On transport, flows, and physics

Brandon Amos
bamos.github.io/presentations

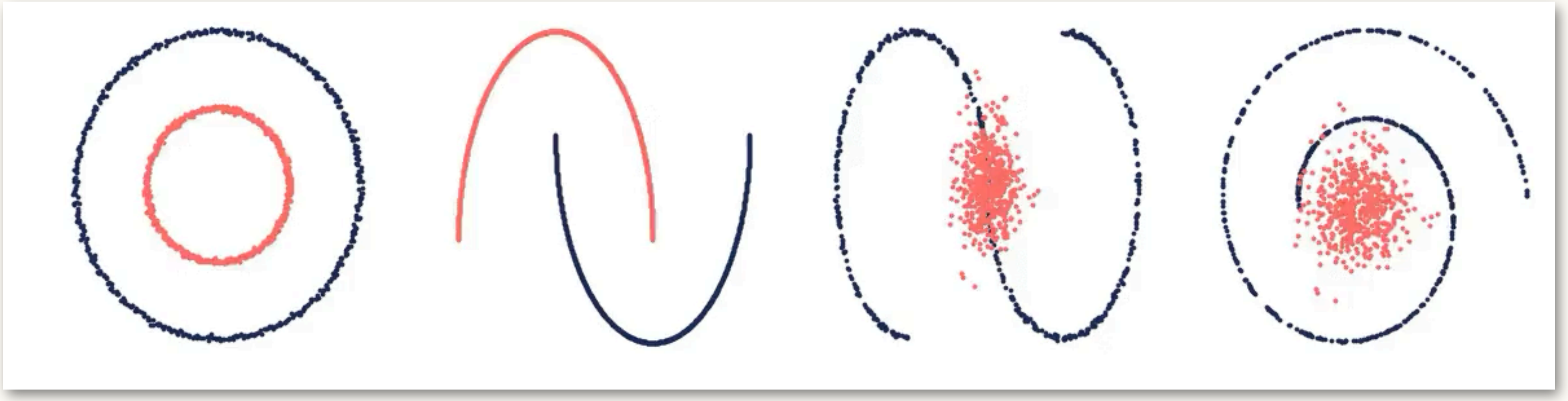
In collaboration with Samuel Cohen, Arnaud Fickinger, Stuart Russell, Aram-Alexandre Pooladian, Doron Haviv, Dana Pe’er, Carles Domingo-Enrich, Ricky Chen, Lazar Atanackovic, Xi Zhang, Mathieu Blanchette, Leo J. Lee, Yoshua Bengio, Alexander Tong, Kirill Neklyudov

To start: some preliminaries

(Optimal) transport and flows

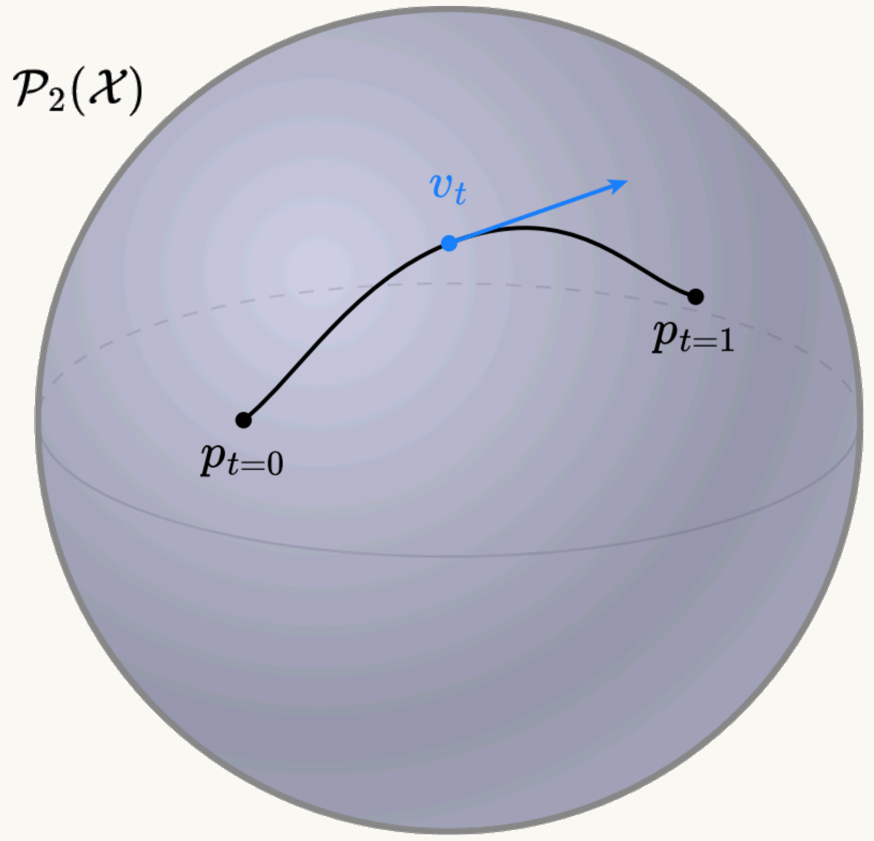
e.g., EMD, Wasserstein distance, diffusion models

Find map T such that $T_{\#}p_0 \approx p_1$



📚 *On amortizing convex conjugates.* Amos, ICLR 2023.

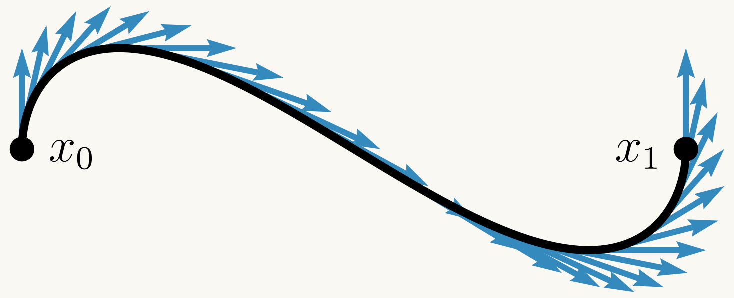
Wasserstein manifold

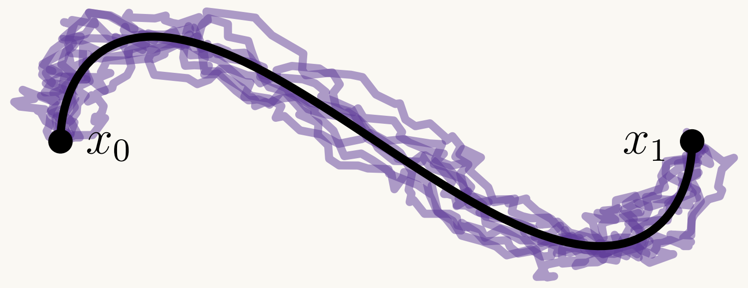


“Physics” in this talk

⚠️ & for me as a non-physicist

differential equations $\dot{x}_t = f(x_t)$ 🙃





e.g., the **Newton-Euler** equations of motion

$M(q_t)\ddot{q}_t + n(q_t, \dot{q}_t) = \tau(q_t) + Bu_t$

Mechanics is the paradise of the mathematical sciences because by means of it one comes to the fruits of mathematics.
—da Vinci (found in Bullo & Lewis)

This talk

Physics-tangential **extensions** and **applications**

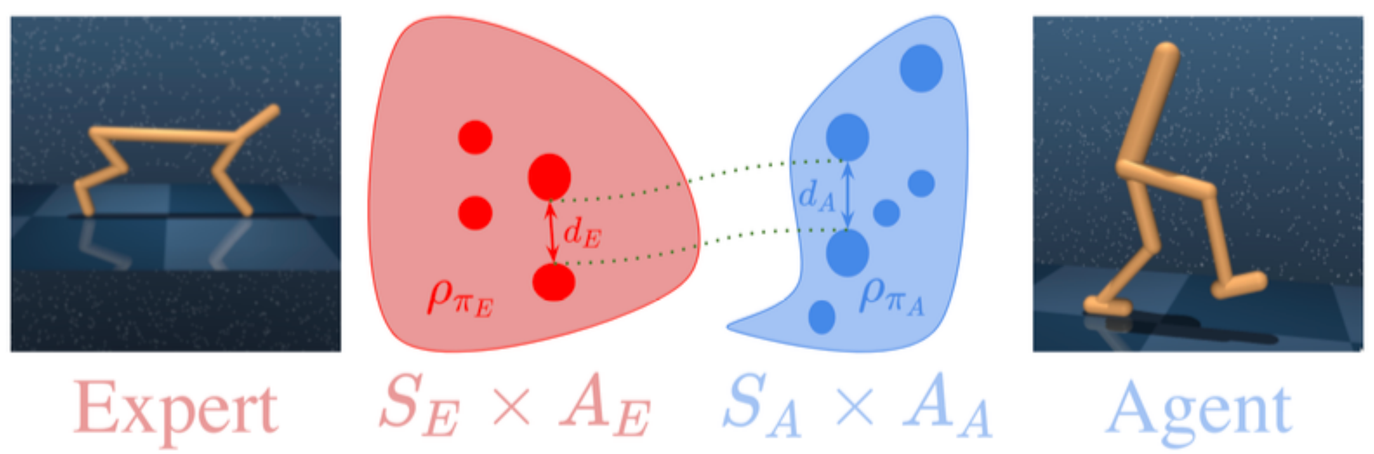
Three lightning sub-talks—slightly niche but interesting topics

Goals:

1. share from the ML side so you can understand capabilities
2. hopefully inspire new ideas and applications :)

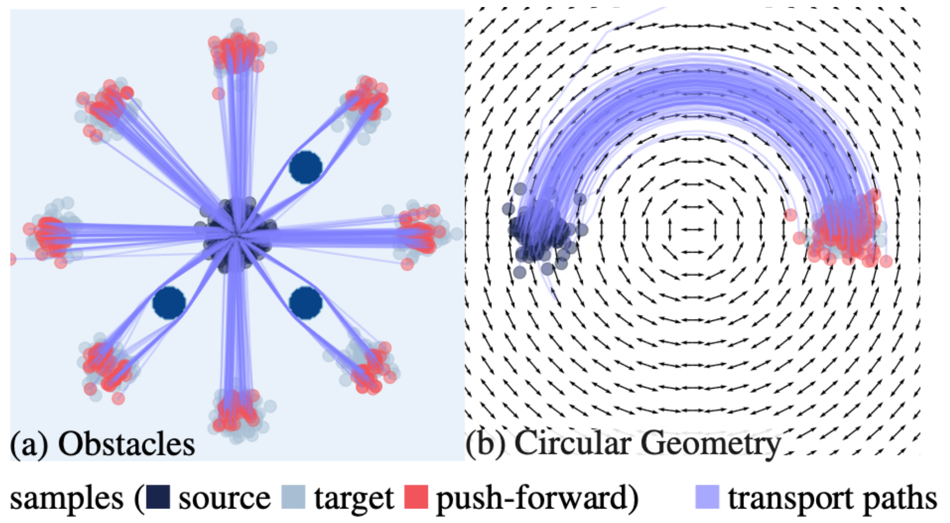
This talk

Transport for Imitation Learning



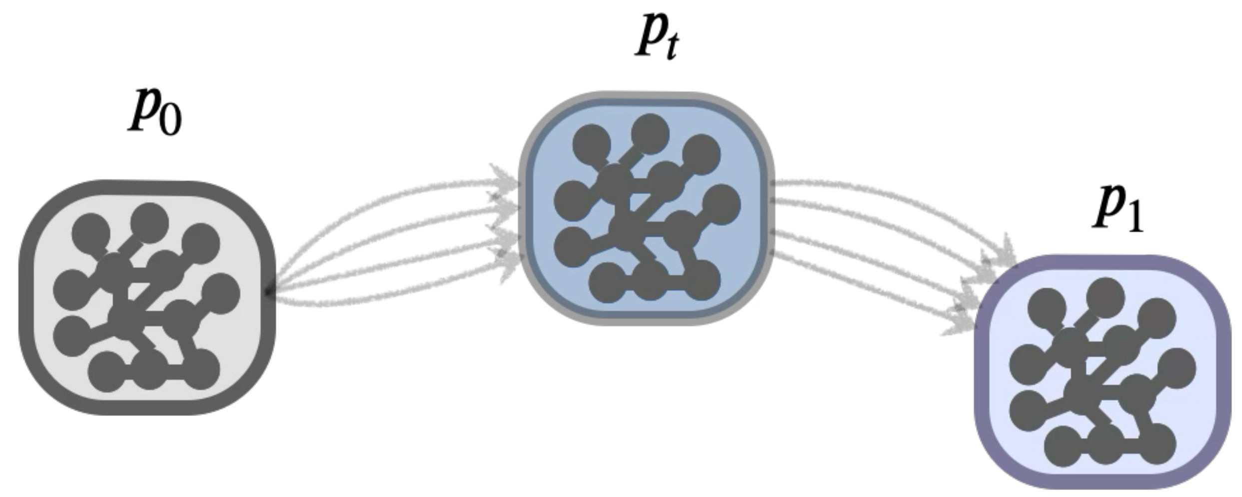
📚 Gromov-Wasserstein Imitation

Non-Euclidean Transport



📚 Lagrangian Optimal Transport

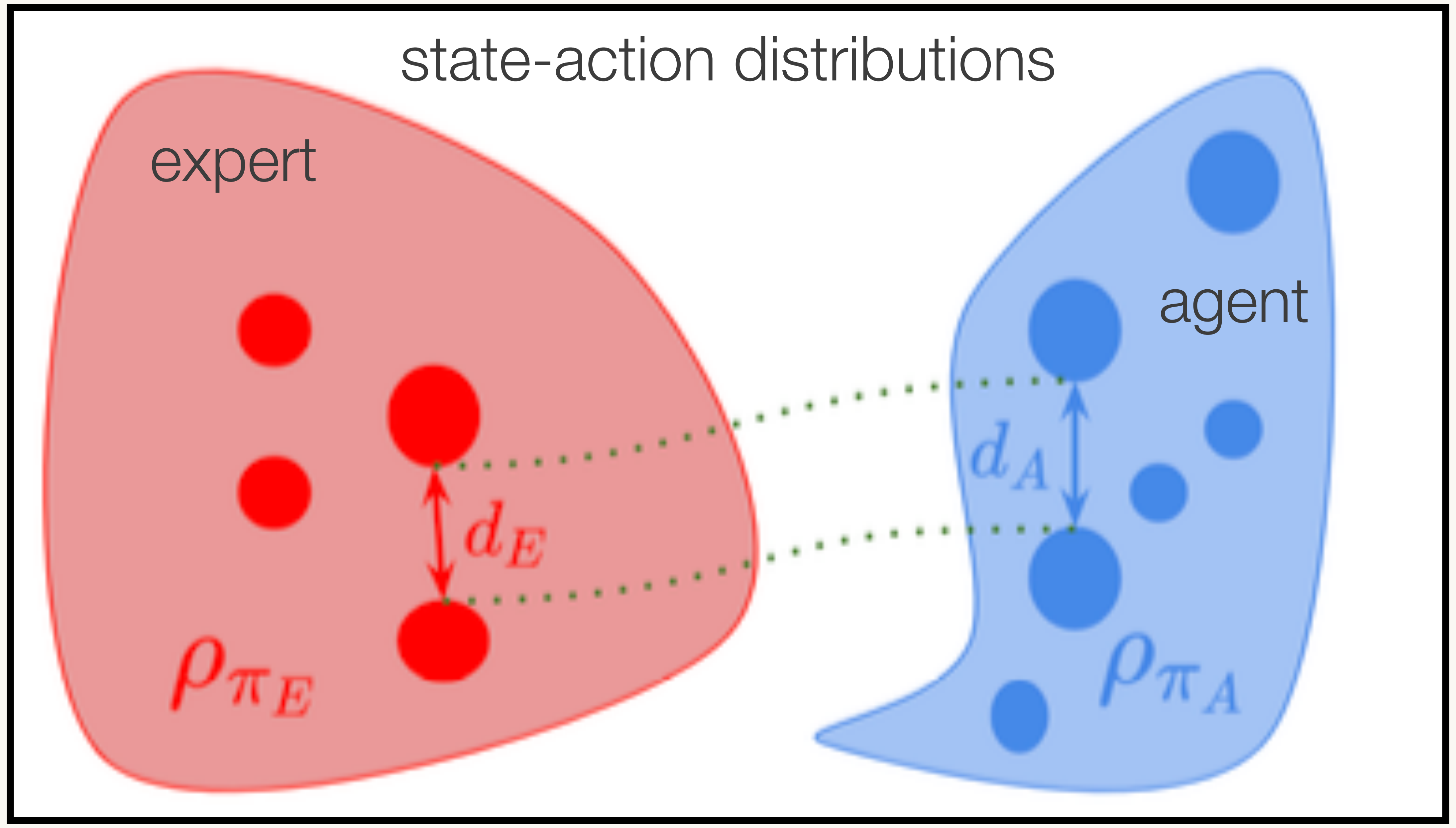
Meta and Wasserstein Flows



📚 Meta Flow Matching
📚 Wasserstein Flow Matching

Motivation: matching states in a physical system

(aka imitation learning)



One way: use optimal transport

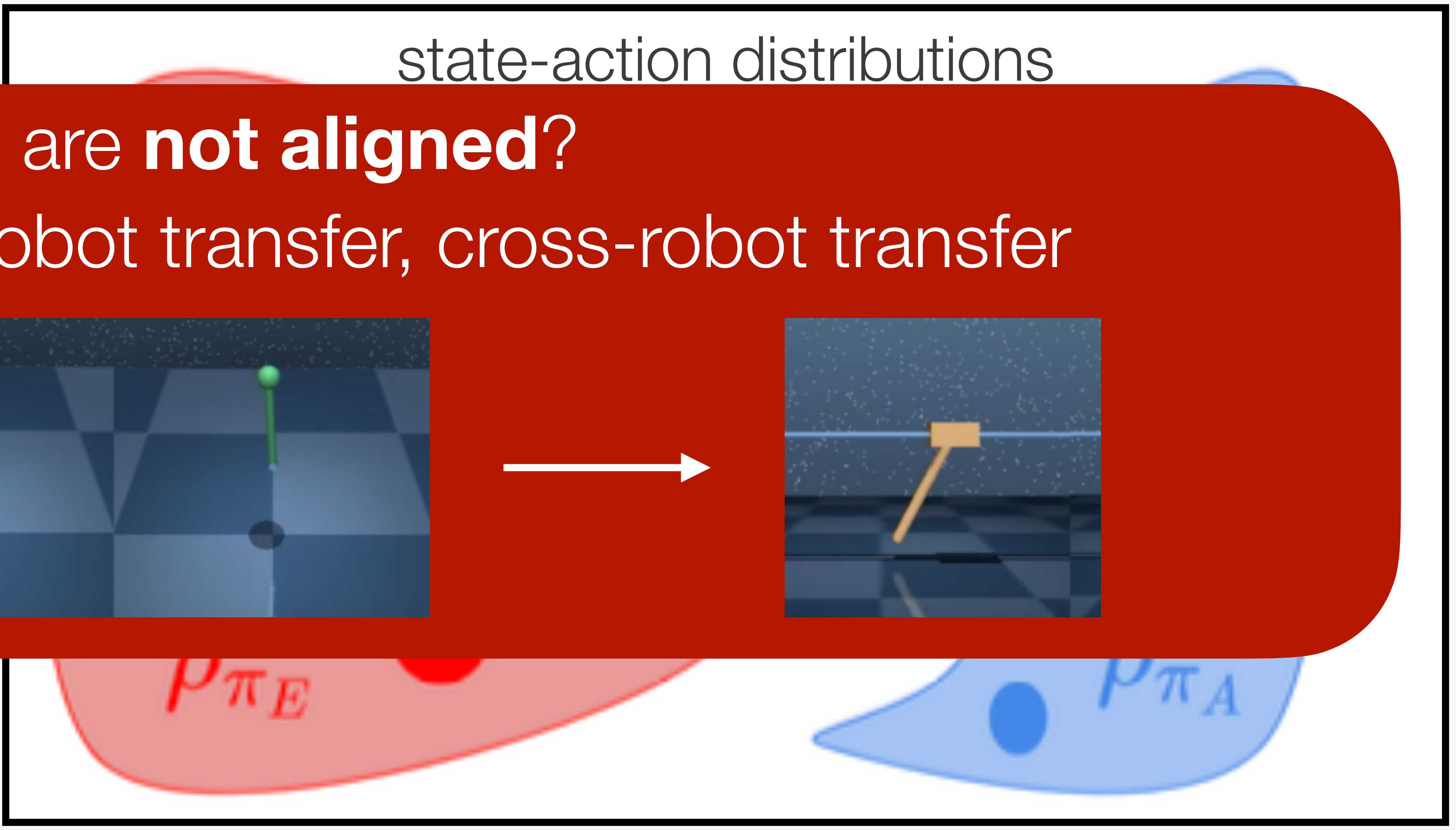
ICLR 2021

PRIMAL WASSERSTEIN IMITATION LEARNING

Robert Dadashi[*1], **Léonard Hussenot**[1,2], **Matthieu Geist**[1], **Olivier Pietquin**[1]
[1]Google Research, Brain Team
[2]Univ. de Lille, CNRS, Inria Scool, UMR 9189 CRIStAL

the MuJoCo domain in a sample efficient manner in terms of agent interactions and of expert interactions with the environment. Finally, we show that the behavior of the agent we train matches the behavior of the expert with the Wasserstein distance, rather than the commonly used proxy of performance.

$$W_p(p_0, p_1) \; := \inf_{\substack{T:X \to X \\ T_\# p_0 = p_1}} \left(\int_X c\big(x, T(x)\big)^p \, dp_0(x) \right)^{\frac{1}{p}}$$



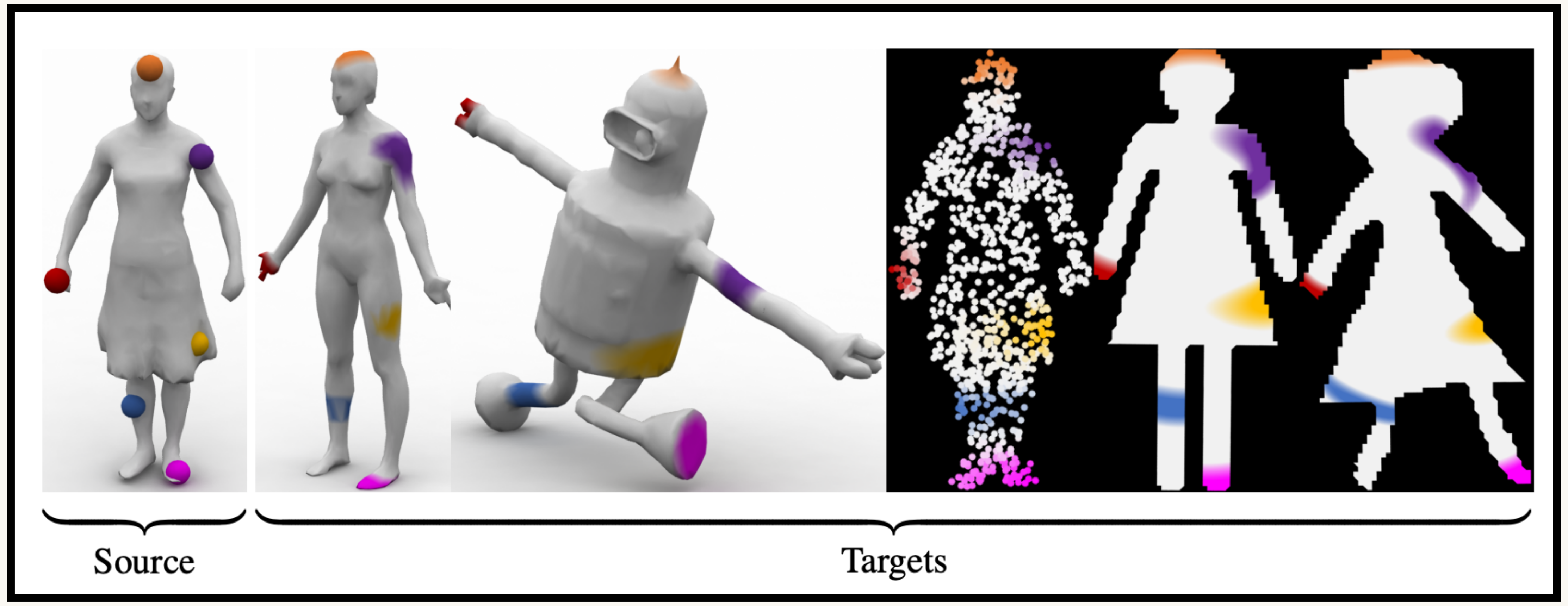
⚠️ **Challenge:** what if the state-action spaces are **not aligned**?
e.g., the **cross-domain** setting—human–robot transfer, cross-robot transfer

Use the Gromov-Wasserstein Distance!

