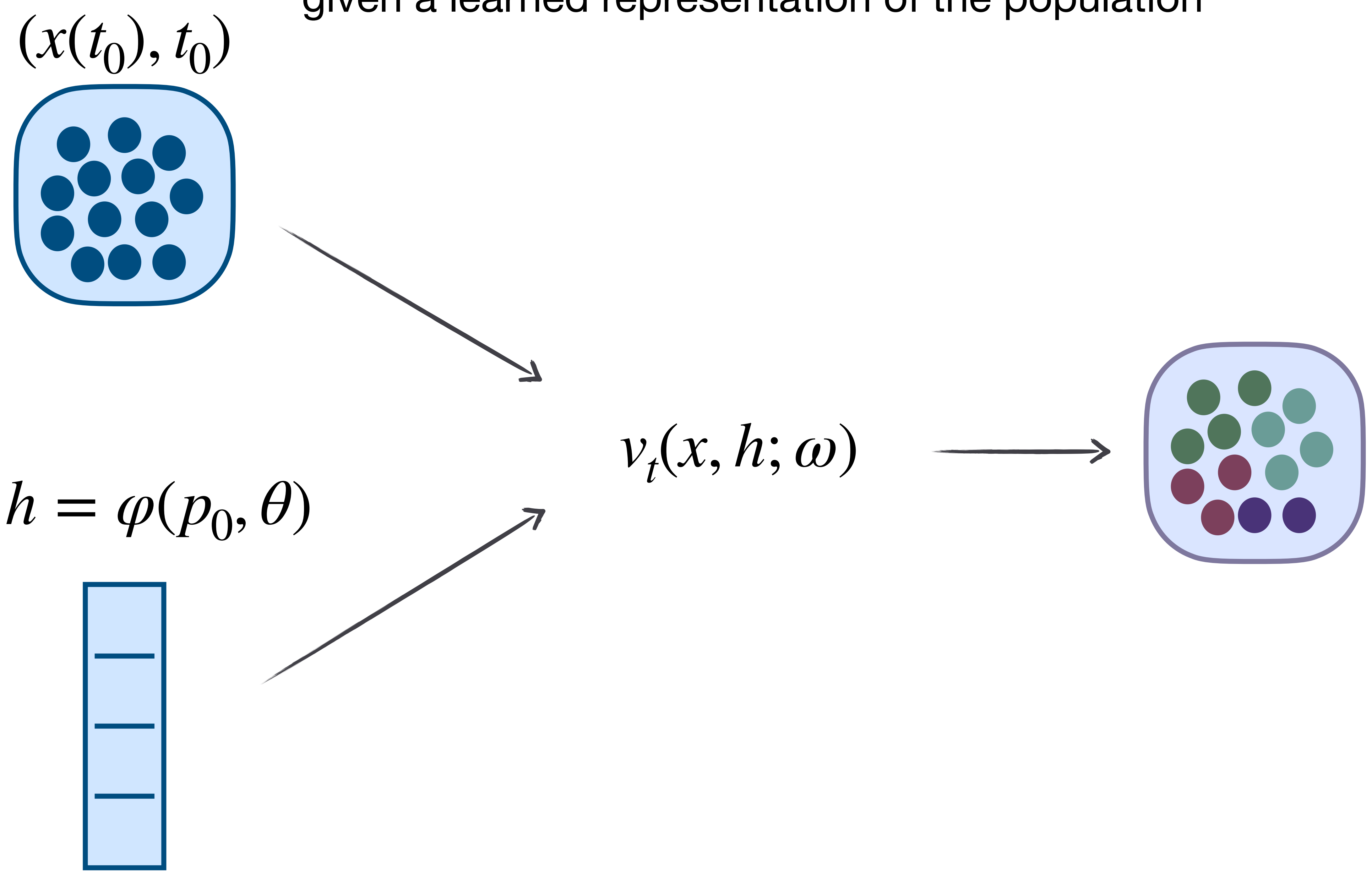
Synthetic Example (MFM)



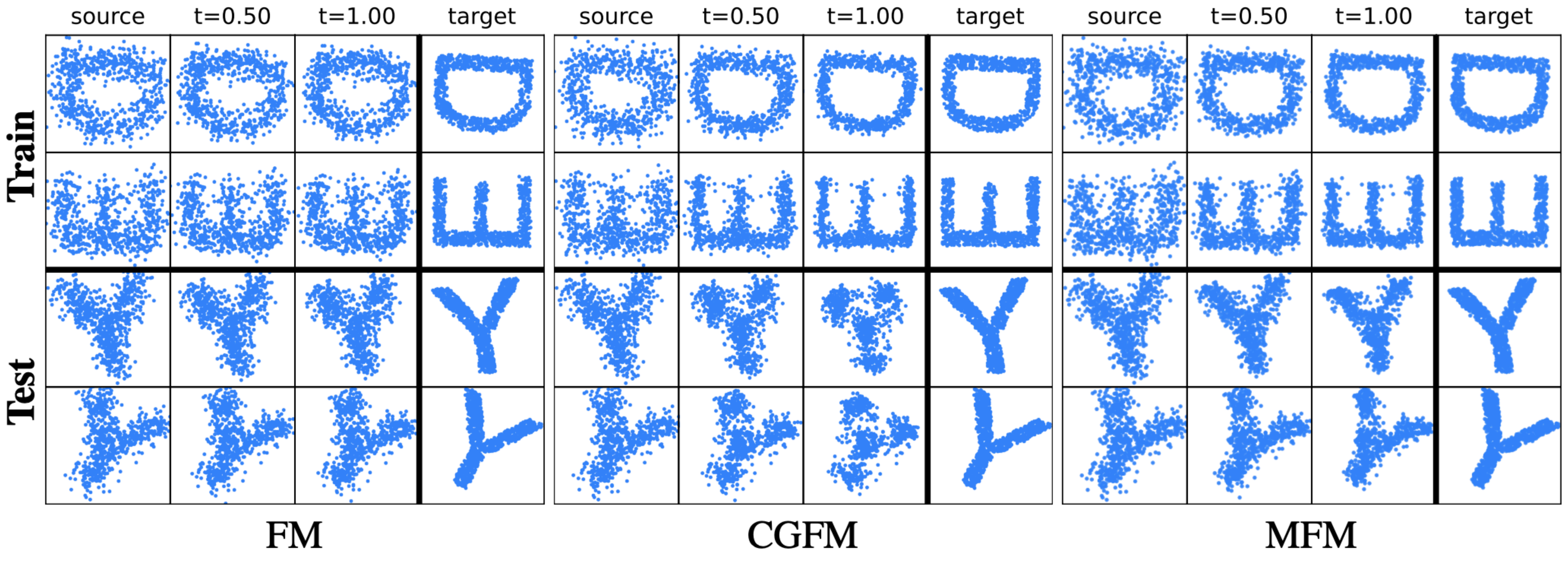
MFM

MFM learns to represent entire populations, hence generalizes across *unseen* populations

A model v to approximate the dynamical response given a learned representation of the population



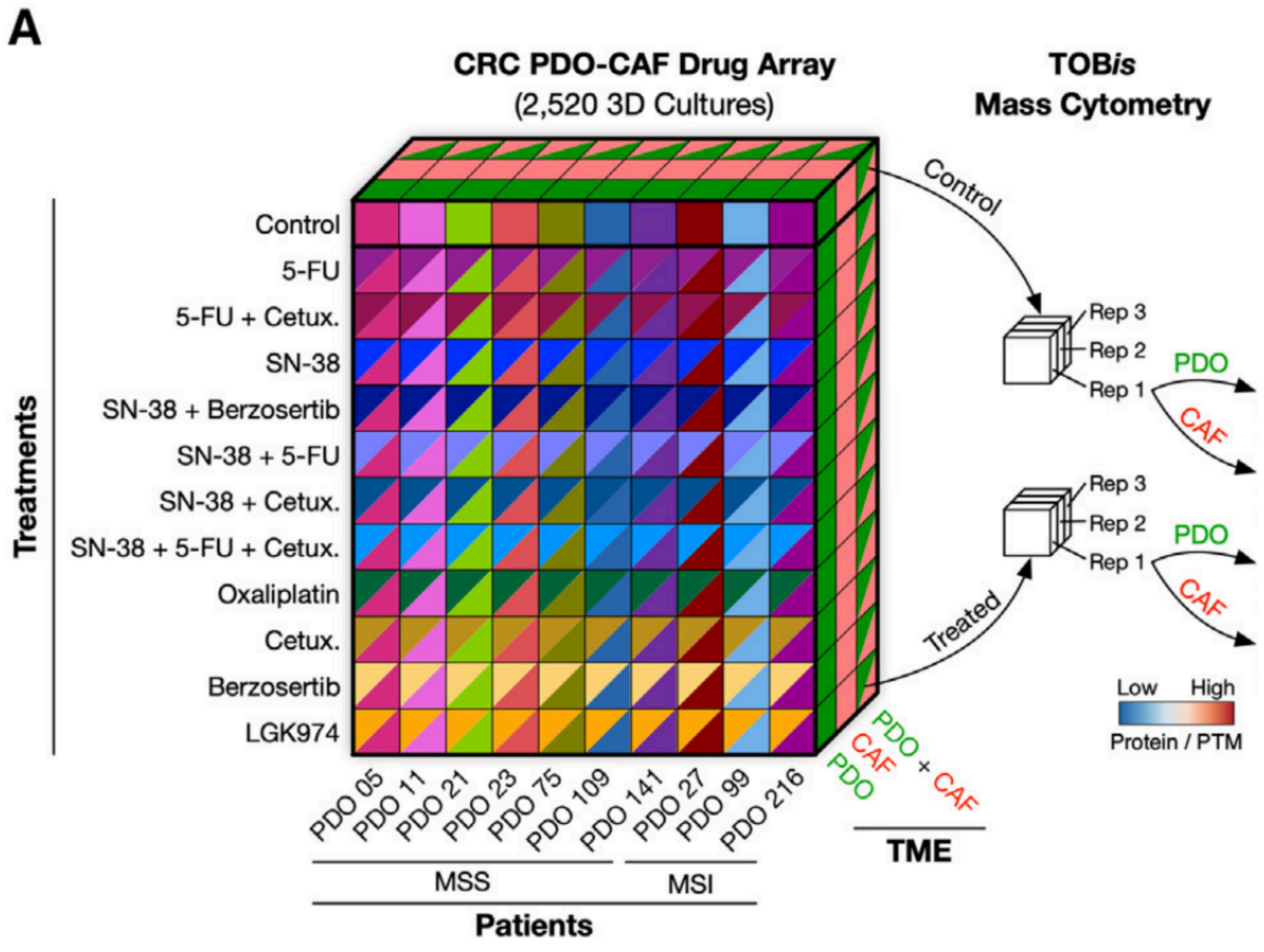
Synthetic Example



	Train			Test (X's)			Test (Y's)		
	$\mathcal{W}_1$	$\mathcal{W}_2$	MMD ($\times 10^{-3}$)	$\mathcal{W}_1$	$\mathcal{W}_2$	MMD ($\times 10^{-3}$)	$\mathcal{W}_1$	$\mathcal{W}_2$	MMD ($\times 10^{-3}$)
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	$\mathbf{2.18 \pm 0.25}$	$\mathbf{0.204 \pm 0.005}$	$\mathbf{0.249 \pm 0.006}$	$\mathbf{3.14 \pm 0.18}$