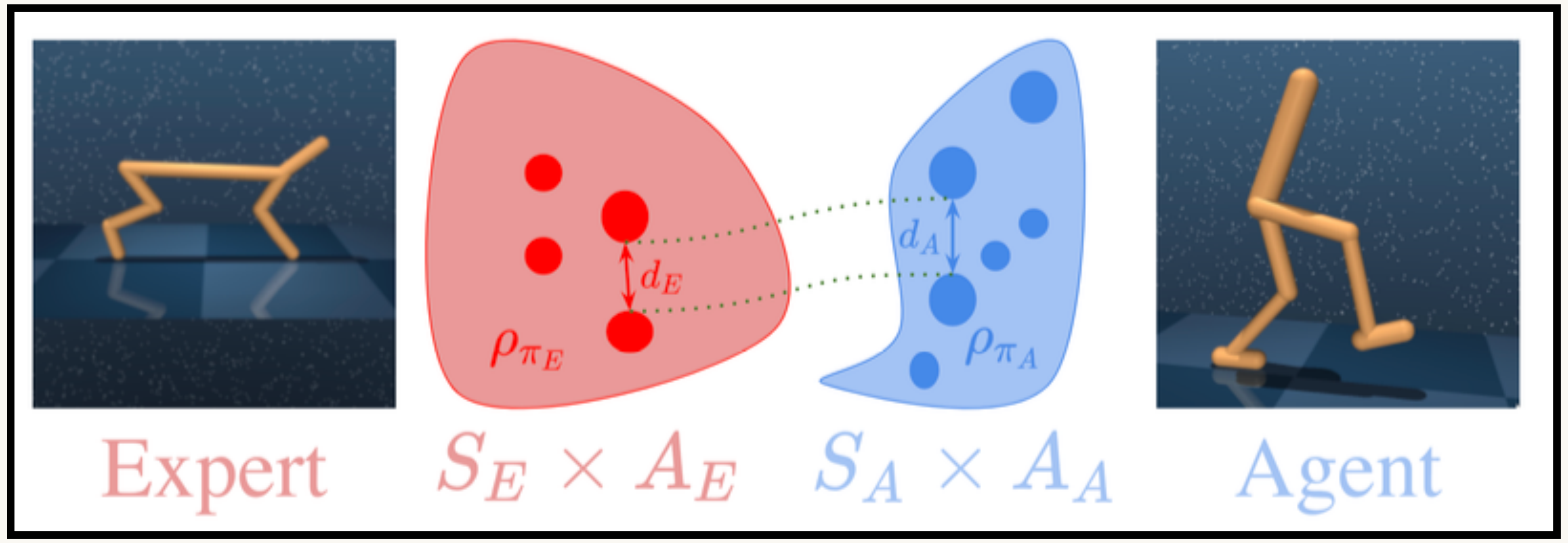
Compares **pairwise distances** between the spaces (metric-measure spaces)

$$\mathrm{GW}_p\big((X, d_X, p_0), (Y, d_Y, p_1)\big) = \inf_{\substack{T: X \to Y \\ T_\# p_0 = p_1}} \left(\iint_{X \times X} \Big| d_X(x, x') - d_Y\big(T(x), T(x')\big) \Big|^p \, dp_0(x)\, dp_0(x') \right)^{\frac{1}{p}}$$

📚 *Gromov-Wasserstein distances.* Memoli, Foundations of Computational Mathematics, 2011.



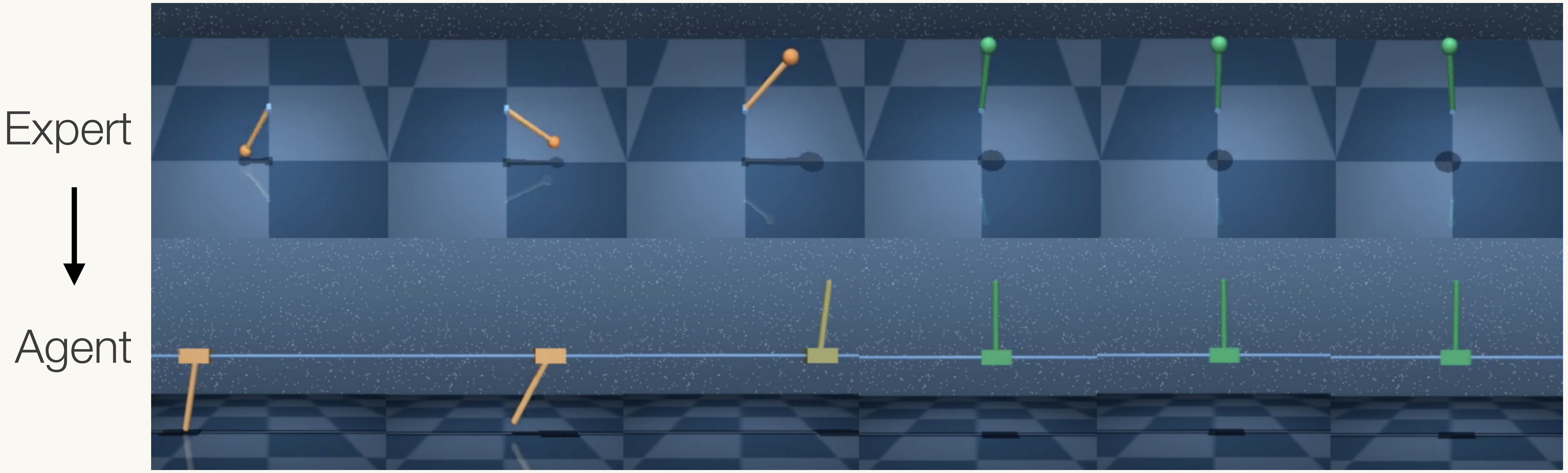
📚 *Entropic metric alignment for correspondence problems.* Solomon et al., SIGGRAPH 2016.

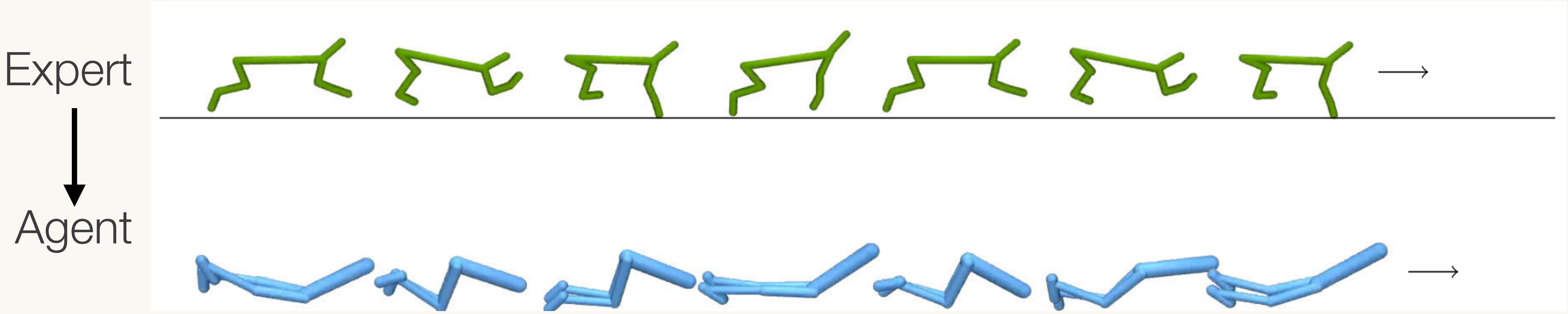


📚 *Cross-domain imitation learning via optimal transport.* Fickinger, Cohen, Russell, Amos, ICLR 2022.

Results transferring from one system to another

📚 *Cross-domain imitation learning via optimal transport.* Fickinger, Cohen, Russell, Amos, ICLR 2022.





We published it, and then... 😱

ICLR 2022 (April)

Cross-Domain Imitation Learning via Optimal Transport

Arnaud Fickinger[1,3]* **Samuel Cohen**[2,3] **Stuart Russell**[1] **Brandon Amos**[3]
[1]Berkeley AI Research [2]University College London [3]Facebook AI

Abstract

Cross-domain imitation learning studies how to leverage expert demonstrations of one agent to train an imitation agent with a different embodiment or morphology. Comparing trajectories and stationary distributions between the expert and imitation agents is challenging because they live on different systems that may not even have the same dimensionality. We propose *Gromov-Wasserstein Imitation Learning (GWIL)*, a method for cross-domain imitation that uses the Gromov-Wasserstein distance to align and compare states between the different spaces of the agents. Our theory formally characterizes the scenarios where GWIL preserves optimality, revealing its possibilities and limitations. We demonstrate the effectiveness of GWIL in non-trivial continuous control domains ranging from simple rigid transformation of the expert domain to arbitrary transformation of the state-action space. [1]

→

arXiv > cs > arXiv:2205.03476v1

Search | Help |

Computer Science > Machine Learning

[Submitted on 6 May 2022 (this version), latest version 10 May 2022 (v2)]

Issues in "Cross-Domain Imitation Learning via Optimal Transport" and a possible fix

Ruichao Jiang, Javad Tavakoli, Yiqinag Zhao

[4] proposes to use the Gromov-Wasserstein (GW) [6] distance as a proxy reward for imitation learning. We show that their approach suffers both mathematical and algorithmic issues. We use hitting-time of a Markov decision process (MDP) to fix their mathematical issues and discuss the difficulty behind the algorithmic issue. To our best knowledge, we are the first to define the first-hitting time in the context of MDP.

Subjects: **Machine Learning (cs.LG)**
Cite as: arXiv:2205.03476 **[cs.LG]**
(or arXiv:2205.03476v1 **[cs.LG]** for this version)
https://doi.org/10.48550/arXiv.2205.03476

Submission history

From: Ruichao Jiang [view email]
[v1] Fri, 6 May 2022 21:14:46 UTC (11 KB)
[v2] Tue, 10 May 2022 21:05:13 UTC (8 KB)

Thankfully the issue was minor

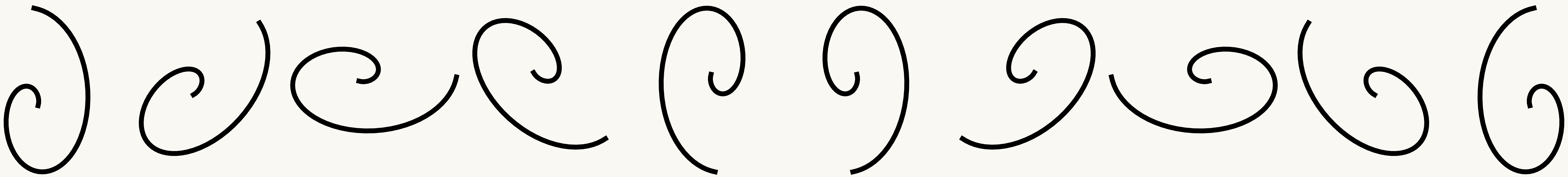
We forgot to include the pushforward term in our isomorphic definition (sorry) 😅

Definition 2 (Isomorphic policies). *Two policies π_E and π_A are isomorphic if there exists a bijection $\phi : \mathrm{supp}[\rho_{\pi_E}] \to \mathrm{supp}[\rho_{\pi_A}]$ that satisfies for all $(s_E, a_E), (s_E{}', a_E{}') \in \mathrm{supp}[\rho_{\pi_E}]$ and $(s_A, a_A) \in \mathrm{supp}[\rho_{\pi_A}]$:*

$$d_E\left((s_E, a_E), (s_E{}', a_E{}')\right) = d_A\left(\phi(s_E, a_E), \phi(s_E{}', a_E{}')\right) \tag{3}$$

$$\rho_{\pi_A}(s_A, a_A) = \rho_{\pi_E}(\phi^{-1}(s_E, a_E)) \tag{4}$$

In other words, ϕ is an isometry between $(\mathrm{supp}[\rho_{\pi_E}], d_E)$ and $(\mathrm{supp}[\rho_{\pi_A}], d_A)$ and ρ_{π_A} is the push-forward measure $\phi_\sharp(\rho_{\pi_E})$.



Easy fix, everybody happy 🎉

arXiv > cs > arXiv:2205.03476v1

Search...

Help |

Computer Science > Machine Learning

[Submitted on 6 May 2022 (this version), latest version 10 May 2022 (v2)]

Issues in "Cross-Domain Imitation Learning via Optimal Transport" and a possible fix

Ruichao Jiang, Javad Tavakoli, Yiqinag Zhao

[4] proposes to use the Gromov-Wasserstein (GW) [6] distance as a proxy reward for imitation learning. We show that their approach suffers both mathematical and algorithmic issues. We use hitting-time of a Markov decision process (MDP) to fix their mathematical issues and discuss the difficulty behind the algorithmic issue. To our best knowledge, we are the first to define the first-hitting time in the context of MDP.

Subjects: **Machine Learning (cs.LG)**

Cite as: arXiv:2205.03476 **[cs.LG]**

(or arXiv:2205.03476v1 **[cs.LG]** for this version)

https://doi.org/10.48550/arXiv.2205.03476

Submission history

From: Ruichao Jiang [view email]

[v1] Fri, 6 May 2022 21:14:46 UTC (11 KB)

[v2] Tue, 10 May 2022 21:05:13 UTC (8 KB)

4 days later →

arXiv > cs > arXiv:2205.03476v2

Search...

Help | A

Computer Science > Machine Learning

[Submitted on 6 May 2022 (v1), last revised 10 May 2022 (this version, v2)]

Hitting time for Markov decision process

Ruichao Jiang, Javad Tavakoli, Yiqinag Zhao

We define the hitting time for a Markov decision process (MDP). We do not use the hitting time of the Markov process induced by the MDP because the induced chain may not have a stationary distribution. Even it has a stationary distribution, the stationary distribution may not coincide with the (normalized) occupancy measure of the MDP. We observe a relationship between the MDP and the PageRank. Using this observation, we construct an MP whose stationary distribution coincides with the normalized occupancy measure of the MDP and we define the hitting time of the MDP as the hitting time of the associated MP.

Comments: The first version of this paper pointed out some issues in the old version of "Cross-Domain Imitation Learning via Optimal Transport". The authors have then addressed these issues according to our suggestions in a new version. We therefore updated our paper, in which we removed contents related to these issues

Subjects: **Machine Learning (cs.LG)**

Cite as: arXiv:2205.03476 **[cs.LG]**

(or arXiv:2205.03476v2 **[cs.LG]** for this version)

https://doi.org/10.48550/arXiv.2205.03476

Submission history

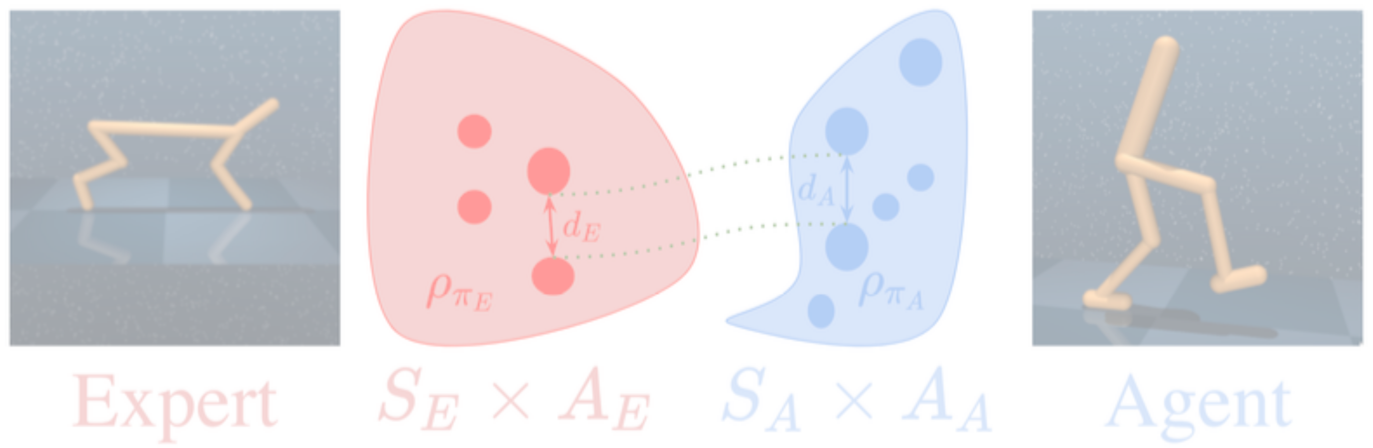
From: Ruichao Jiang [view email]

[v1] Fri, 6 May 2022 21:14:46 UTC (11 KB)

[v2] Tue, 10 May 2022 21:05:13 UTC (8 KB)

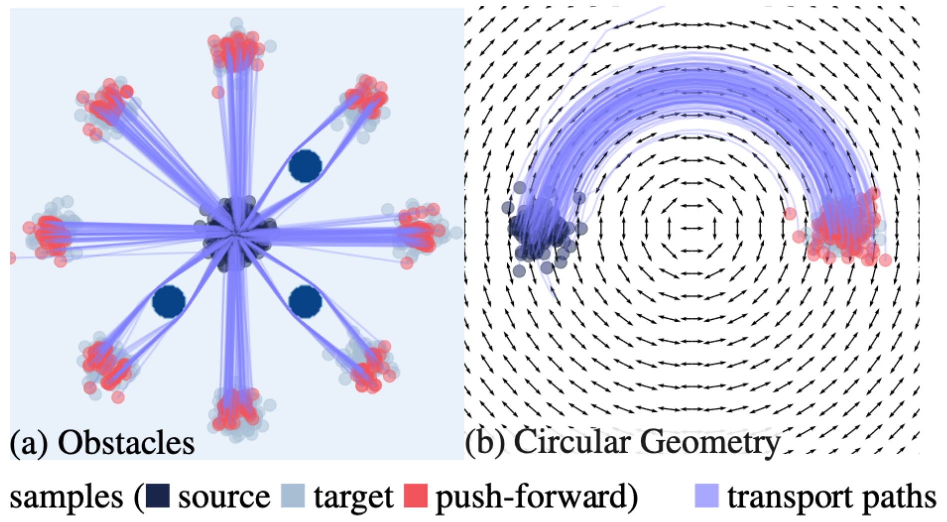
This talk

Transport for Imitation Learning



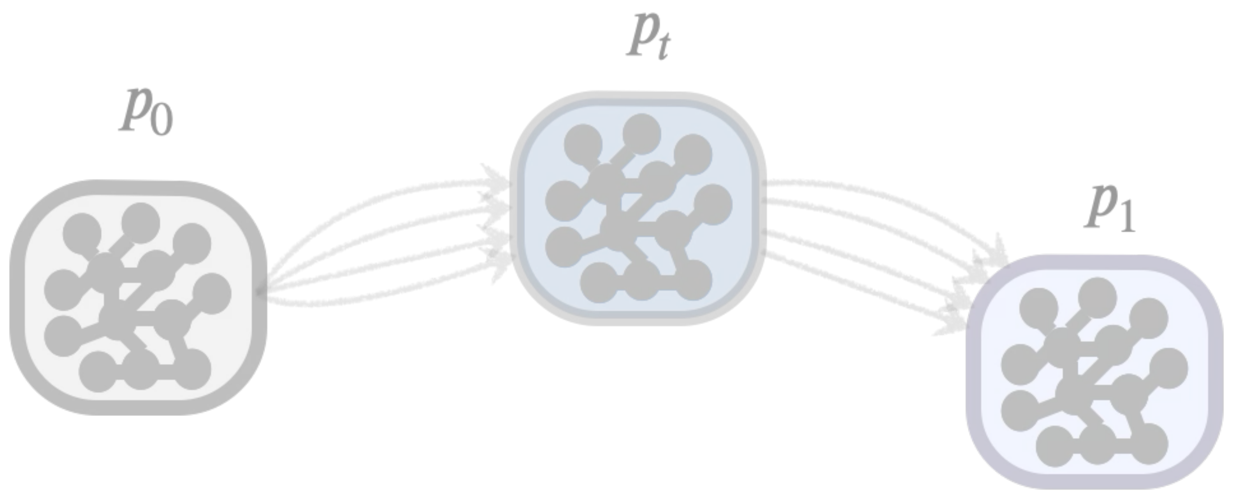
📚 Gromov-Wasserstein Imitation

Non-Euclidean Transport



📚 Lagrangian Optimal Transport

Meta and Wasserstein Flows



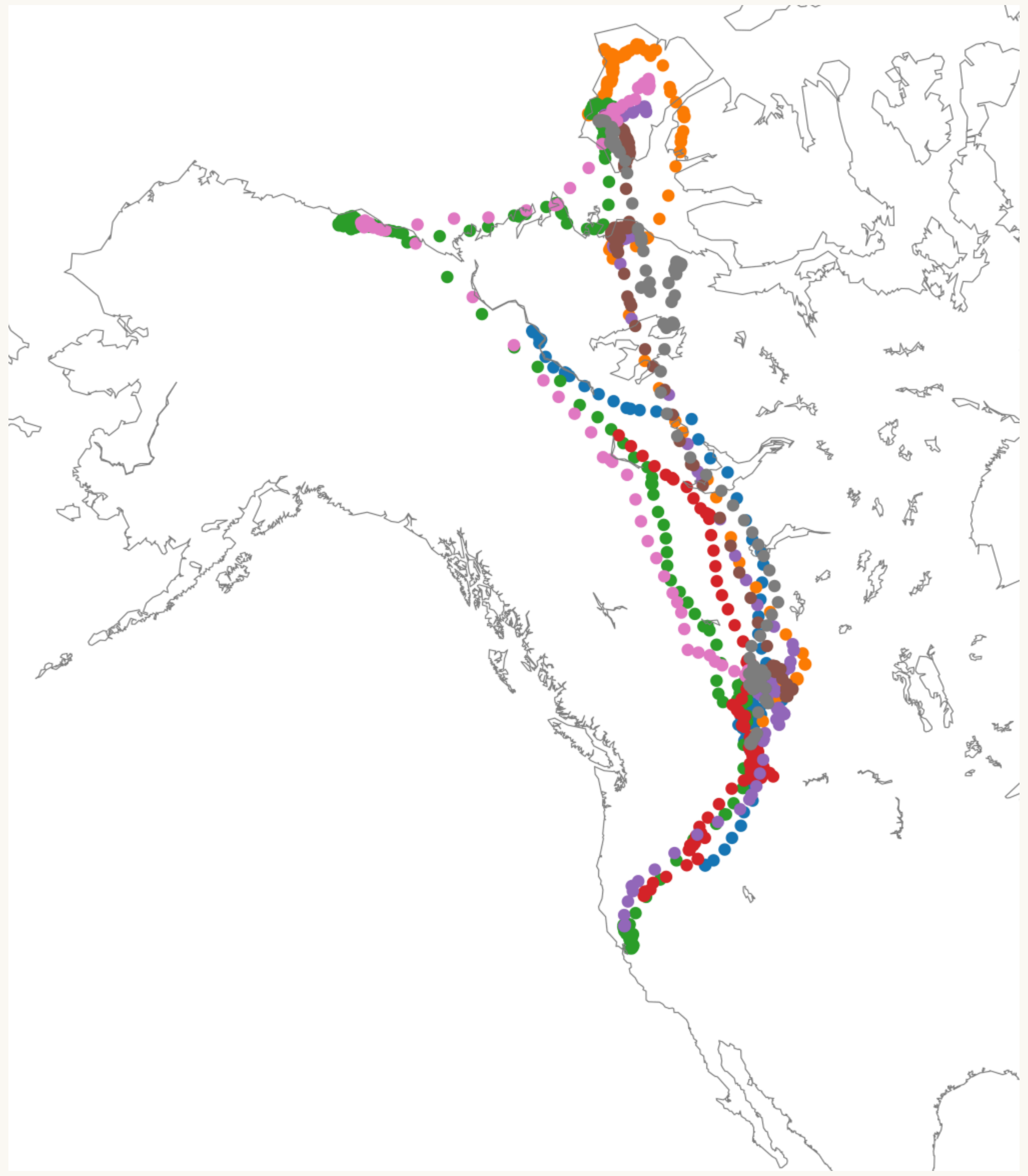
📚 Meta Flow Matching

📚 Wasserstein Flow Matching

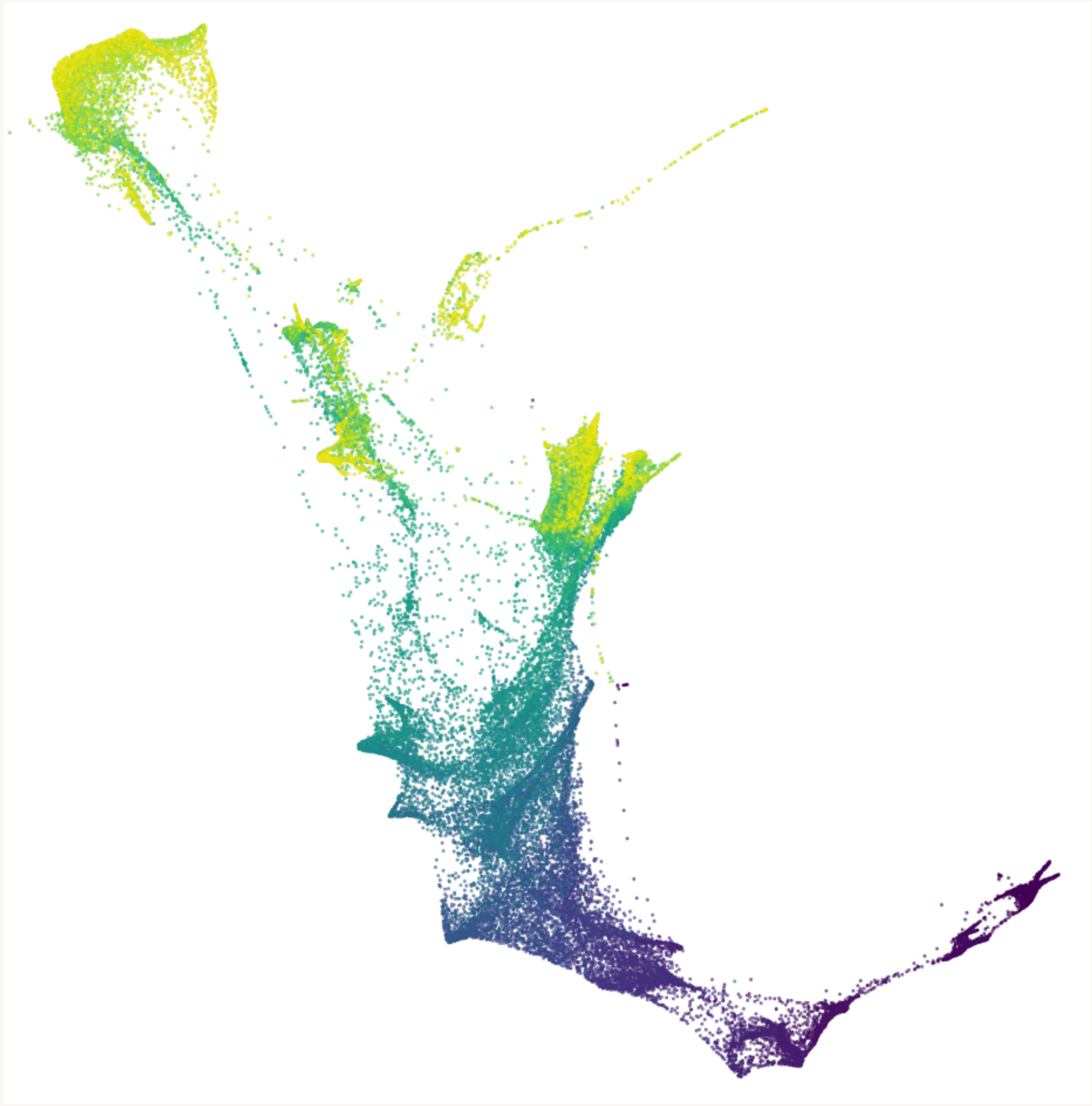
Motivation: transport under physical constraints

$$W_p(p_0, p_1) \;:=\; \inf_{\substack{T:X \to X \\ T_\# p_0 = p_1}} \left(\int_X c\big(x, T(x)\big)^p \, dp_0(x) \right)^{\frac{1}{p}}$$

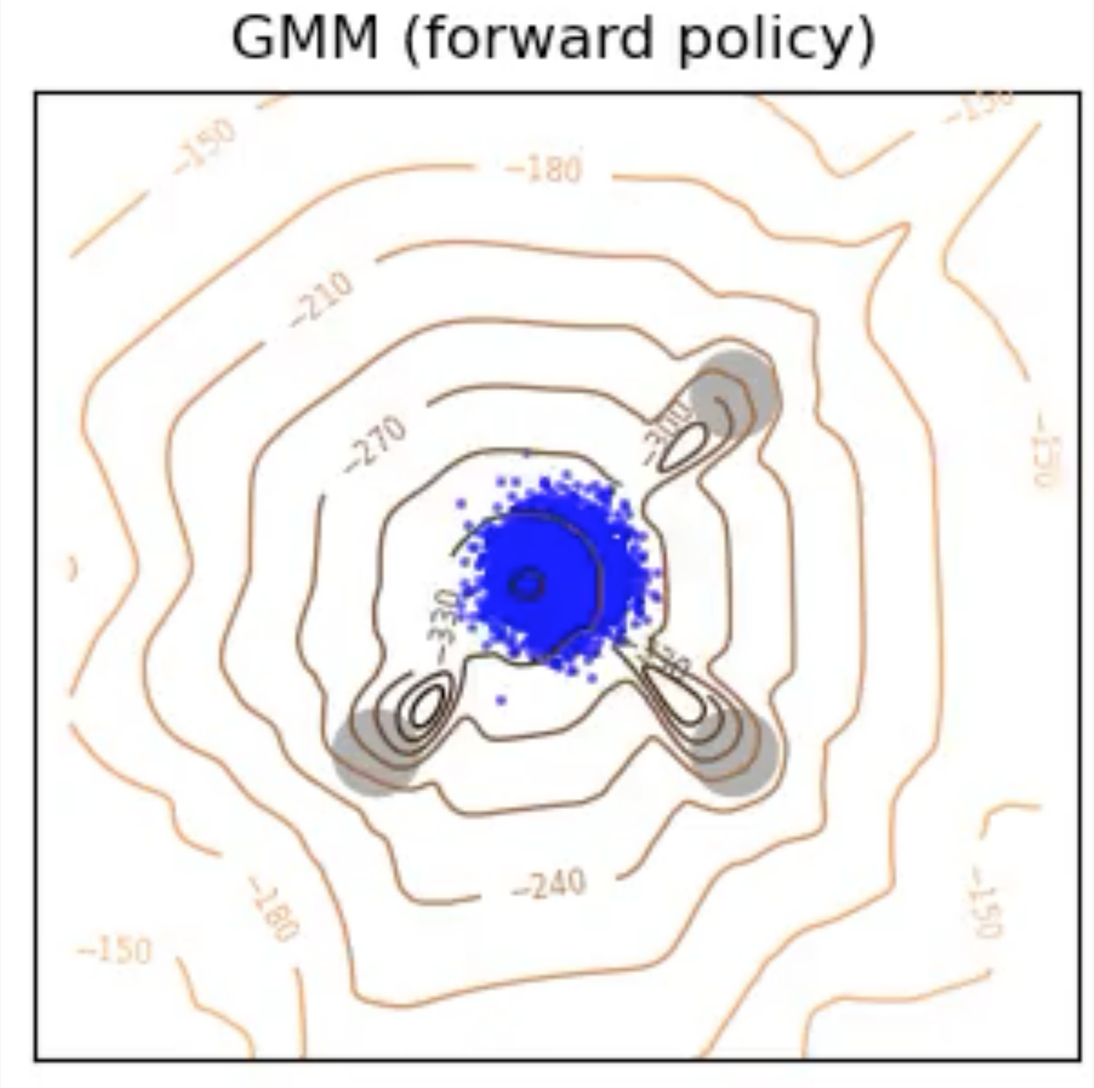
Bird population dynamics



Cell population dynamics



Physical obstacles



📚 Riemannian Metric Learning via Optimal Transport. Scarvelis and Solomon, ICLR 2023.

📚 Deep Generalized Schrödinger Bridge. Liu et al., NeurIPS 2022.

Focus: minimum-action Lagrangian paths

Squared Euclidean

$$c(x,y) = \|x-y\|_2^2$$

Lagrangian (e.g., geodesics)

$$c(x,y) = \inf_{\gamma \in \mathcal{C}(x,y)} \int_0^1 \mathcal{L}(\gamma_t, \dot{\gamma}_t)\mathrm{d}t$$

Euclidean $c(x,y) = \|x-y\|_2^2$

$$\mathcal{L}(\gamma_t, \dot{\gamma}_t) = \frac{1}{2}\|\dot{\gamma}_t\|_2^2$$

Potential term (e.g., obstacles)

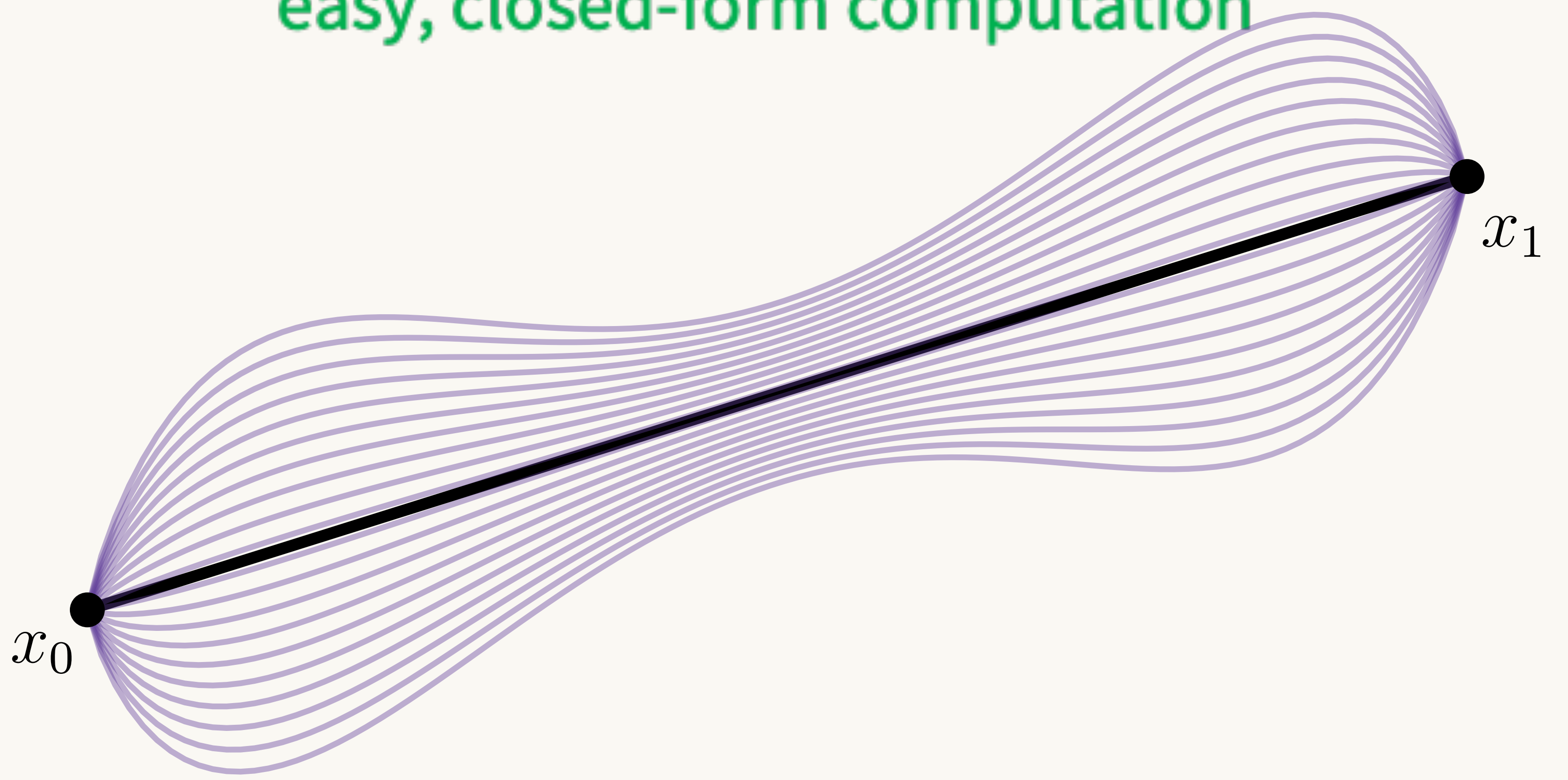
$$\mathcal{L}(\gamma_t, \dot{\gamma}_t) = \frac{1}{2}\|\dot{\gamma}_t\|_2^2 - U(\gamma_t)$$

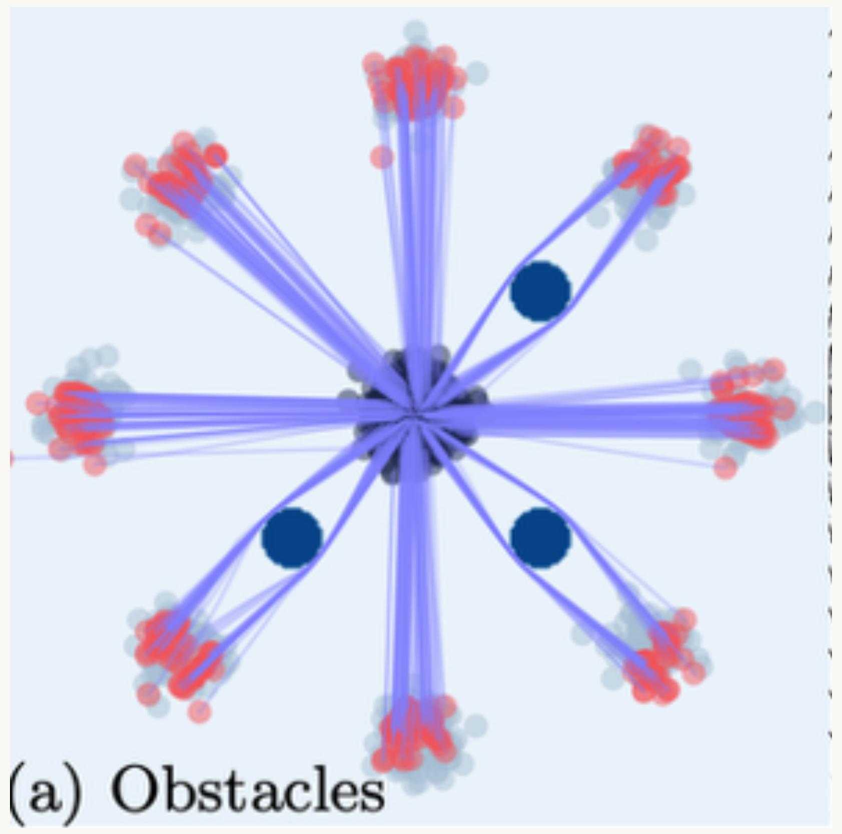
Riemannian geodesics

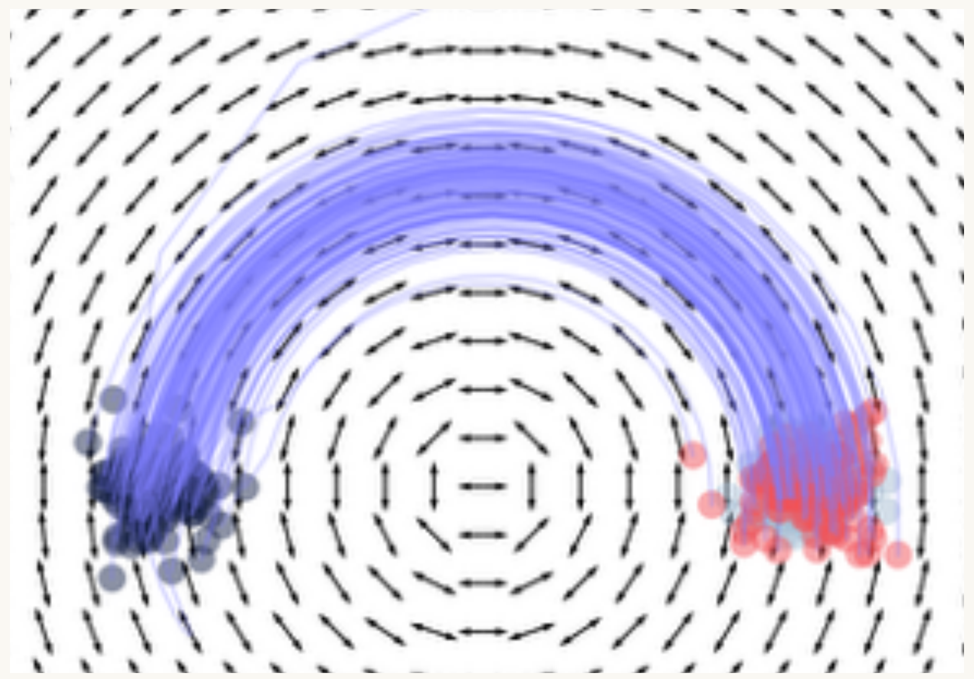
$$\mathcal{L}(\gamma_t, \dot{\gamma}_t) = \frac{1}{2}\|\dot{\gamma}_t\|_{A(\gamma_t)}^2$$

easy, closed-form computation

challenging in general, no known closed-form solutions







How to solve?

📚 *Neural Optimal Transport with Lagrangian Costs*. Pooladian, Domingo-Enrich, Chen, Amos, UAI 2024.

Primal formulations

Monge

$$\inf_{T\in\mathcal{T}(p_0,p_1)} \int_{\mathcal{X}} c(x,T(x))\mathrm{d}p_0(x)$$

Kantorovich

...

Kantorovich dual

$$\hat{\psi}(p_0,p_1,c) \in \operatorname*{argsup}_{\psi\in L^1(p_0)} \int_{\mathcal{Y}} \psi^c(y)dp_1(y) - \int_{\mathcal{X}} \psi(x)dp_0(x)$$

neural network

like other neural OT

c-transform

$$\psi^c(y) \stackrel{\text{def}}{=} \inf_x \psi(x) + c(x,y)$$

neural network

📚 *On amortizing convex conjugates.* Amos, ICLR 2023.