Biological data — patient-specific organoid drug screen dataset

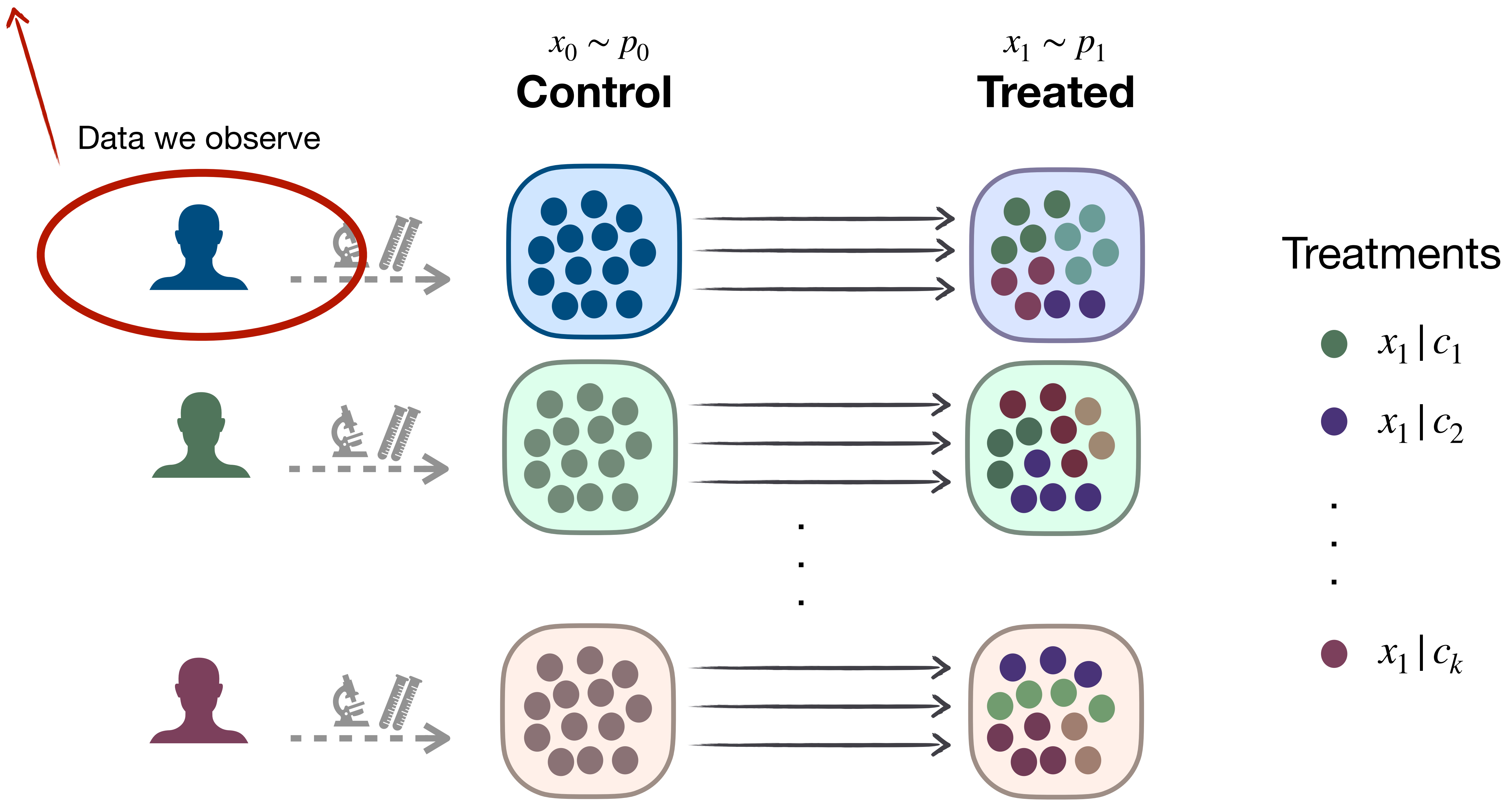
Mass Cytometry



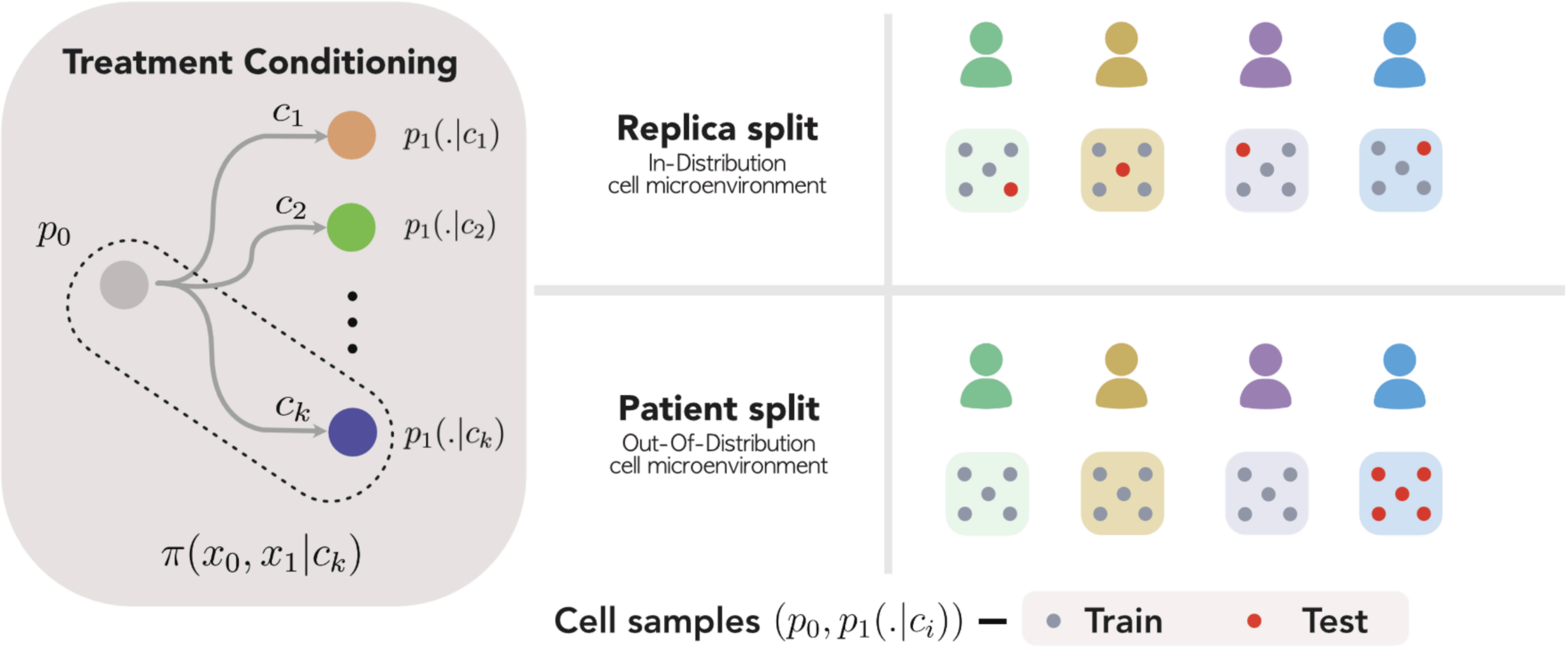
(Zapatero et al, *Cell*, 2023)