Costs

Squared Euclidean

$$c(x,y) = \|x-y\|_2^2$$

Lagrangian (e.g., geodesics)

$$c(x,y) = \inf_{\gamma\in\mathcal{C}(x,y)} \int_0^1 \mathcal{L}(\gamma_t,\dot{\gamma}_t)\mathrm{d}t$$

neural network

predict parameters of a spline

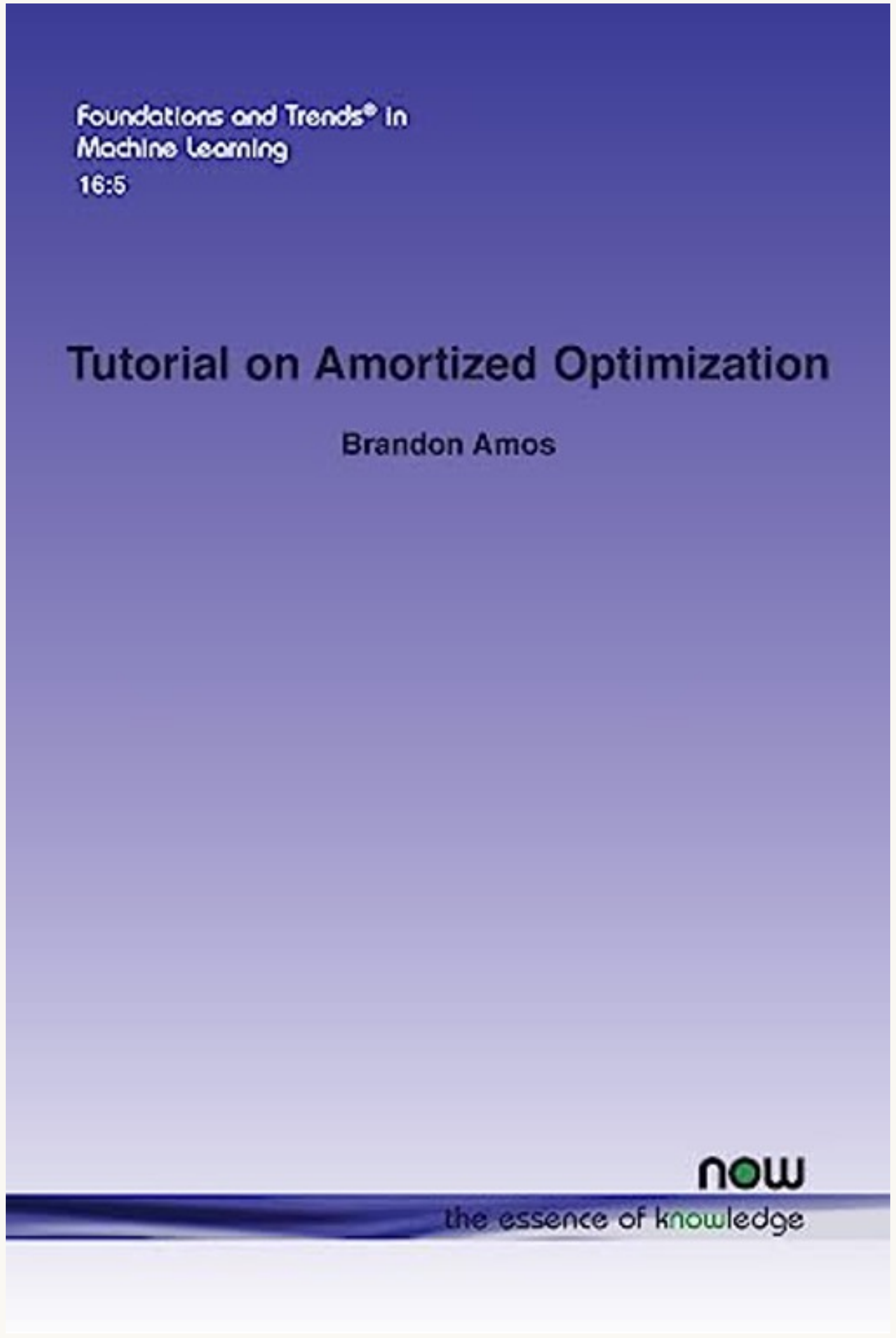
Other applications to HJB, see also:

📚 *Deep Legendre Transform.* Minabutdinov, Cheridito. NeurIPS 2025.

And then? Can also **learn the cost/metric** with all of this as a subproblem

$$\min_c J(T^\star(c))$$

Aside: predicting solutions to optimization problems



Reinforcement learning and control
actor-critic methods, SAC, DDPG, GPS, BC

Variational inference
VAEs, semi-amortized VAEs

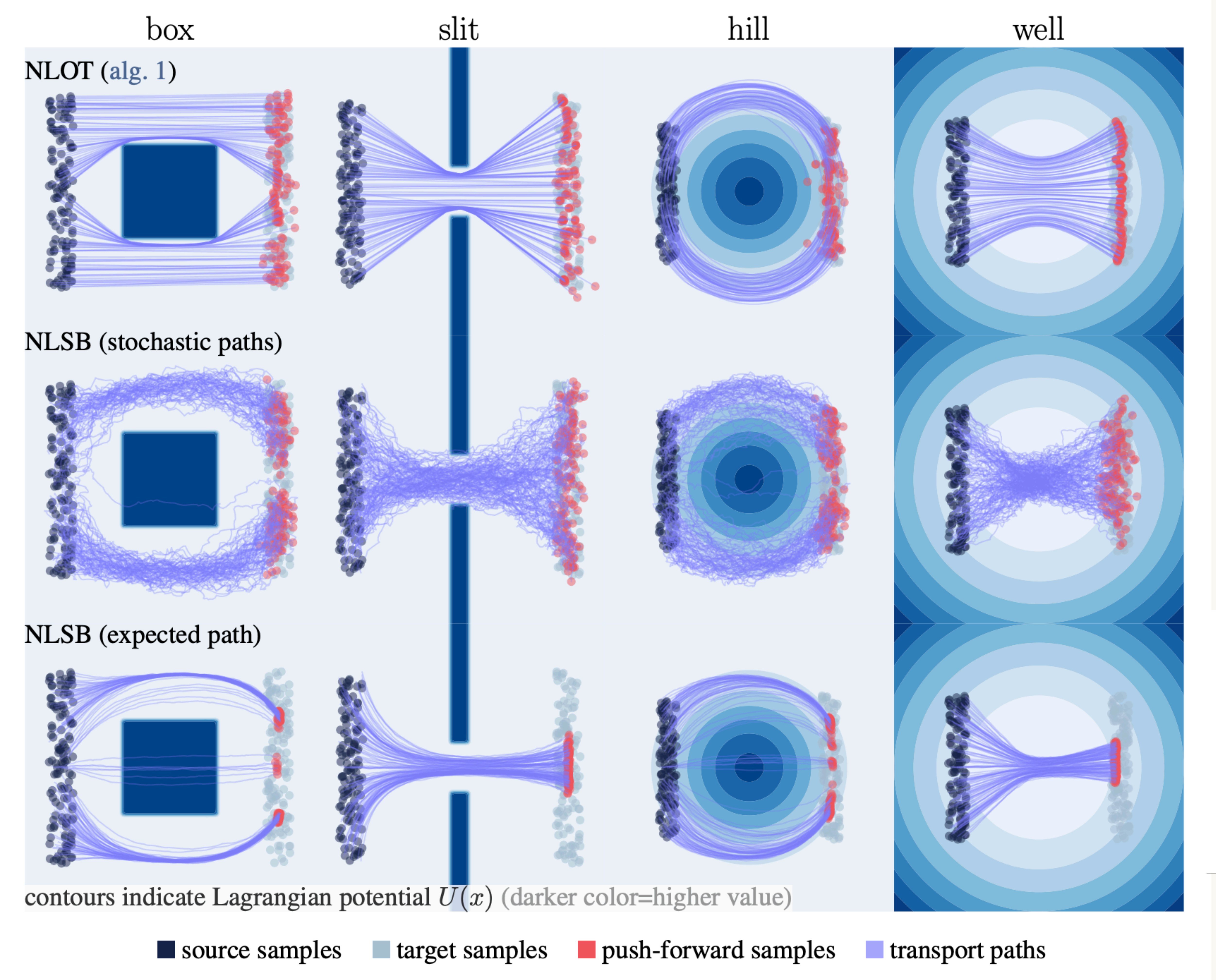
Meta-learning
HyperNets, MAML

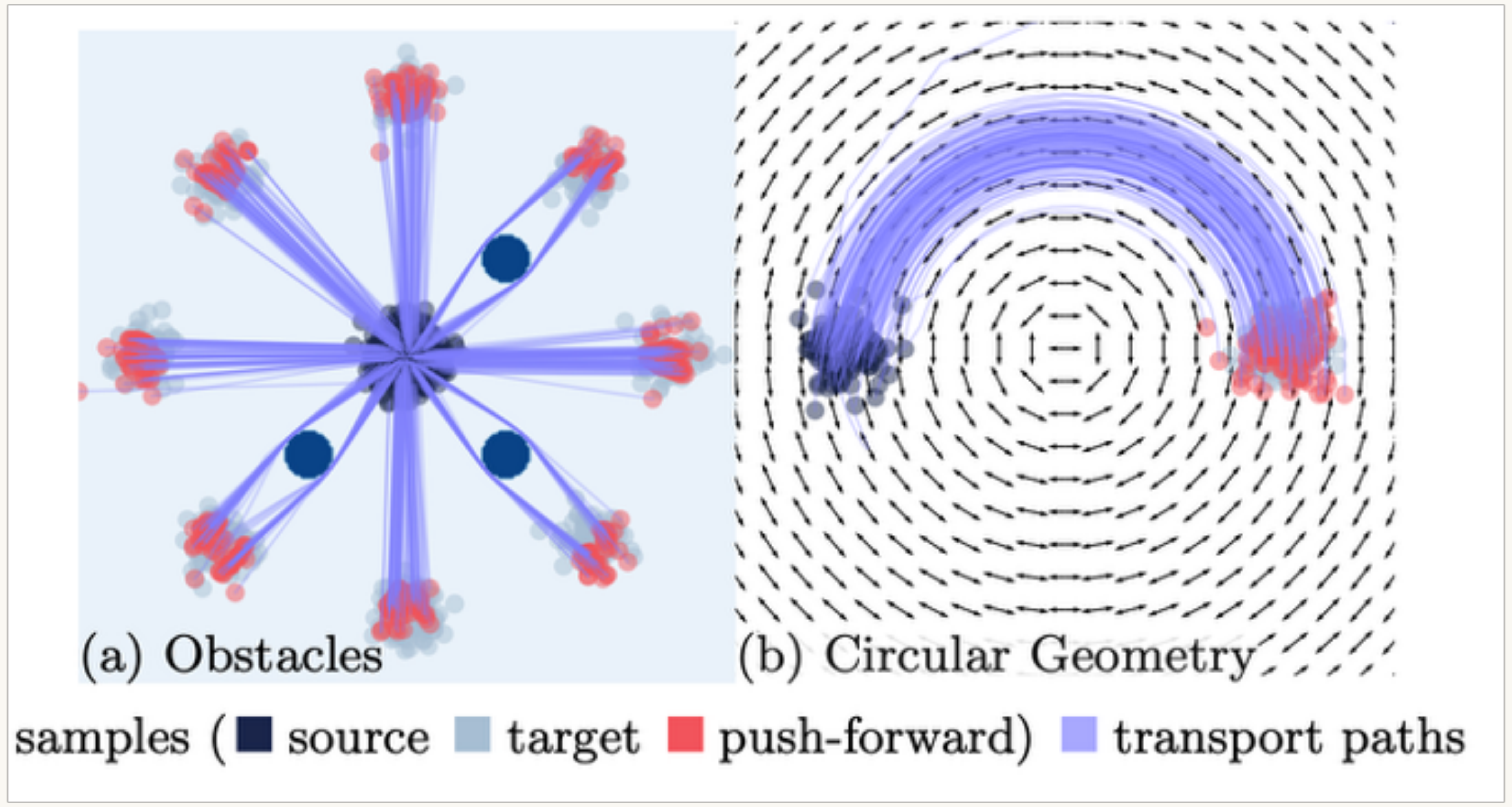
Sparse coding
PSD, LISTA

Roots, fixed points, and convex optimization
NeuralDEQs, RLQP, NeuralSCS

Optimal transport
slicing, conjugation, Meta Optimal Transport

Lagrangian Transport Results





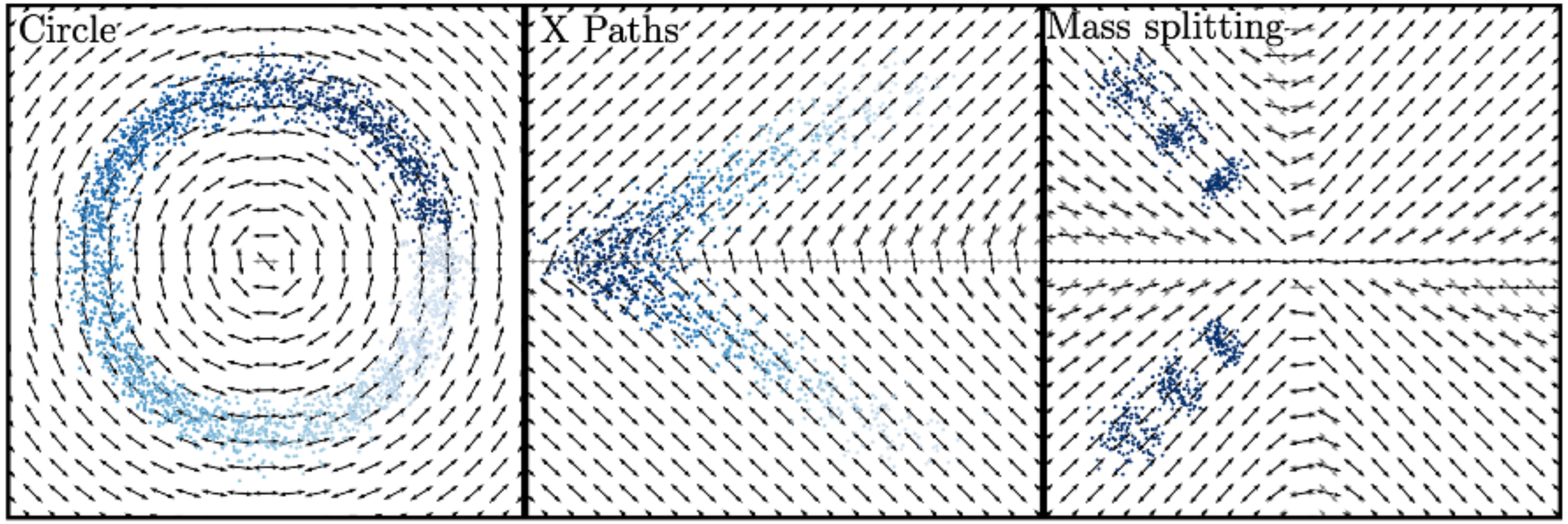
	box	slit	hill	well
NLOT (ours)	**1.6 ± 0.2**	**1.3 ± 0.2**	**1.8 ± 1.3**	**1.3 ± 0.3**
NLSB (stochastic)	2.4 ± 0.6	**1.3 ± 0.4**	**2.0 ± 0.1**	2.6 ± 1.6
NLSB (expected)	6.0 ± 0.5	17.6 ± 1.8	4.0 ± 0.9	16.1 ± 3.5

*Results are from training three trials for every method.

Metric Learning Results

Table 2: Alignment scores $\ell_{\text{align}} \in [0, 1]$ for metric recovery in Fig. 4. (higher is better)

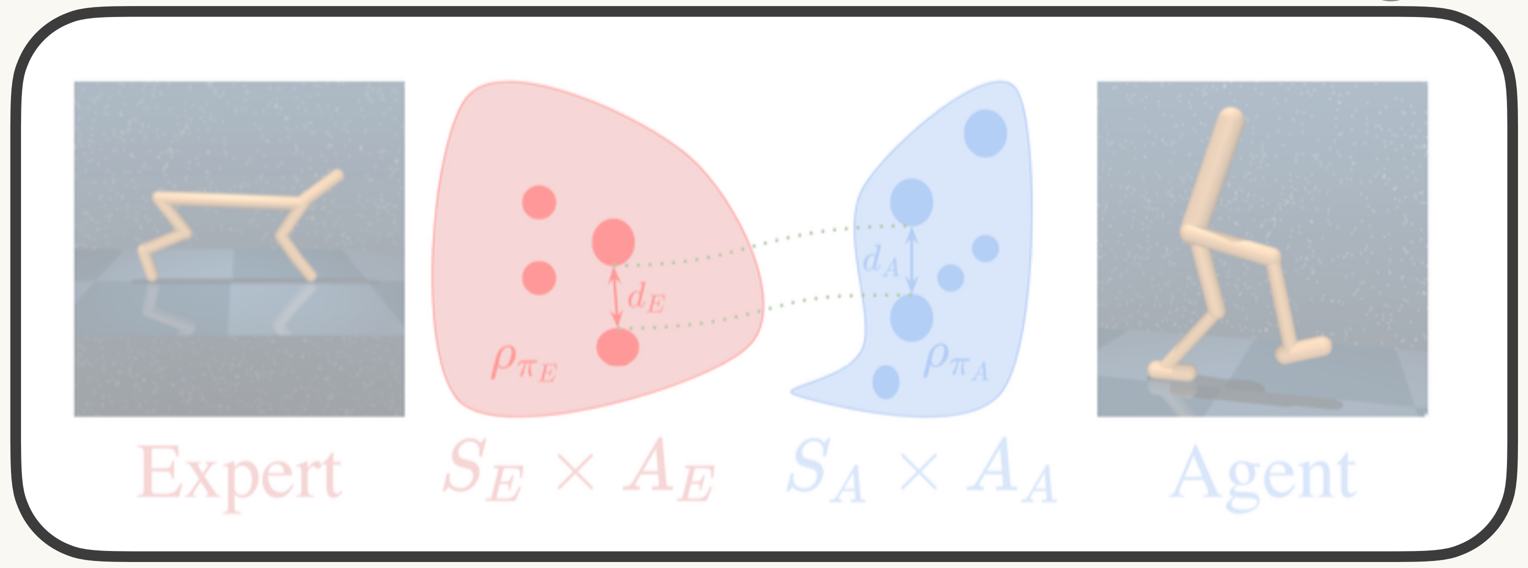
	Circle	Mass Splitting	X Paths
Metric learning with NLOT (ours)	$\mathbf{0.997 \pm 0.002}$	$\mathbf{0.986 \pm 0.001}$	$\mathbf{0.957 \pm 0.001}$
Scarvelis and Solomon (2023)	0.995	0.839	0.916



smallest eigenvectors of A (■ learned ■ ground-truth)
■ data (lighter colors=later time)