10 patients, 11 treatments, varying doses, 3 different cell cultures … ***up to 2500 different environmental conditions!*** (we use ~ 1000)

Each patient has ~ 250 different (control, treated) pairs



Organoid Drug Screen Data



“Replica” Split

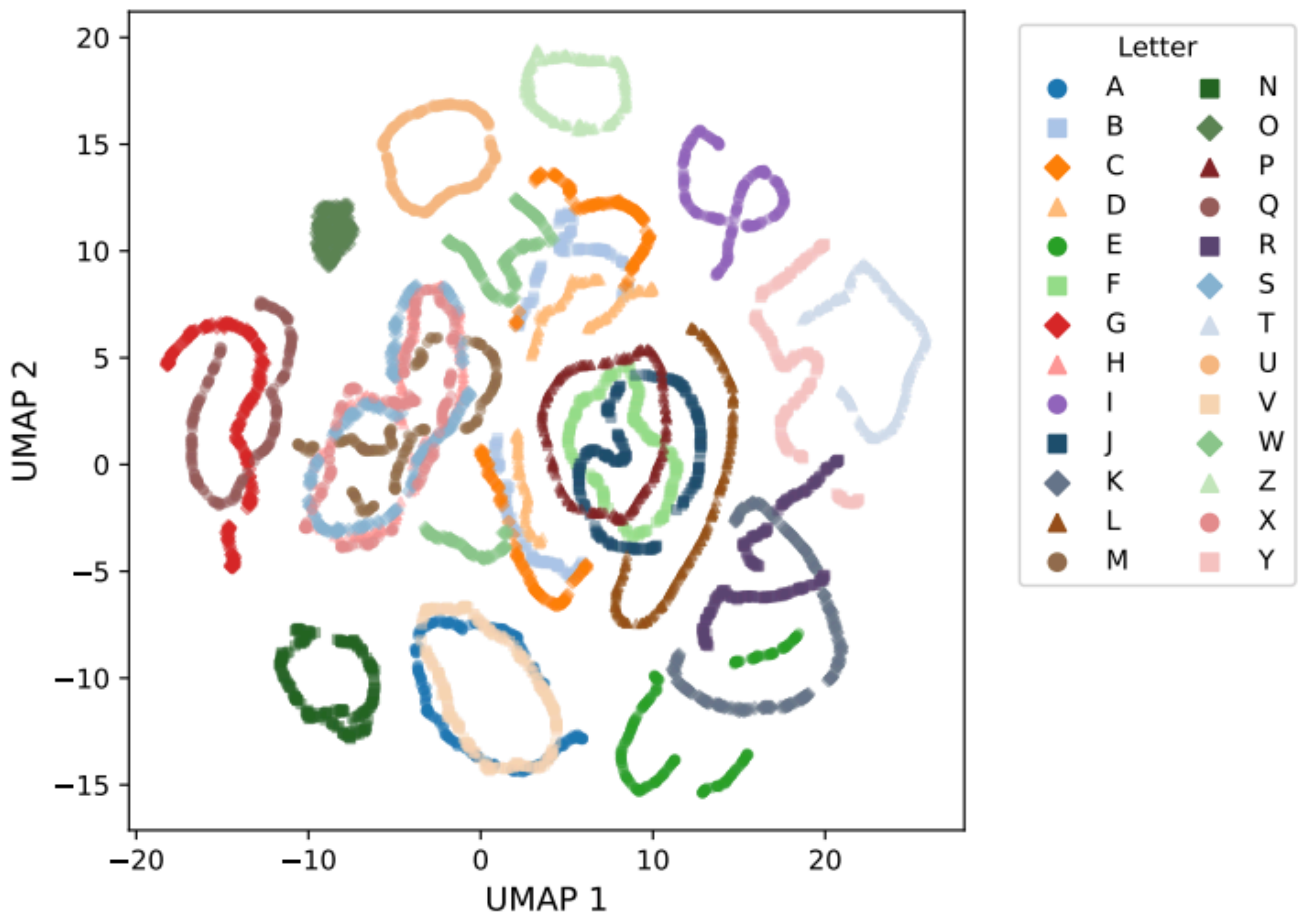
		$\mathcal{W}_1(\downarrow)$	$\mathcal{W}_2(\downarrow)$	MMD $(\times 10^{-3})$ $(\downarrow)$	$r^2(\uparrow)$
$d(p_0, p_1)$		4.513	4.695	19.14	0.876
$d(p_0 - \mu_0 + \tilde{\mu}_1, p_1)$		6.222	6.346	74.90	0.876
FM		4.340 ± 0.078	4.564 ± 0.111	13.00 ± 0.67	0.865 ± 0.034
CGFM		4.443 ± 0.033	4.621 ± 0.041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

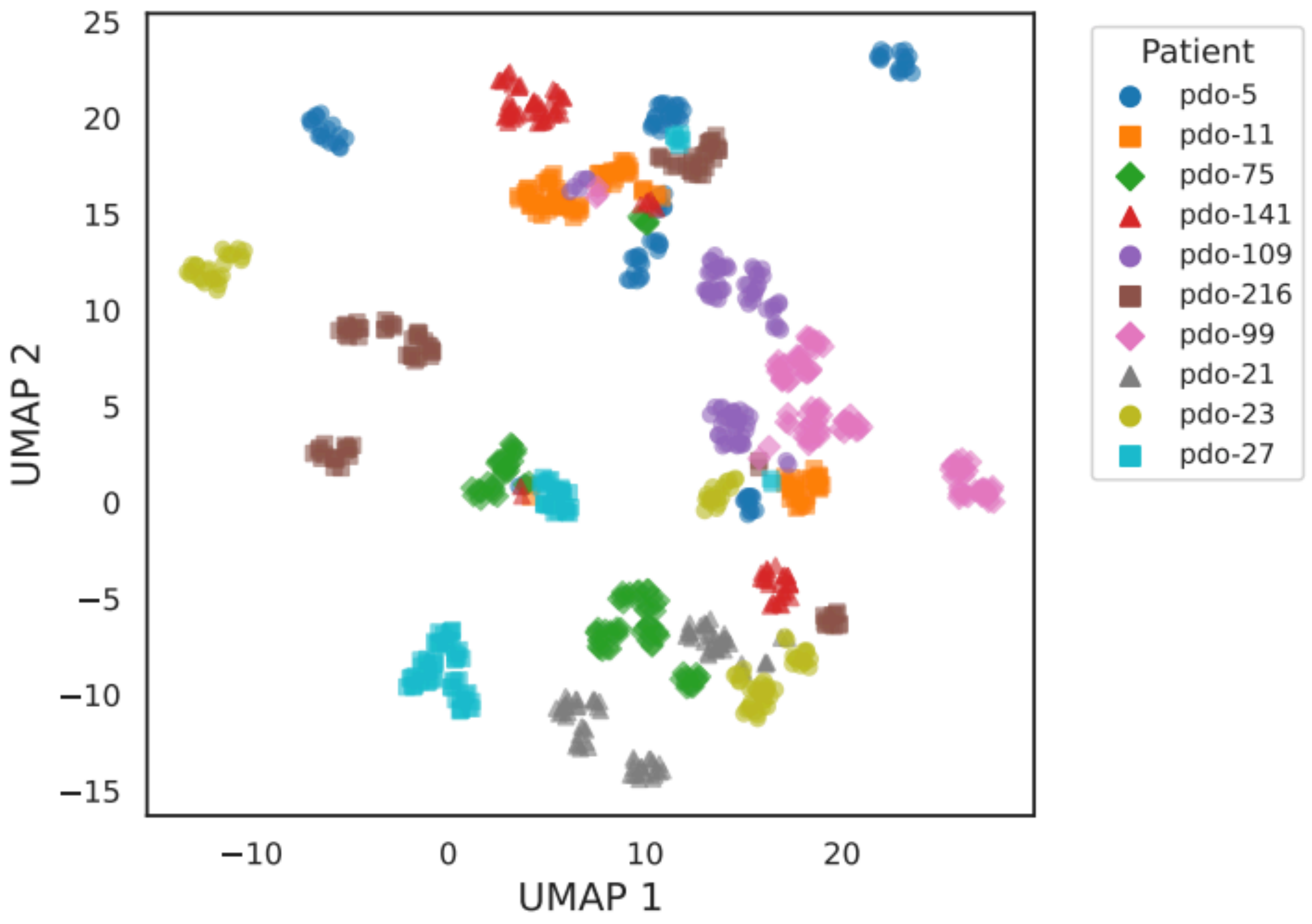
VECTOR INSTITUTE

Patient Split(s)

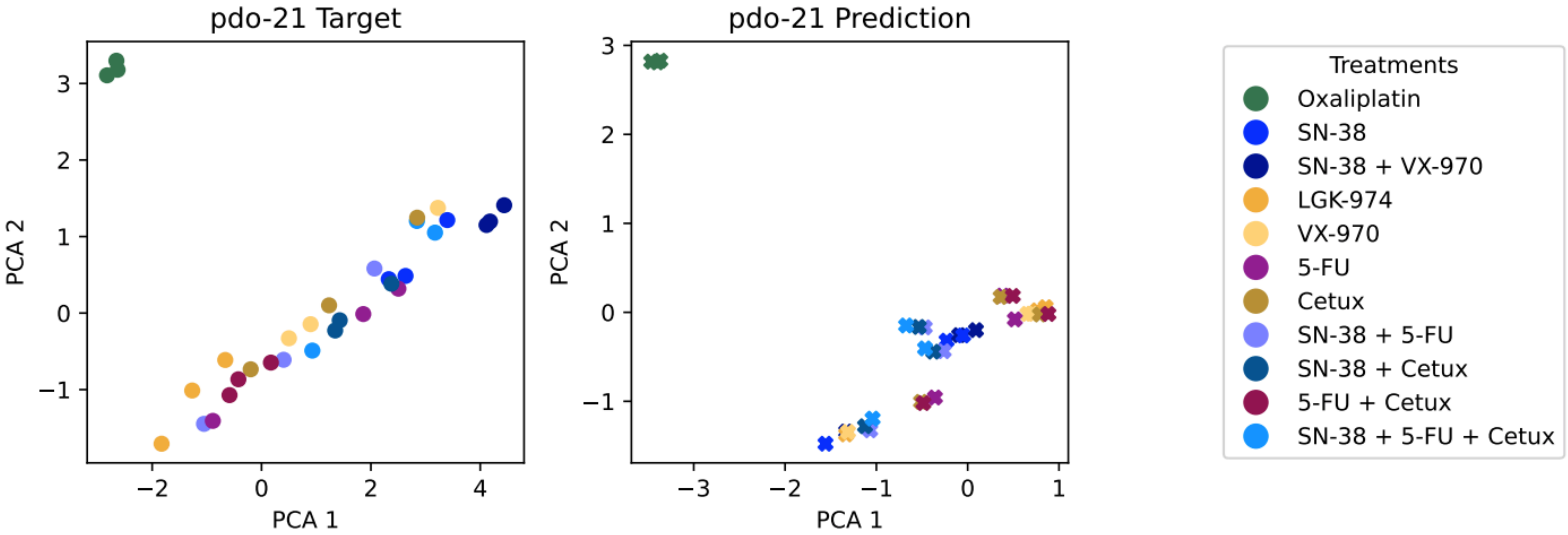
	$\mathcal{W}_1(\downarrow)$	$\mathcal{W}_2(\downarrow)$	MMD $(\times 10^{-3})$ $(\downarrow)$	$r^2(\uparrow)$
$d(p_0, p_1)$	4.175 ± 0.135	4.303 ± 0.174	12.11 ± 2.07	0.902 ± 0.006
$d(p_0 - \mu_0 + \tilde{\mu}_1, p_1)$	6.158 ± 0.239	6.235 ± 0.229	77.74 ± 10.72	0.902 ± 0.006
FM	4.171 ± 0.107	4.315 ± 0.142	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
$\text{MFM}_{k=0}$ (ours)	4.135 ± 0.094	4.268 ± 0.128	10.18 ± 1.28	0.918 ± 0.007
$\text{MFM}_{k=10}$ (ours)	4.112 ± 0.086	4.243 ± 0.121	9.90 ± 0.99	0.925 ± 0.008
$\text{MFM}_{k=50}$ (ours)	$\mathbf{4.087 \pm 0.122}$	$\mathbf{4.218 \pm 0.160}$	$\mathbf{9.26 \pm 1.56}$	0.926 ± 0.007
$\text{MFM}_{k=100}$ (ours)	4.112 ± 0.148	4.244 ± 0.186	9.63 ± 2.08	$\mathbf{0.931 \pm 0.002}$
$\text{FM}^{w/}\text{OT}$	$\underline{4.064} \pm \underline{0.152}$	$\underline{4.189} \pm \underline{0.194}$	$\underline{9.44} \pm \underline{2.49}$	$\underline{0.932} \pm \underline{0.005}$
$\text{CGFM}^{w/}\text{OT}$	4.087 ± 0.129	4.217 ± 0.165	9.83 ± 2.03	0.924 ± 0.009
ICNN	4.157 ± 0.168	4.282 ± 0.213	11.18 ± 2.51	0.904 ± 0.005