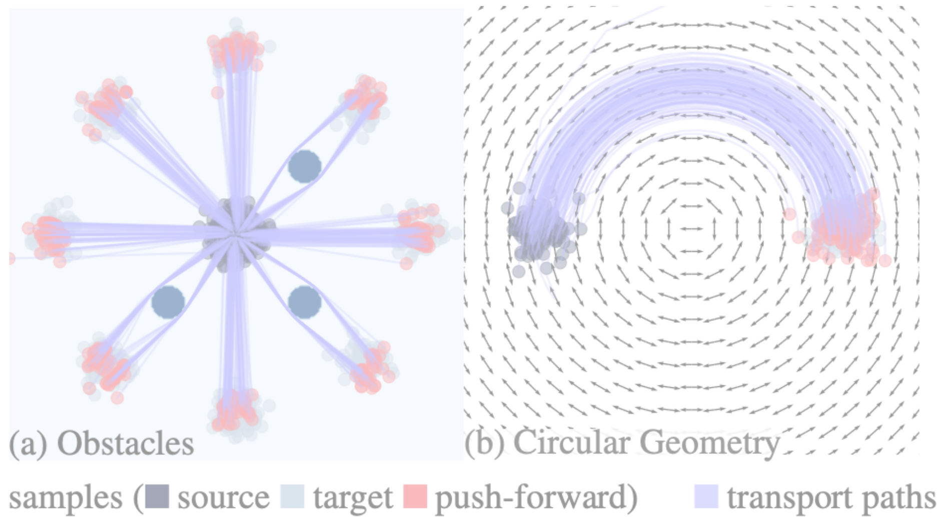
This talk

Transport for Imitation Learning



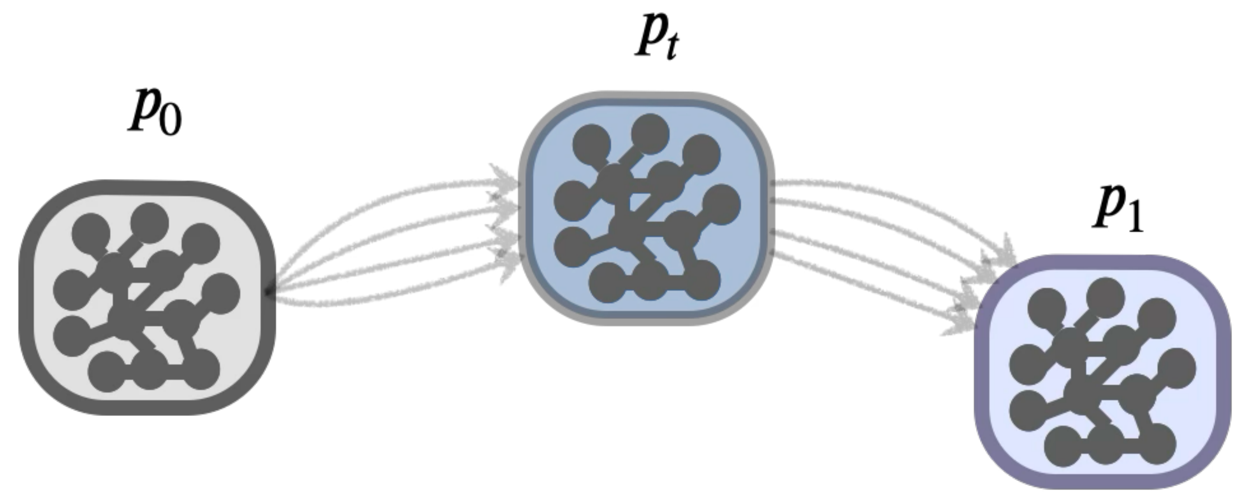
📚 Gromov-Wasserstein Imitation

Non-Euclidean Transport



📚 Lagrangian Optimal Transport

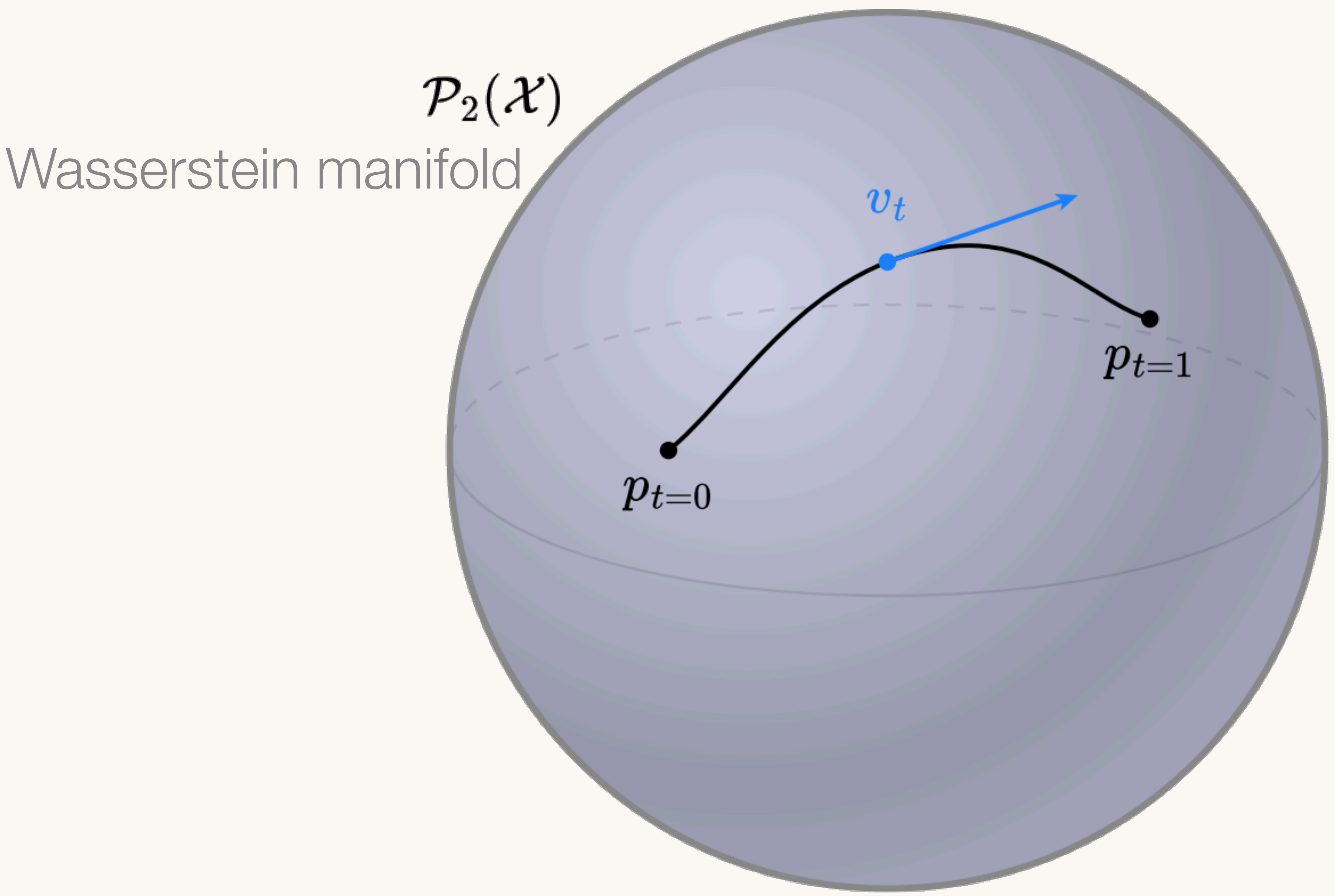
Meta and Wasserstein Flows



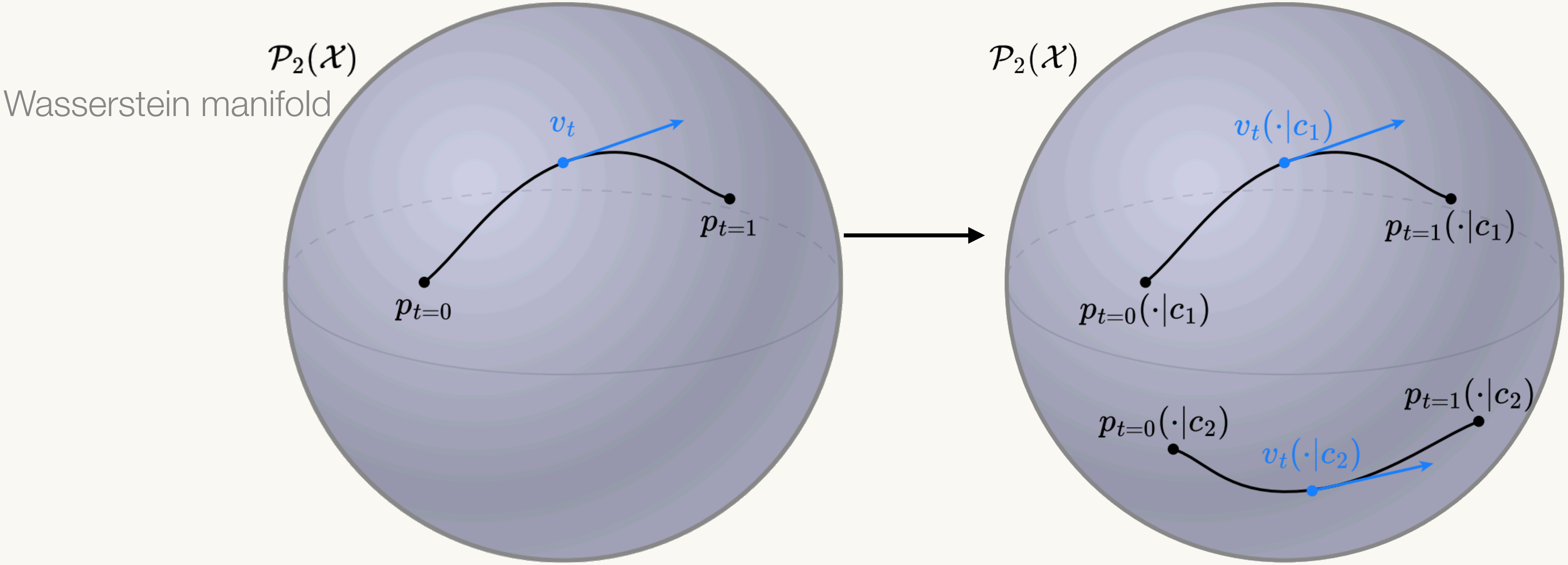
📚 Meta Flow Matching

📚 Wasserstein Flow Matching

So far: single distribution to single distribution



Next: multiple distributions 🚀

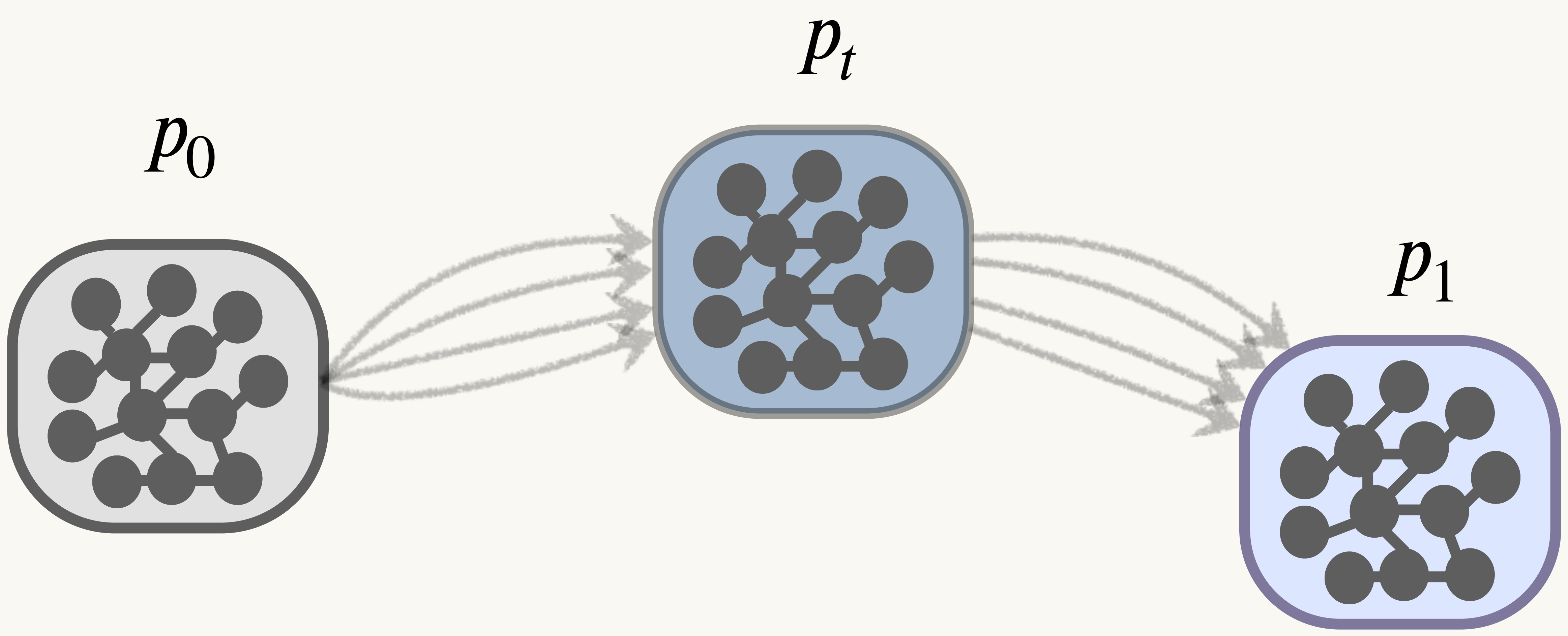


📚 *Meta Flow Matching*. Atanackovic*, Zhang*, Amos, Blanchette, Lee, Bengio, Tong, Neklyudov. ICLR 2025.

Background and Motivation

Scientific goal: understand and model the **dynamics of many-body systems** (the dynamic evolution of interacting particles)

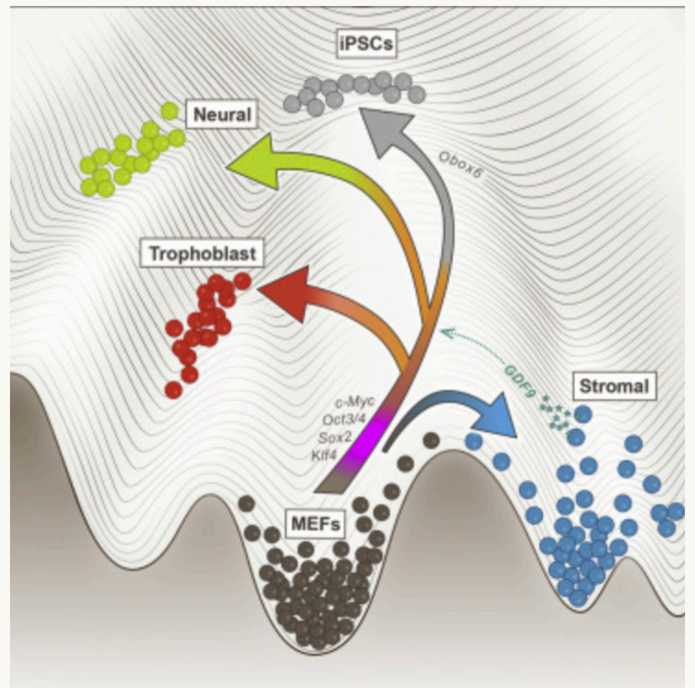
E.g. the dynamic processes **cells** undergo w.r.t. their **environment** and **interactions** with each other



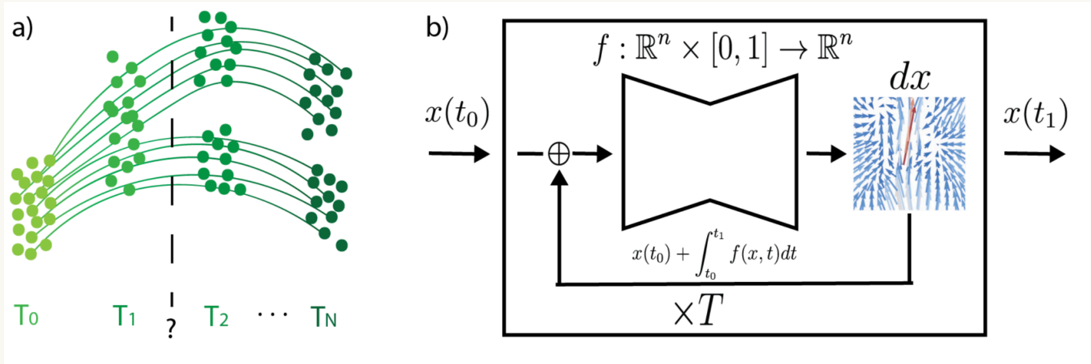
Background and Motivation

We want to model the dynamics of particles (or cells) at the **population level**.

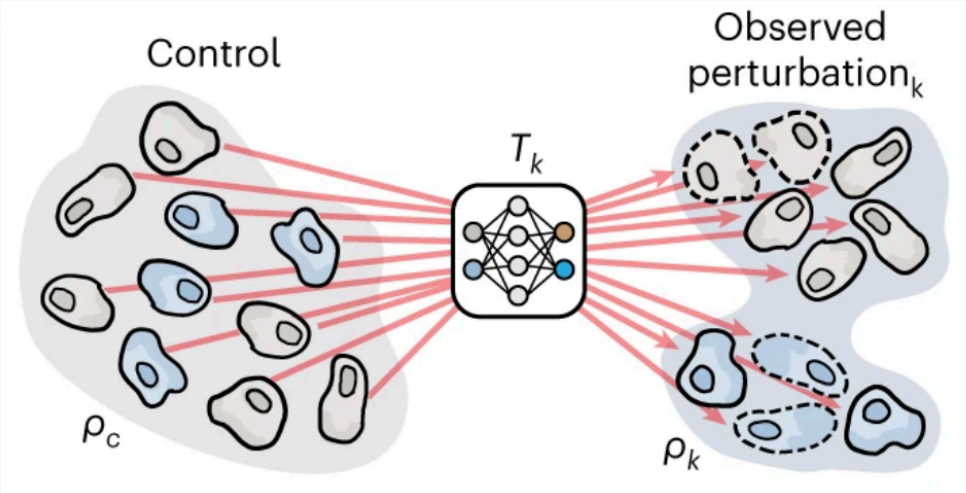
Many methods do this:



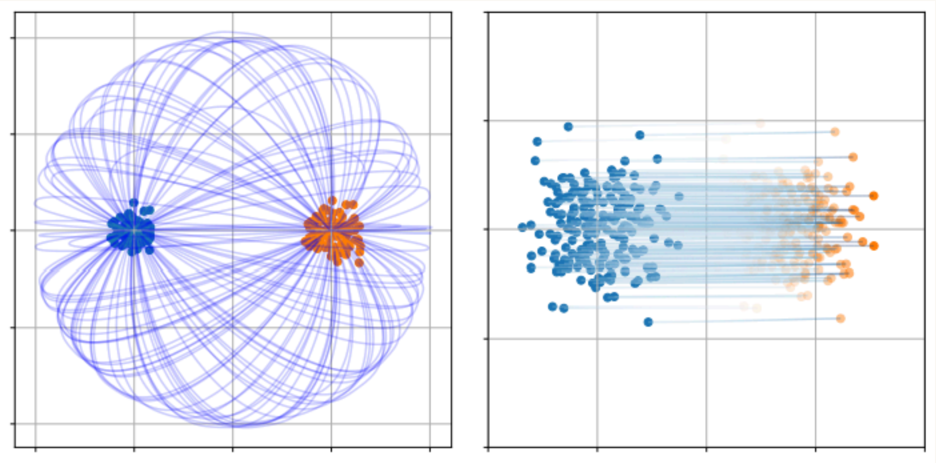
Schiebinger et al.
Cell, 2019



Tong et al.
ICML, 2020



Bunne et al.
Nature Methods, 2023

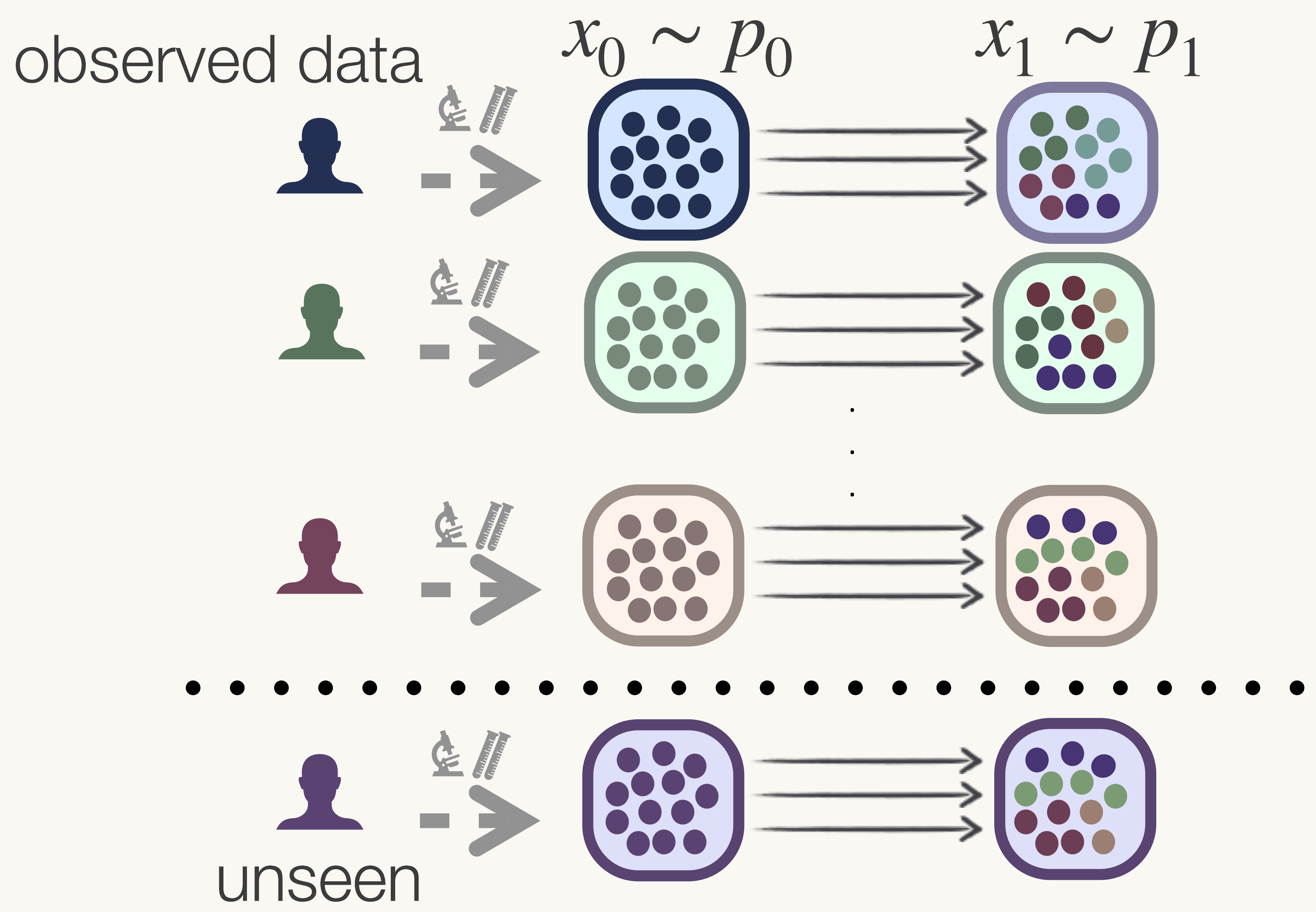


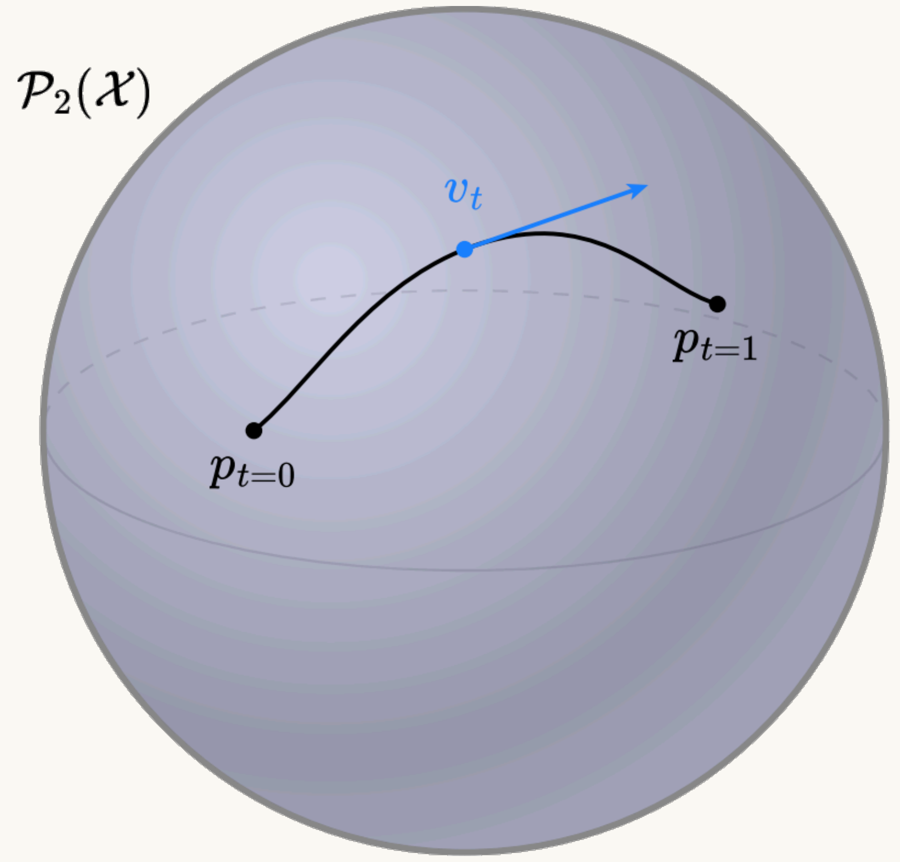
Neklydov et al.
ICML, 2024

They typically assume the evolution of cells are **independent particles**

Background and Motivation

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



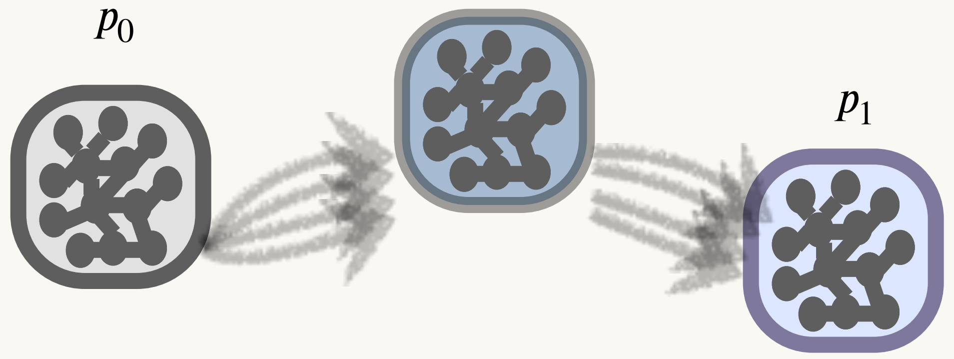


Existing methods are typically restricted to a **single measure** (population, patient)