MFM Learns Meaningful Embeddings





MFM Predict Patient Specific Treatment Response

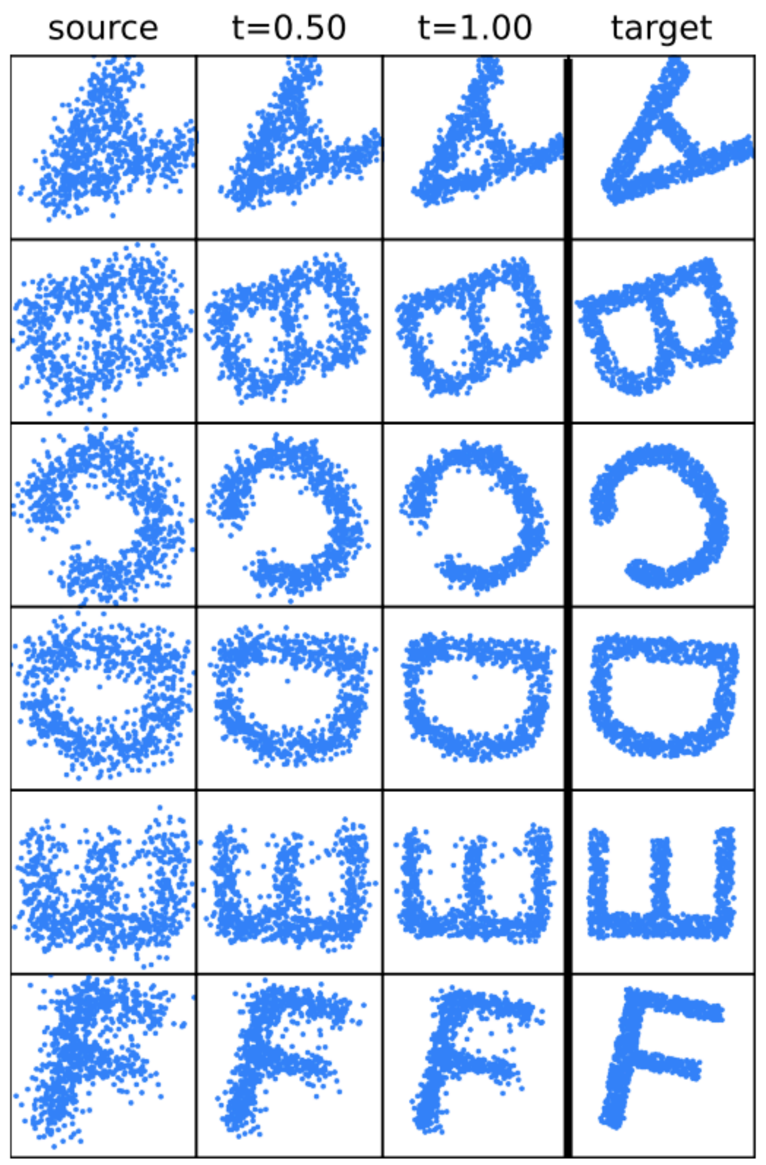


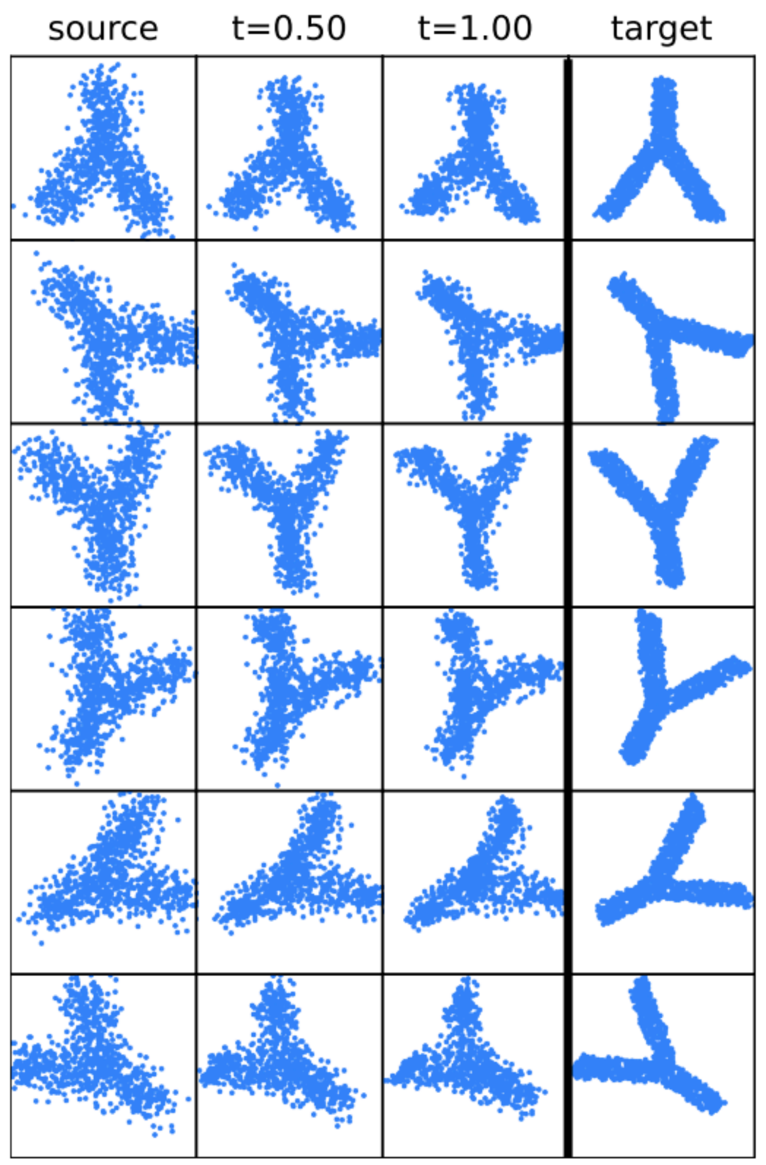
Conclusions

(1) We highlighted the importance of modelling dynamics based on the entire distribution and introduce a practical approach (MFM) to address this.

(2) We showed that we can use MFM to learn meaningful embeddings to generalize over various initial distributions.

(3) We showed that MFM can learn meaningful embeddings of single-cell populations along the developmental model of these populations.





Thanks for your Attention!

Acknowledgments

- Xi (Nicole) Zhang
- Brandon Amos
- Mathieu Blanchette
- Leo J. Lee
- Yoshua Bengio
- Alexander Tong
- Kirill Neklyudov

UNIVERSITY OF TORONTO

VECTOR INSTITUTE