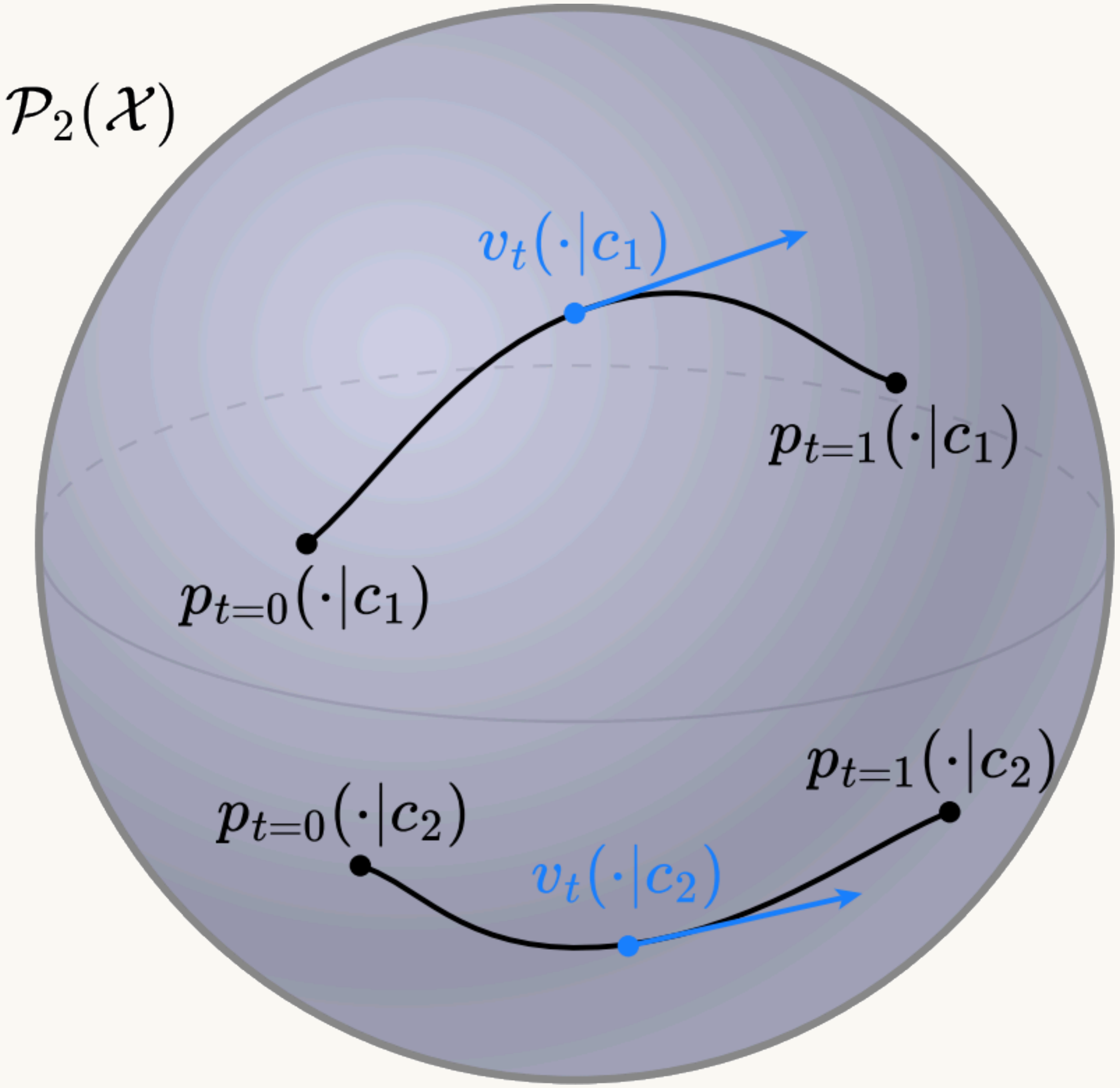
Problem setup

We want a model that can:
1. model the **evolution of particles** while taking into account their **interactions**
2. **generalize** across **unseen populations**

Assumptions.
1. **Coupled** distribution/population pairs $\{(p_0(x_0 \mid i), p_1(x_1 \mid i))\}_{i=1}^{N}$
2. The collected data undergoes a **universal developmental process**
   depends only on the population itself
   e.g., interacting particles or communicating cells



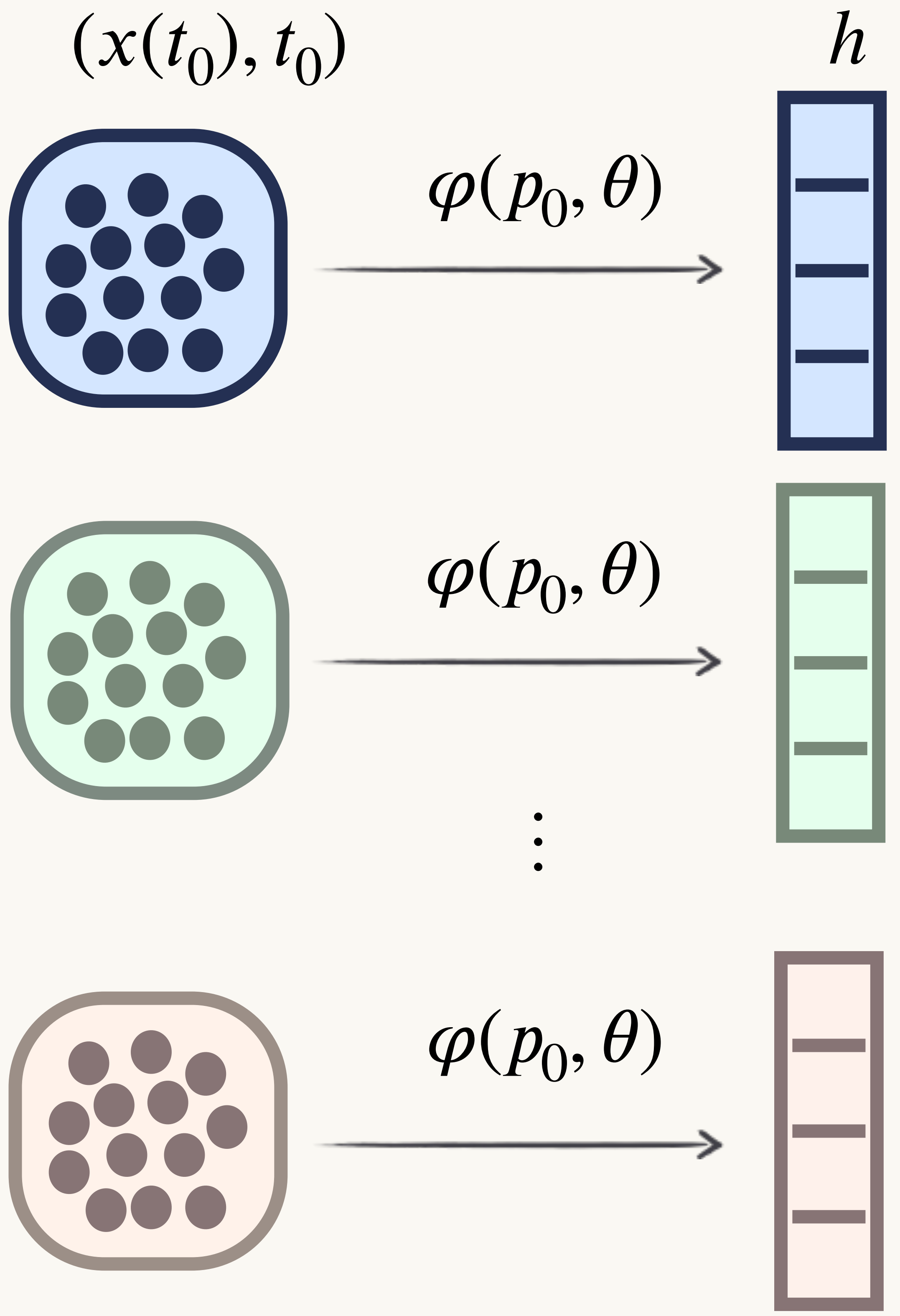
Meta Flow Matching



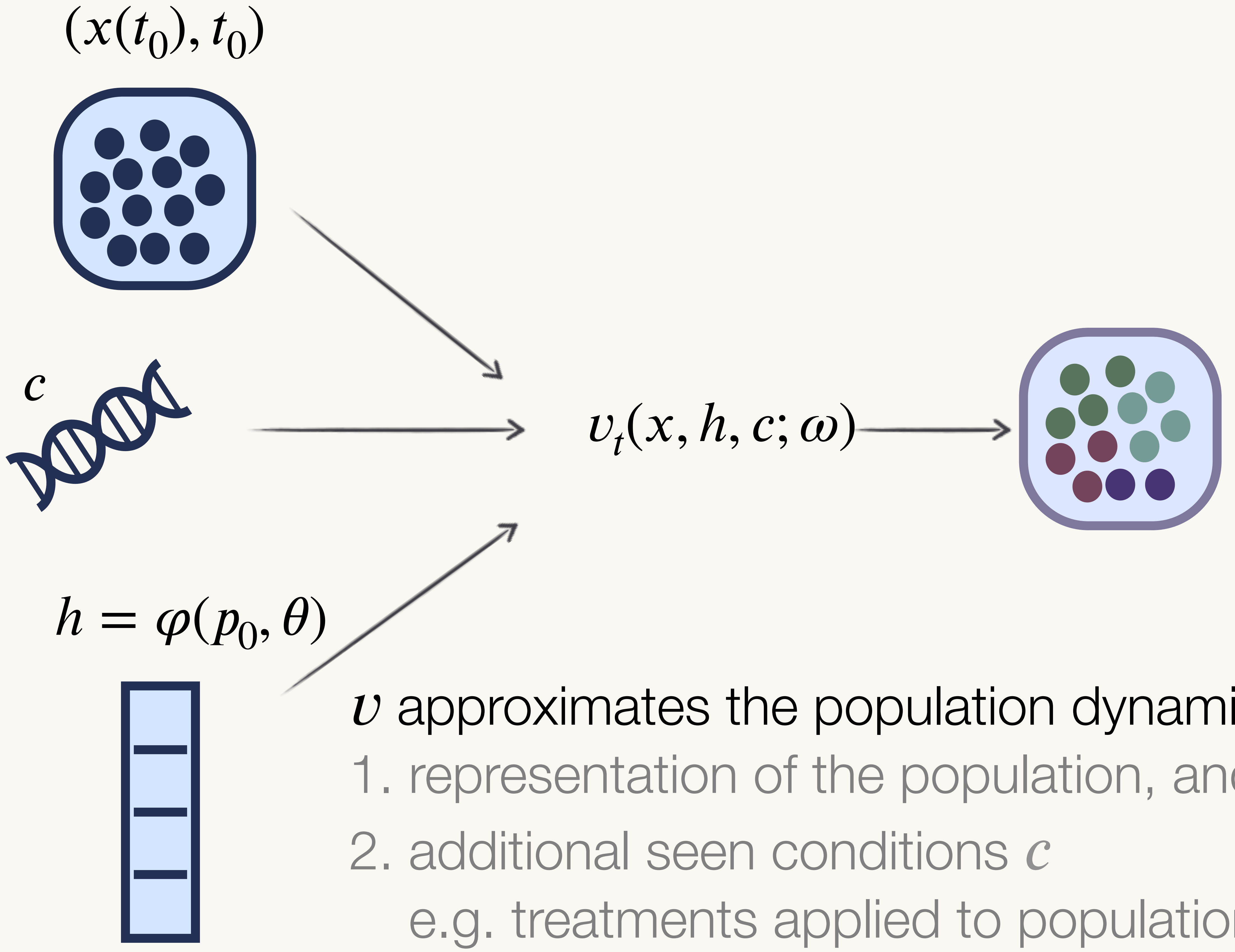
📚 *Meta Flow Matching*. Atanackovic*, Zhang*, Amos, Blanchette, Lee, Bengio, Tong, Neklyudov. ICLR 2025.

Meta Flow Matching

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



$\varphi(p_0, \theta)$ captures particle **interactions**



v approximates the population dynamics given

1. representation of the population, and
2. additional seen conditions c
   e.g. treatments applied to population

Synthetic Example

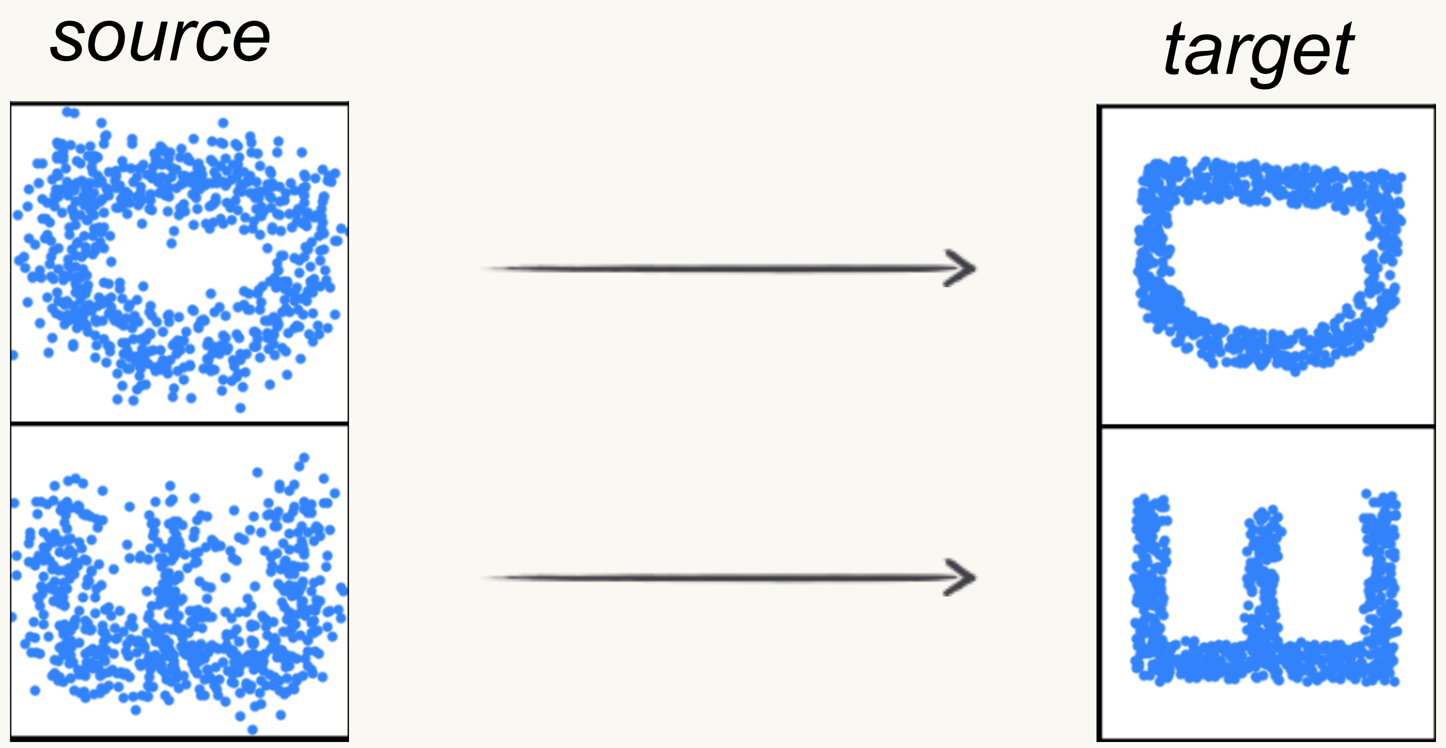
Paired joint distributions $\{(p_0(x_0 \mid i), p_1(x_1 \mid i))\}_{i=1}^{N}$

Target distributions $p_1(x_1 \mid i)$ for $i = 1, \ldots, N$ (letter silhouettes)

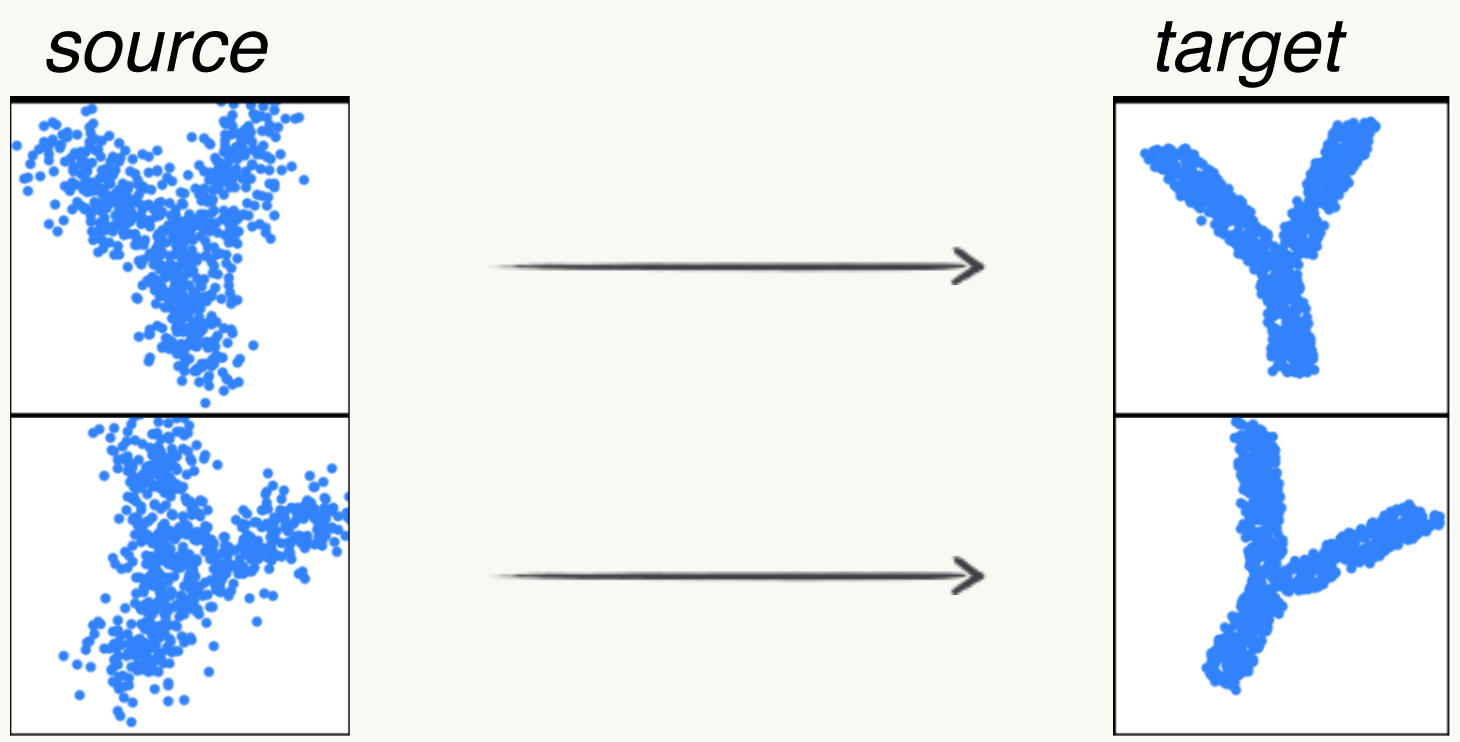
To get $p_0(x_0 \mid i)$ we simulate a **forward diffusion process**

⚠️ not images, but renderings of 2D particle systems

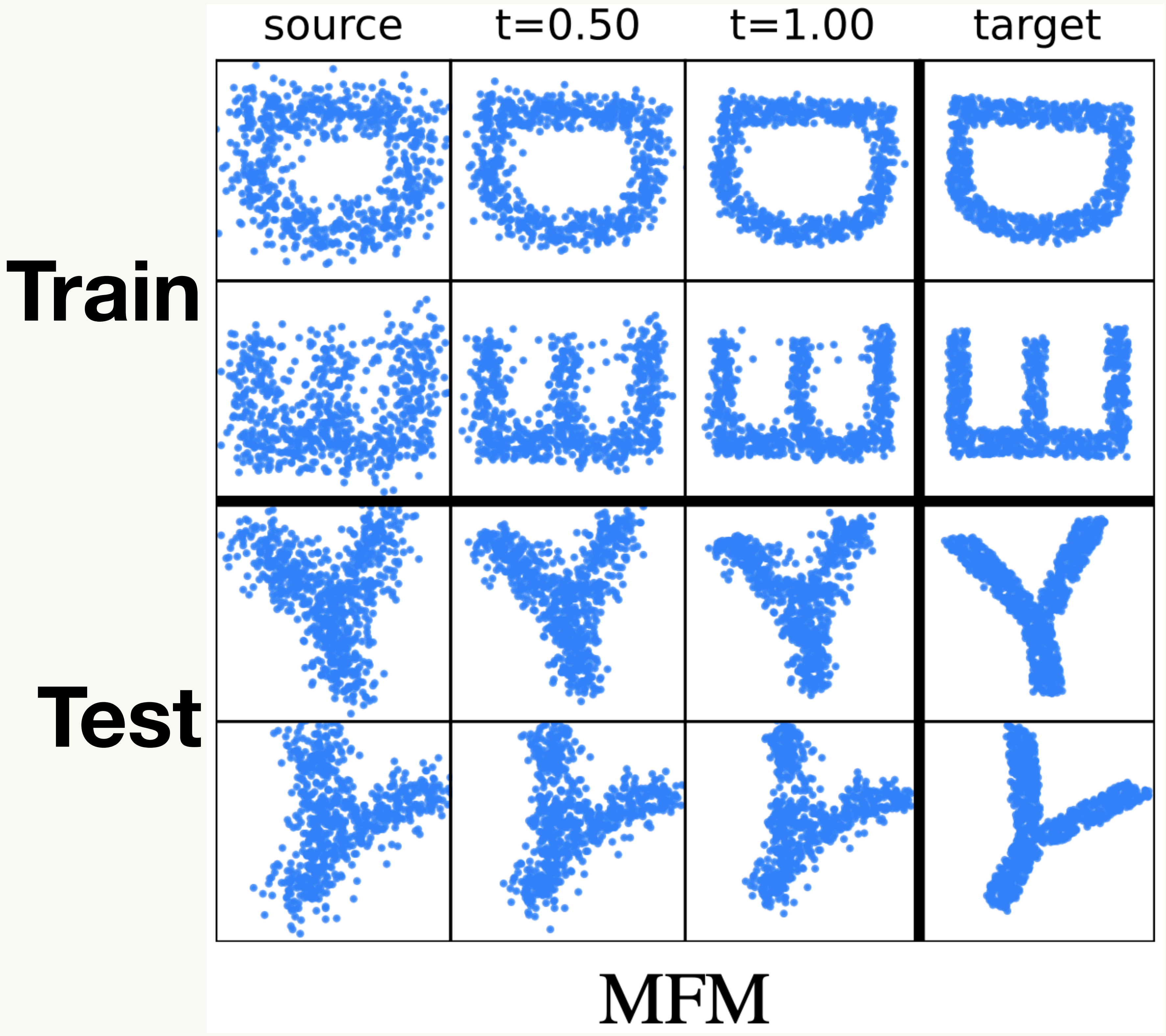
Train: 24 letters (excluding ‘Y’ and ‘X’)



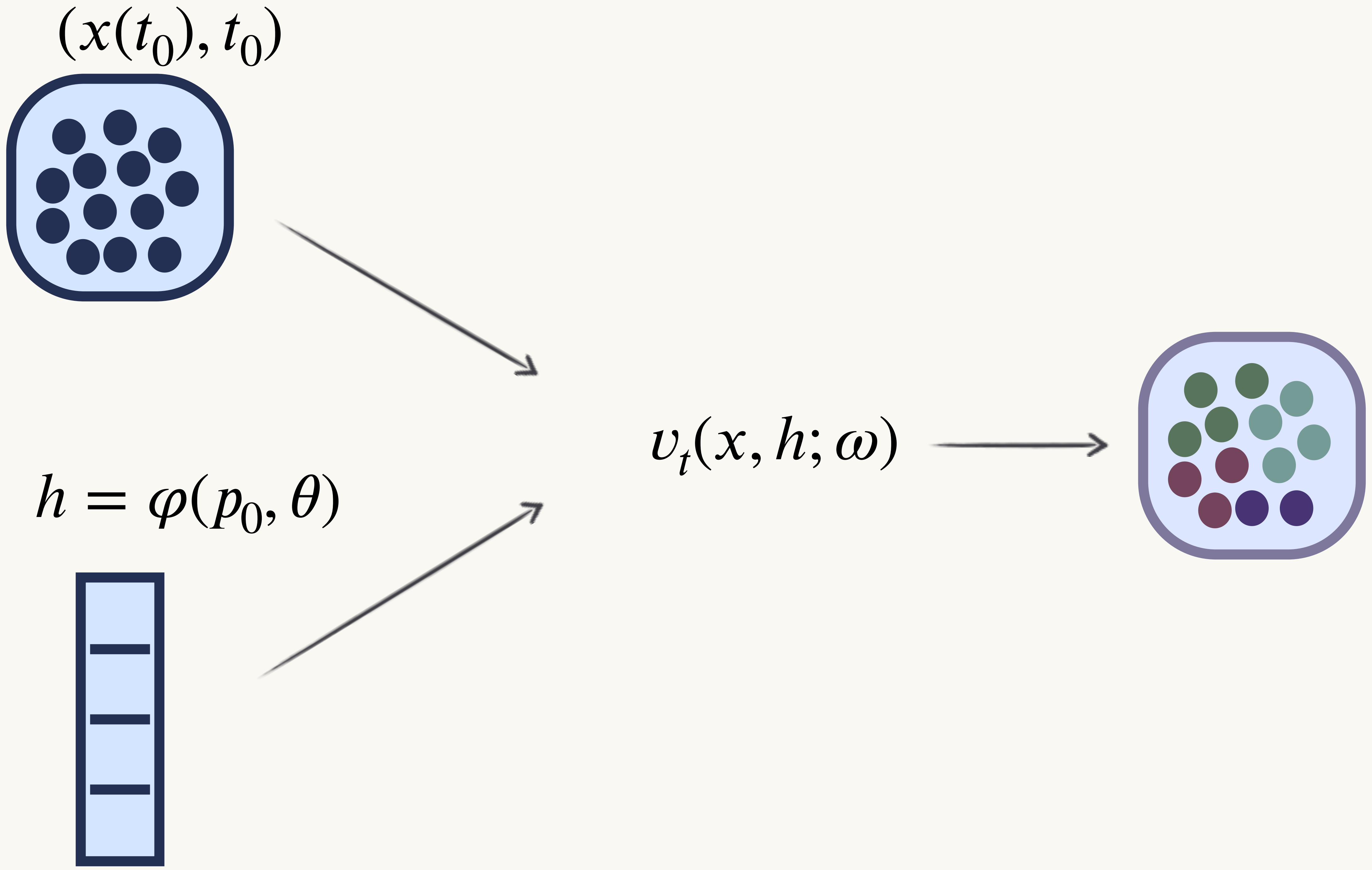
Test: ’Y’ and ‘X’



Synthetic Setting Results



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



📚 *Meta Flow Matching*. Atanackovic*, Zhang*, Amos, Blanchette, Lee, Bengio, Tong, Neklyudov. ICLR 2025.

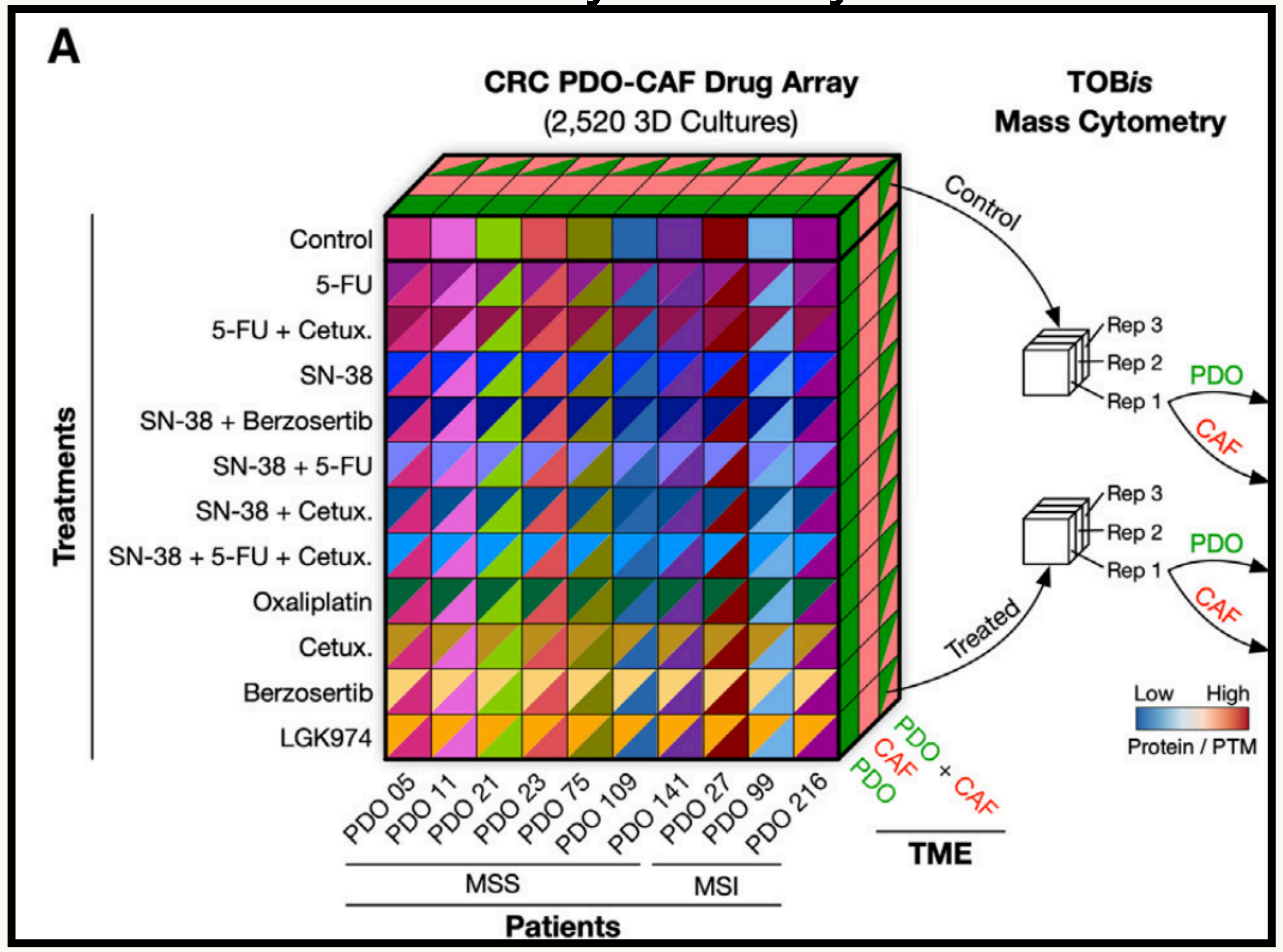
Synthetic Setting Results

	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 ($k = 0$)	$\mathbf{0.081} \pm \mathbf{0.003}$	$\mathbf{0.100} \pm \mathbf{0.004}$	$\mathbf{0.16} \pm \mathbf{0.06}$	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	$\mathbf{0.16} \pm \mathbf{0.01}$	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	$\mathbf{0.201} \pm \mathbf{0.006}$	$\mathbf{0.248} \pm \mathbf{0.006}$	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	$\mathbf{2.18} \pm \mathbf{0.25}$	$\mathbf{0.204} \pm \mathbf{0.005}$	$\mathbf{0.249} \pm \mathbf{0.006}$	$\mathbf{3.14} \pm \mathbf{0.18}$

📚 *Meta Flow Matching*. Atanackovic*, Zhang*, Amos, Blanchette, Lee, Bengio, Tong, Neklyudov. ICLR 2025.

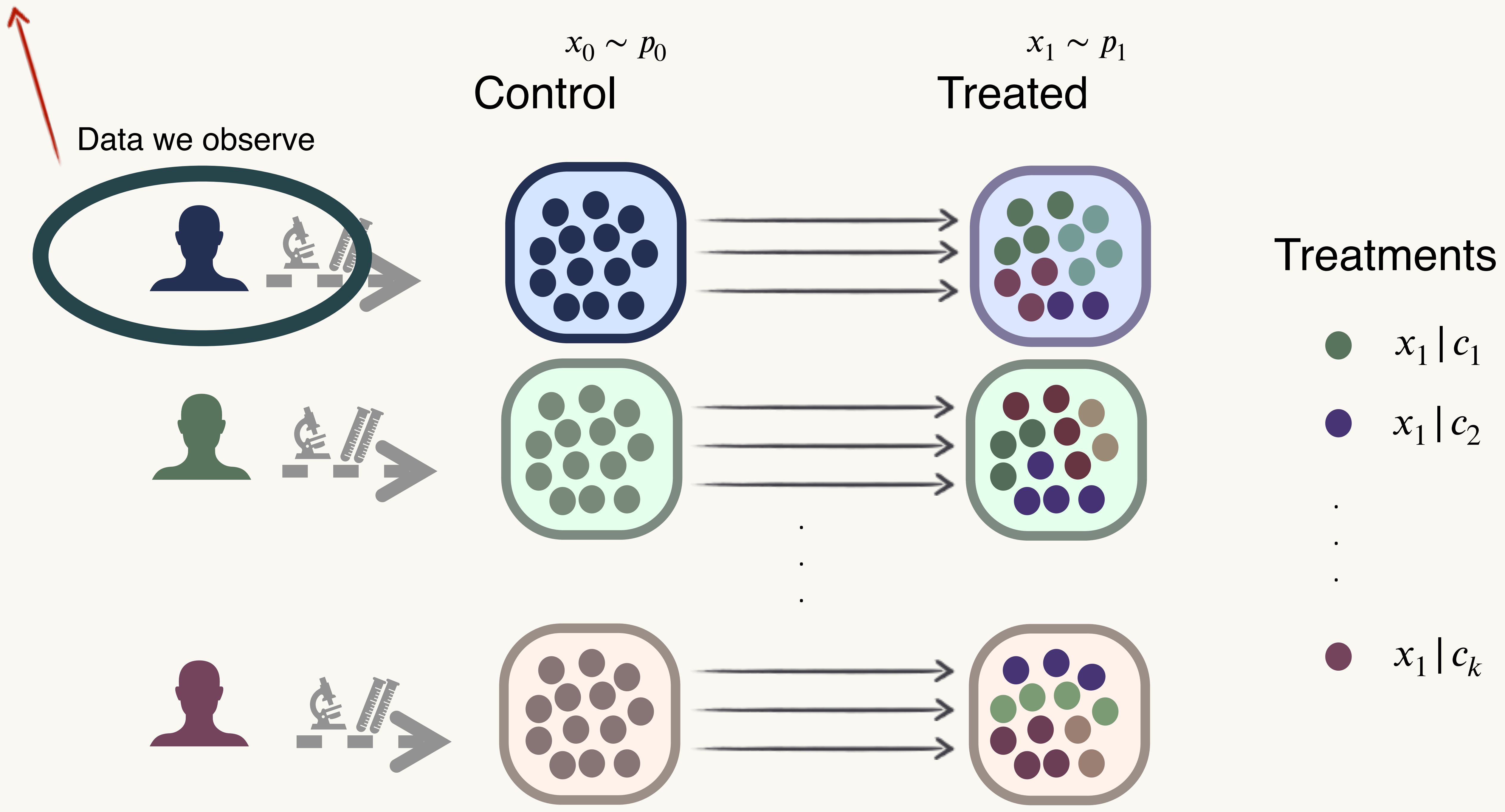
Biological data—patient-specific organoid drug screen dataset

Mass Cytometry



(Zapatero et al, *Cell*, 2023)

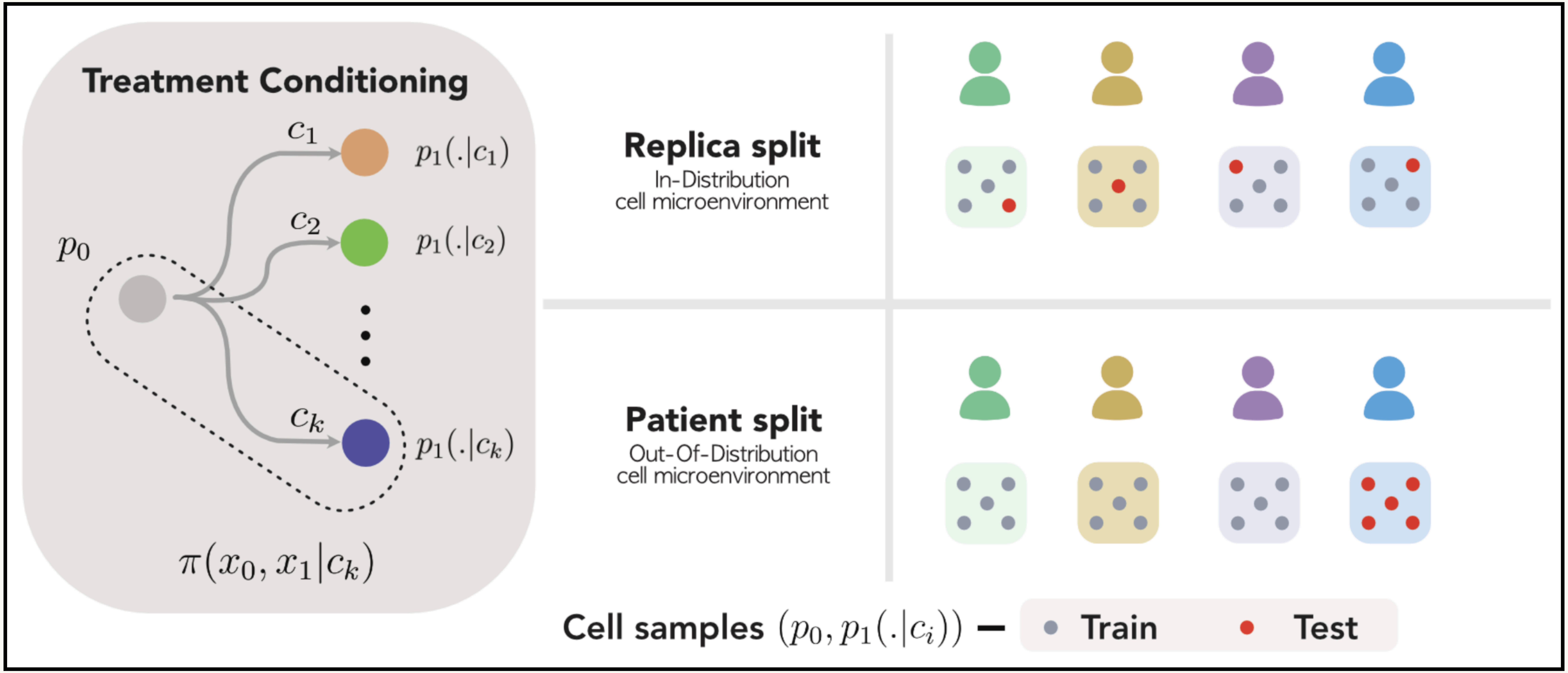
Each patient has ~250 different (control, treated) pairs



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

📚 *Meta Flow Matching*. Atanackovic*, Zhang*, Amos, Blanchette, Lee, Bengio, Tong, Neklyudov. ICLR 2025.

Organoid Drug Screen Data



📚 *Meta Flow Matching*. Atanackovic*, Zhang*, Amos, Blanchette, Lee, Bengio, Tong, Neklyudov. ICLR 2025.