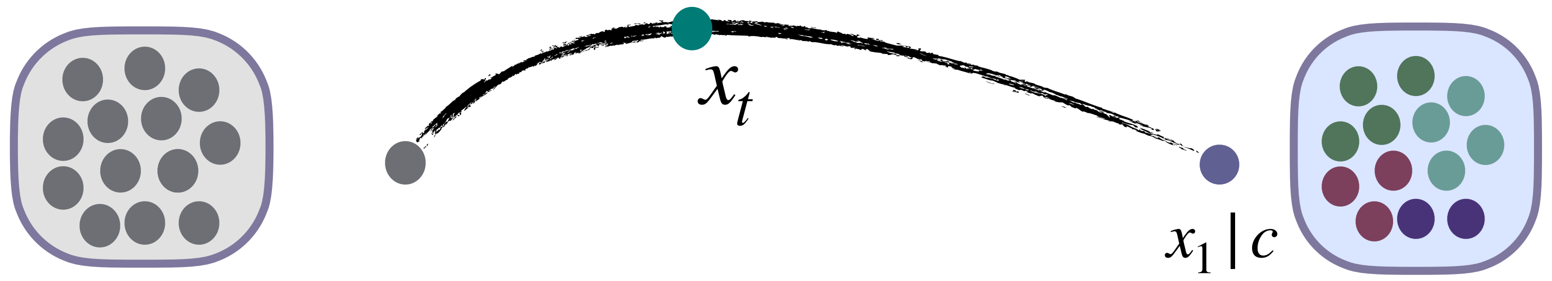
Mila

Meta

Naive approach: Conditional Generative Flow Matching (CGFM)

Similar to FM, we can assume there exists a **continuous interpolation** between densities $p_0(x_0)$ and $p_1(x_1)$ — now with the addition of a condition c.

$$x_t = f_t(x_0, x_1), \ \ (x_0, x_1) \sim \pi(x_0, x_1 \mid c)$$



Again, from the continuity equation, we can describe the changes of density through a vector field $v_t^*(x, c)$. We can learn a $v_t(x, c; \omega)$

$$\frac{\partial p_t(x)}{\partial t} = - \langle \nabla_x, p_t(x \mid c) v_t^*(x, c) \rangle \ ,$$

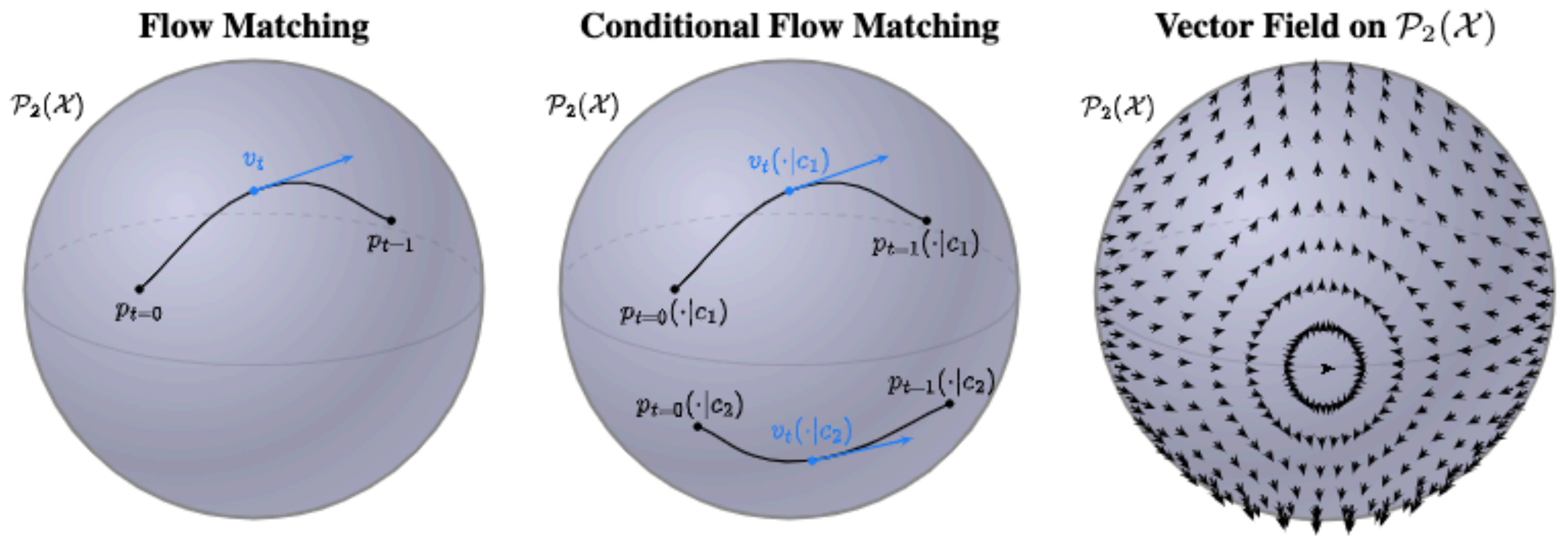
$$v_t^*(\xi \mid c) = \frac{1}{p_t(\xi \mid c)} \mathbb{E}_{\pi(x_0, x_1 \mid c)}[\delta(f_t(x_0, x_1) - \xi) \frac{\partial f_t(x_0, x_1)}{\partial t}]$$