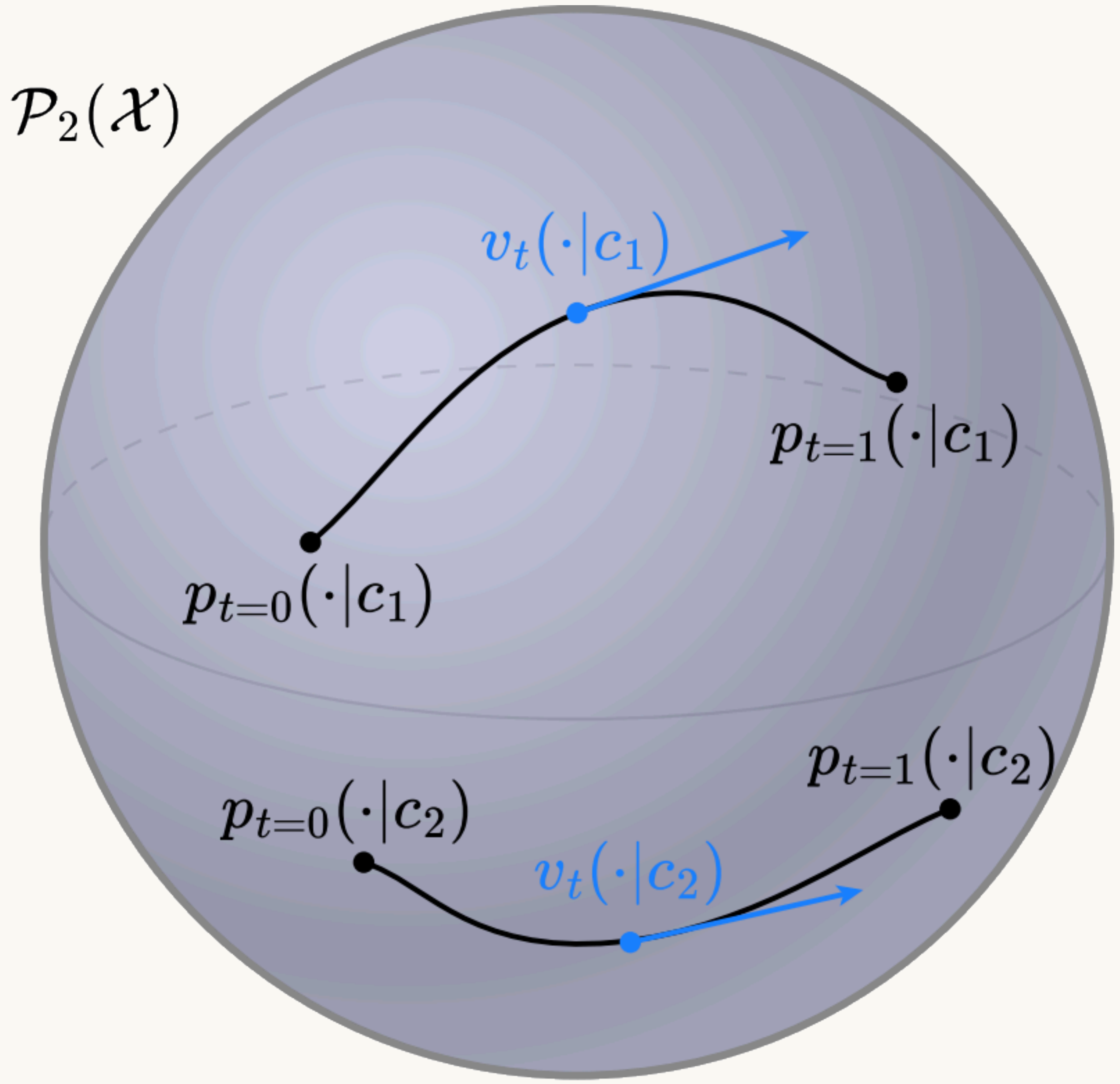
Organoid Drug Screen Results

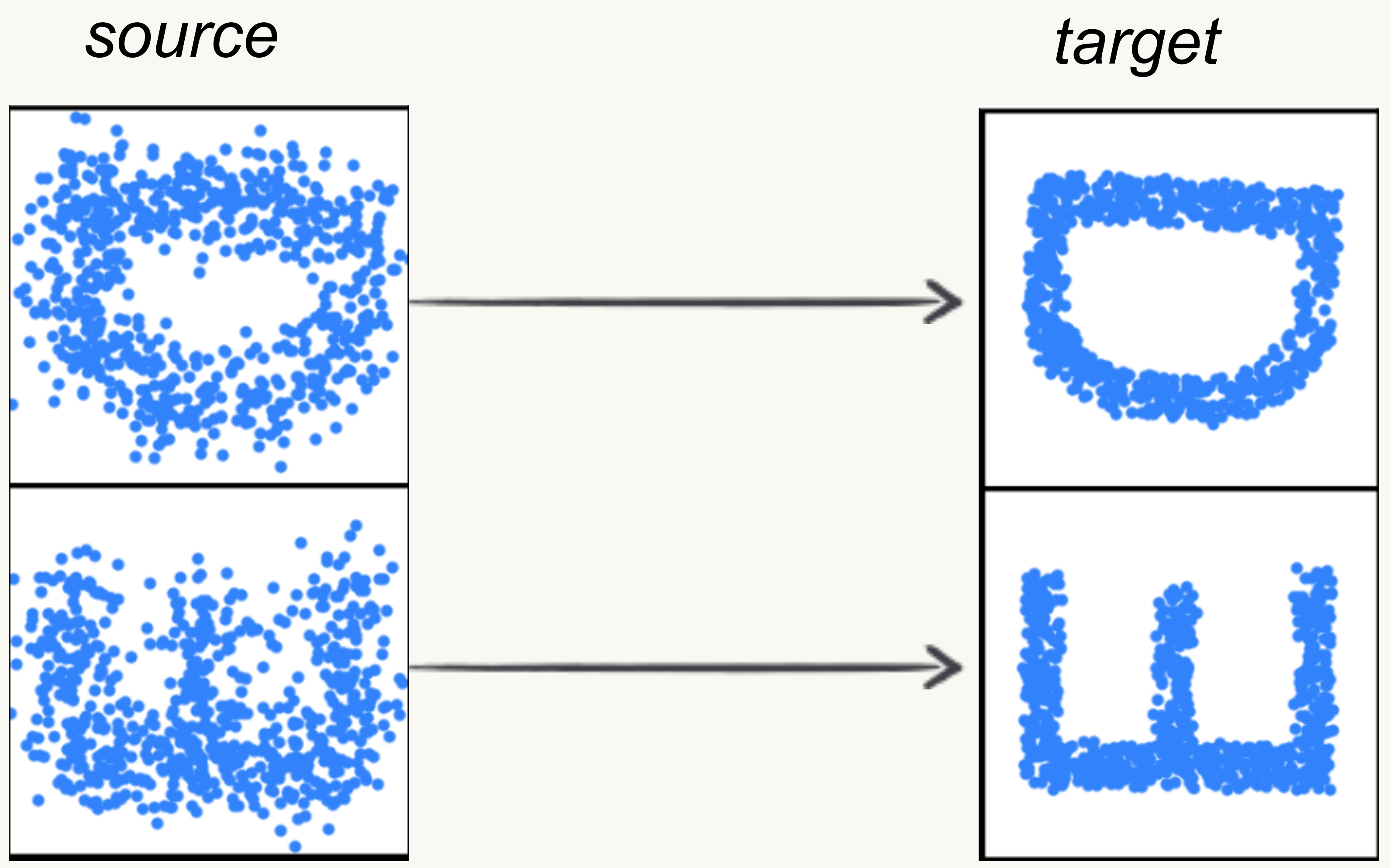
	Train				Test			
	$\mathcal{W}_1(\downarrow)$	$\mathcal{W}_2(\downarrow)$	MMD $(\times 10^{-3})$ $(\downarrow)$	$r^2(\uparrow)$	$\mathcal{W}_1(\downarrow)$	$\mathcal{W}_2(\downarrow)$	MMD $(\times 10^{-3})$ $(\downarrow)$	$r^2(\uparrow)$
FM	3.925 ± 0.019	4.041 ± 0.023	3.76 ± 0.26	0.952 ± 0.007	3.961 ± 0.036	4.089 ± 0.042	5.90 ± 0.25	0.941 ± 0.010
FM$^{\text{w/}}\mathcal{N}$	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$^{\text{w/}}\mathcal{N}$	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	38.6 ± 0.212	0.897 ± 0.031	4.194 ± 0.110	4.313 ± 0.112	37.9 ± 2.84	0.897 ± 0.008
MFM$^{\text{w/}}\mathcal{N}$ $(k = 0)$	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$^{\text{w/}}\mathcal{N}$ $(k = 10)$	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$^{\text{w/}}\mathcal{N}$ $(k = 50)$	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$^{\text{w/}}\mathcal{N}$ $(k = 100)$	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	$\mathbf{0.957 \pm 0.000}$

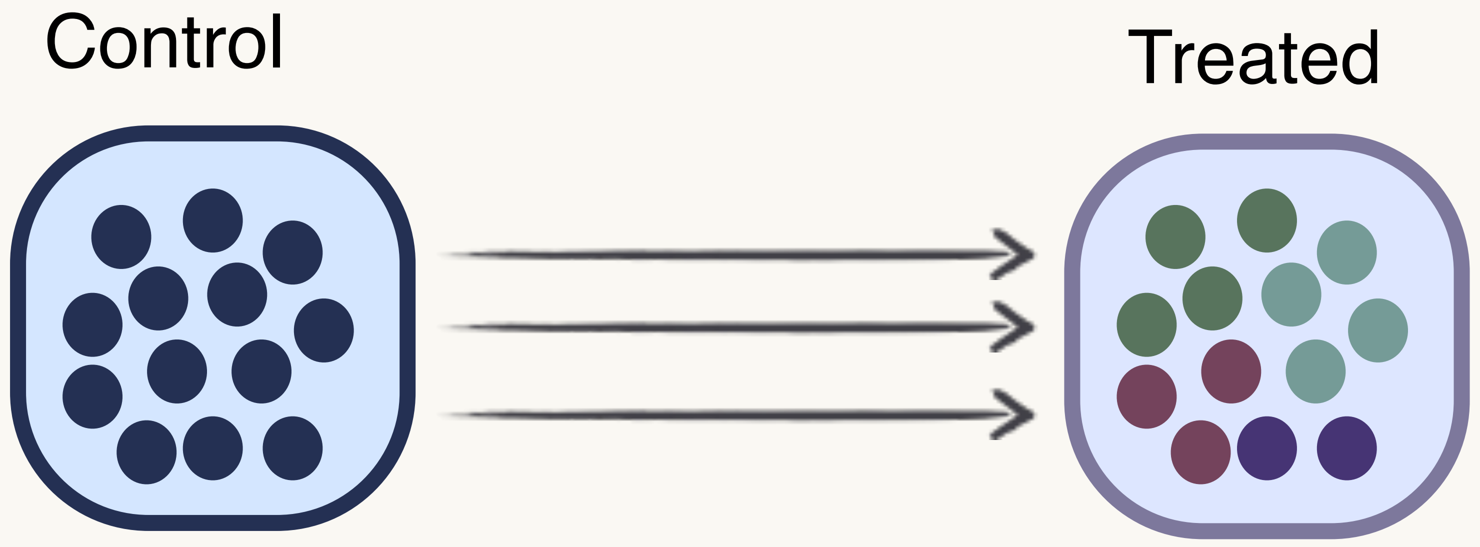
📚 *Meta Flow Matching*. Atanackovic*, Zhang*, Amos, Blanchette, Lee, Bengio, Tong, Neklyudov. ICLR 2025.

So far: pairs of distributions

Meta FM assumes **coupled** pairs of distributions

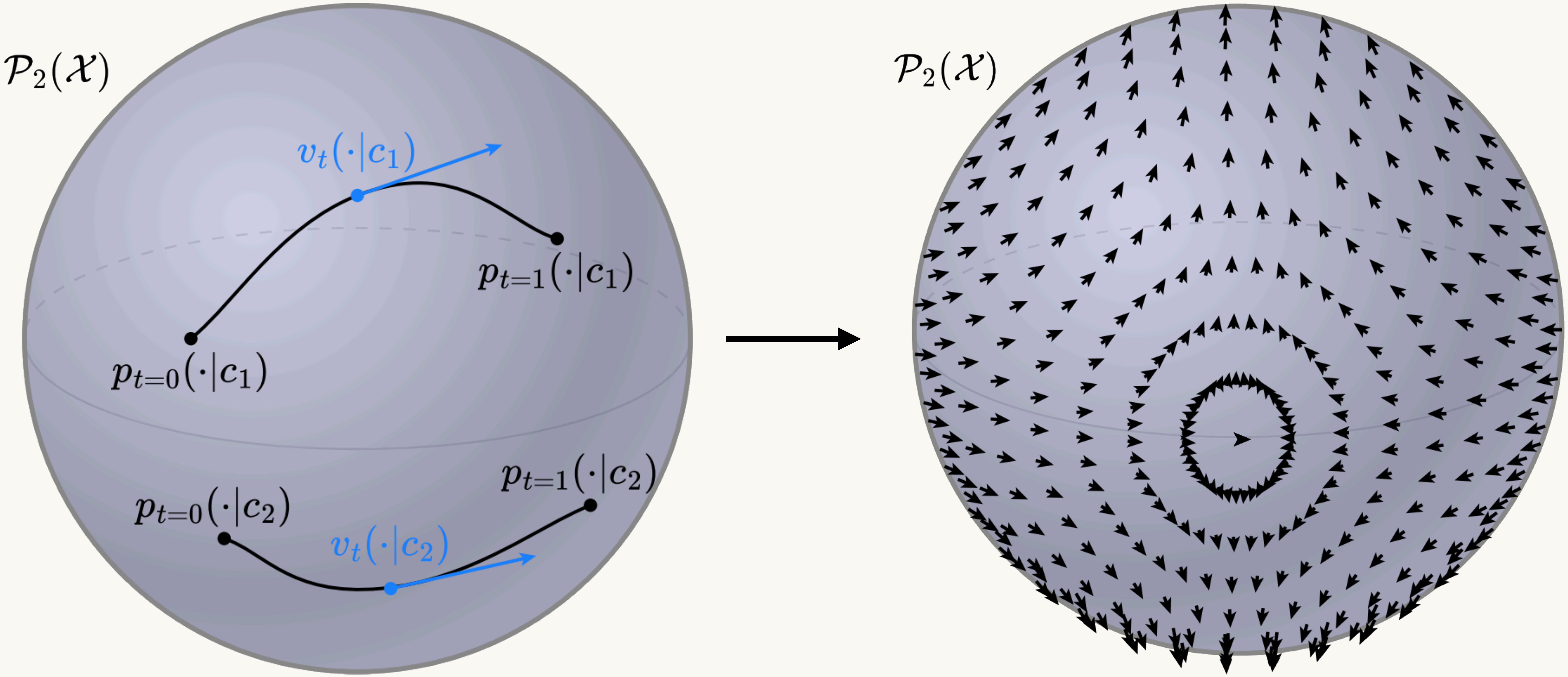






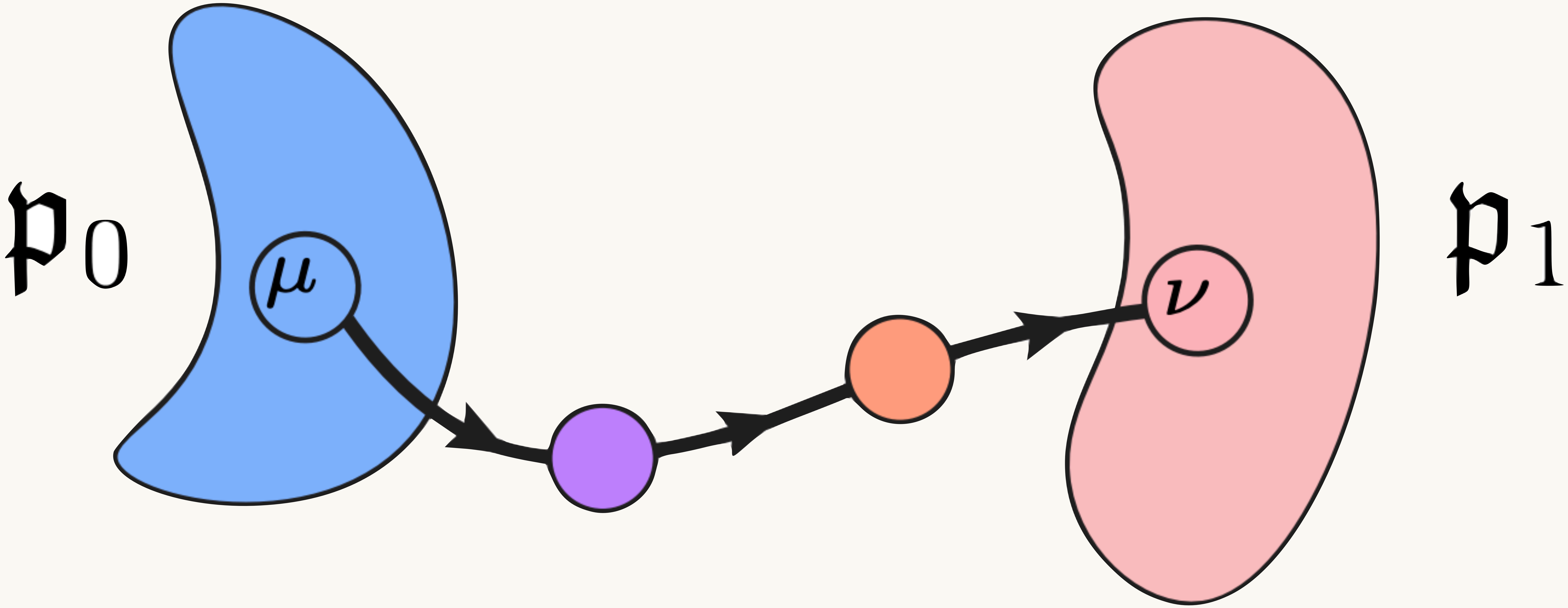
Can we learn a flow *on* the Wasserstein manifold? Yes!

📚 Wasserstein Flow Matching. Haviv, Pooladian, Pe'er, Amos. NeurIPS 2025.



Transport, but each point is a distribution

📚 Wasserstein Flow Matching. Haviv, Pooladian, Pe’er, Amos. NeurIPS 2025.



Ok transport ❤️ physics, now what?

The usual ones: **scale, applications, extensions**

For example, drug discovery, other population and particle dynamics

Ok transport ❤️ physics, now what?

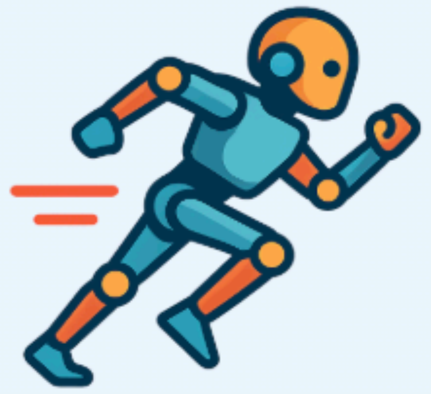
The *other* usual one: **AGI**

1. Integrate **broader knowledge** and **information** into the transports
2. Language models can't solve transport problems (at least not yet…)

Ok transport ❤️ physics, now what?

The *other* usual one: **AGI**

1. Integrate **broader knowledge** and **information** into the transports
2. Language models can’t solve transport problems (at least not yet…)



AlgoTune

Can Language Models Speed Up General-Purpose Numerical Programs?

NeurIPS D&B 2025

Ori Press Brandon Amos Haoyu Zhao Yikai Wu Samuel K. Ainsworth Dominik Krupke Patrick Kidger
Touqir Sajed Bartolomeo Stellato Jisun Park Nathanael Bosch Eli Meril Albert Steppi
Arman Zharmagambetov Fangzhao Zhang David Pérez-Piñeiro Alberto Mercurio Ni Zhan
Talor Abramovich Kilian Lieret Hanlin Zhang Shirley Huang Matthias Bethge Ofir Press

UNIVERSITÄT TÜBINGEN

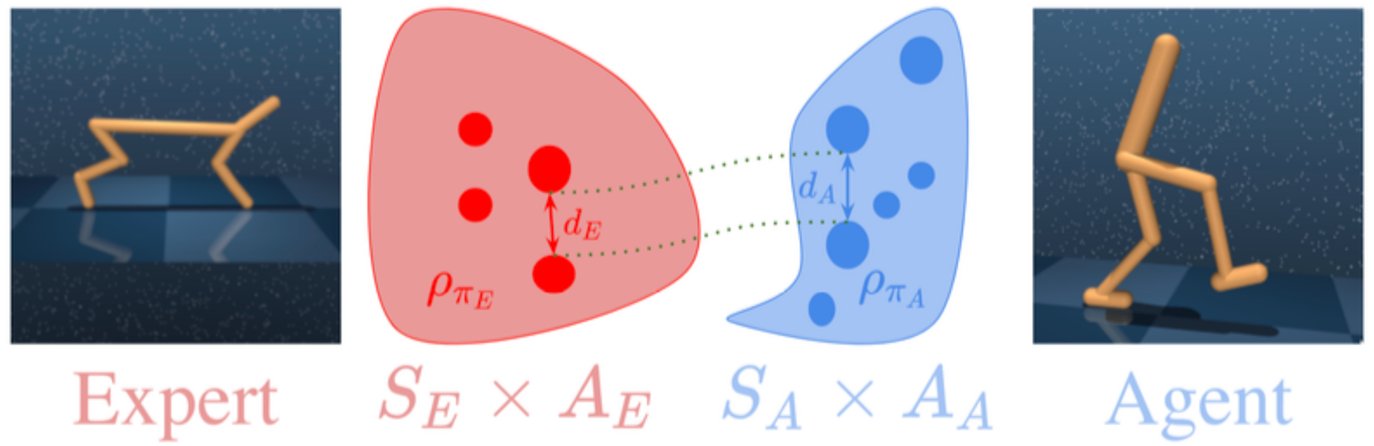
PRINCETON UNIVERSITY

On transport, flows, and physics

Brandon Amos

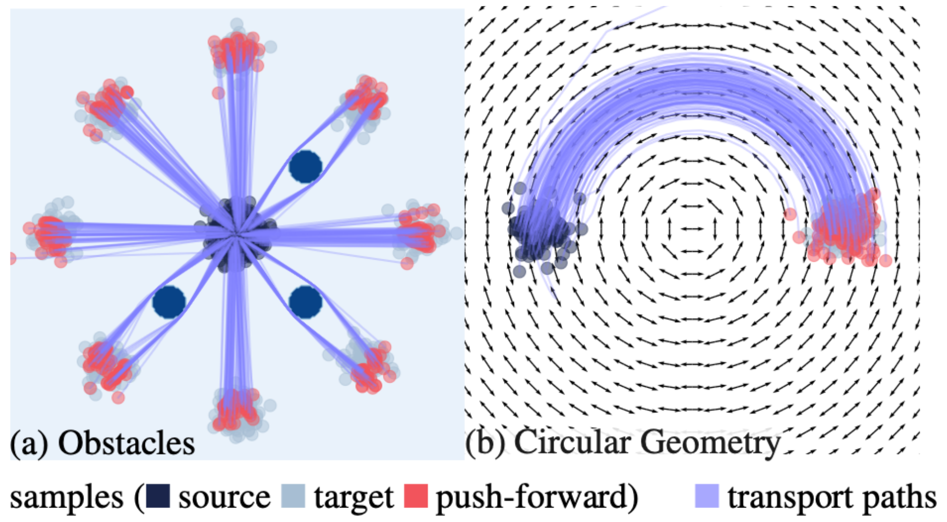
`bamos.github.io/presentations`

Transport for Imitation Learning



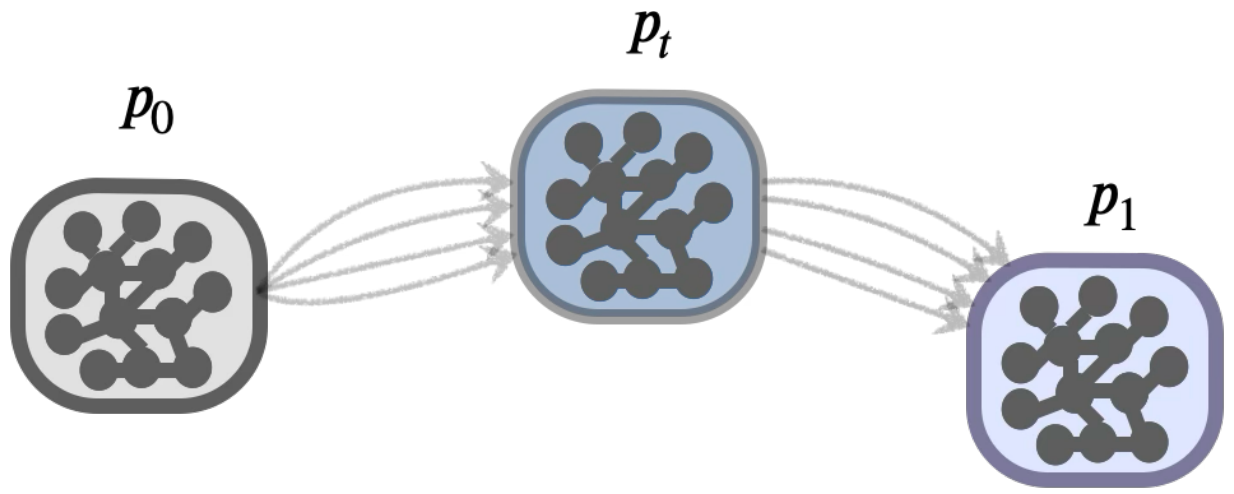
📚 Gromov-Wasserstein Imitation

Non-Euclidean Transport



📚 Lagrangian Optimal Transport

Meta and Wasserstein Flows



📚 Meta Flow Matching

📚 Wasserstein Flow Matching

In collaboration with Samuel Cohen, Arnaud Fickinger, Stuart Russell, Aram-Alexandre Pooladian, Doron Haviv, Dana Pe'er, Carles Domingo-Enrich, Ricky Chen, Lazar Atanackovic, Xi Zhang, Mathieu Blanchette, Leo J. Lee, Yoshua Bengio, Alexander Tong, Kirill Neklyudov