We can approximate $v_t^*(x)$ via a parameterized function $v_t(x; \omega)$. And likewise, derive a tractable objective/loss for learning:

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

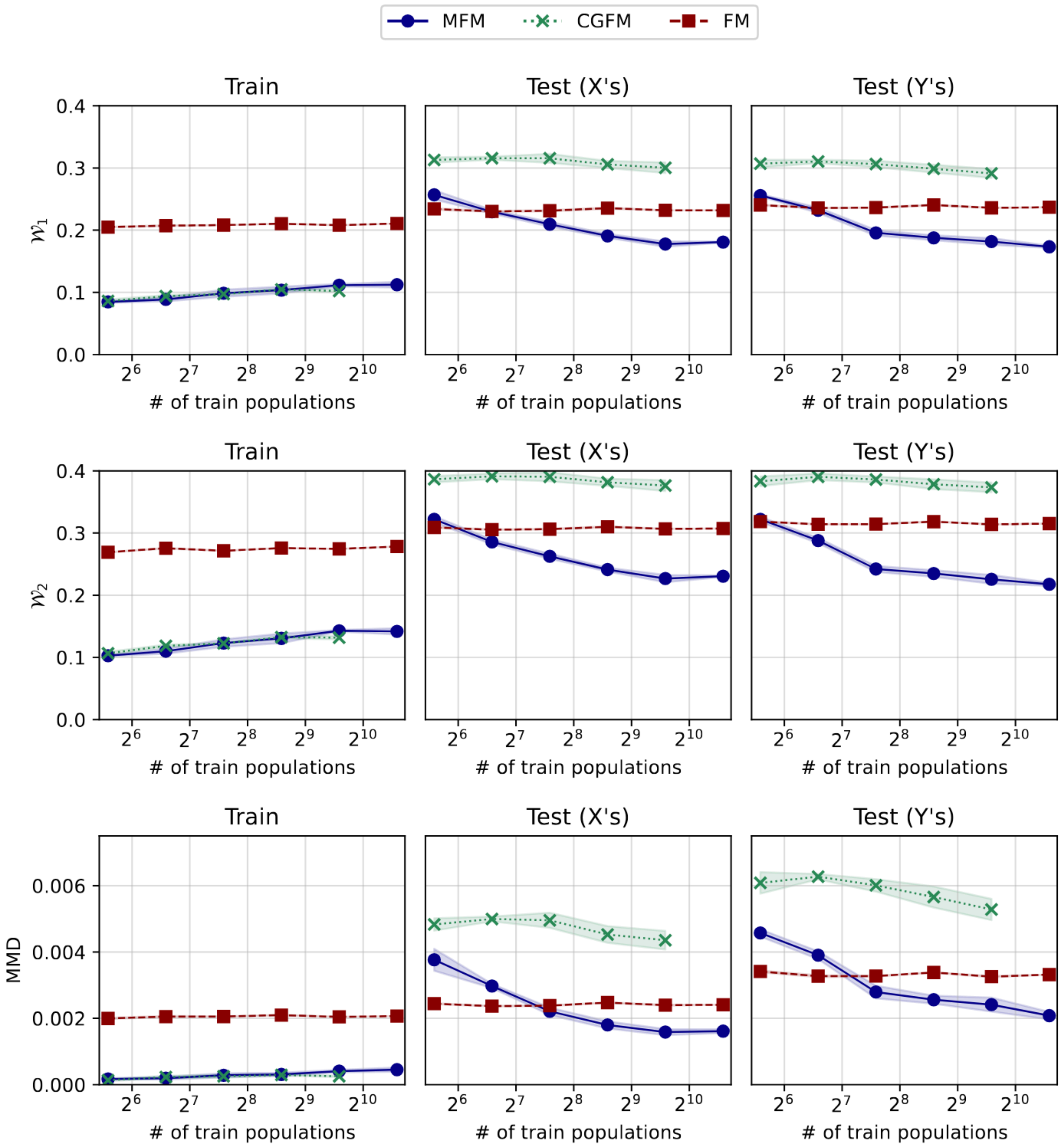
Integrating Vector Fields on the Wasserstein Manifold

Wasserstein Manifold



Manifold/space of distributions (*meta*). Think, each training point is an entire distribution!

Data Ablation



Algorithm (training)

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` $h \leftarrow \emptyset$`.`

$\mathcal{L}_{\text{MFM}}(\omega, \theta) \leftarrow \frac{1}{n}\sum_i \frac{1}{n_i}\sum_j \left\| \frac{d}{dt} f_t(x_0^j, x_1^j) - v_{t^j}\Big(f_t(x_0^j, x_1^j) \mid h^i(\theta), c^i; \omega\Big) \right\|^2$

$\omega' \leftarrow \text{Update}(\omega, \nabla_\omega \mathcal{L}_{\text{MFM}}(\omega, \theta))$ `// 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 $v_t(\cdot;\omega^*), \varphi(\cdot;\theta^*)$

Algorithm (sampling)

Algorithm 2: Meta Flow Matching (sampling)

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

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

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

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