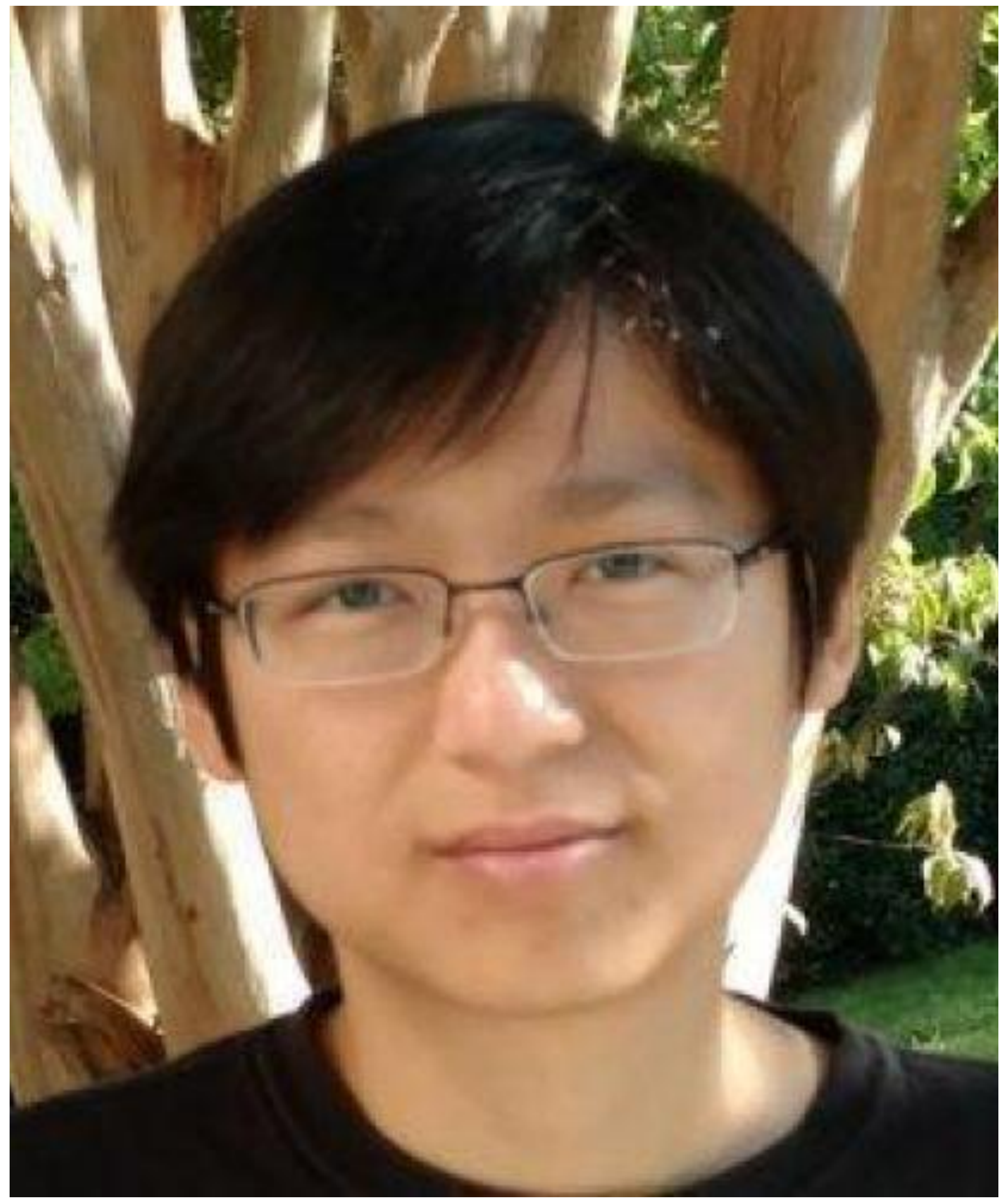
Computing high-dimensional optimal transport by flow neural networks

Yao Xie
Georgia Institute of Technology

Women in OT Workshop, UBC, Vancouver
April 19, 2024



Chen Xu
GT

Xiuyuan Cheng
Duke

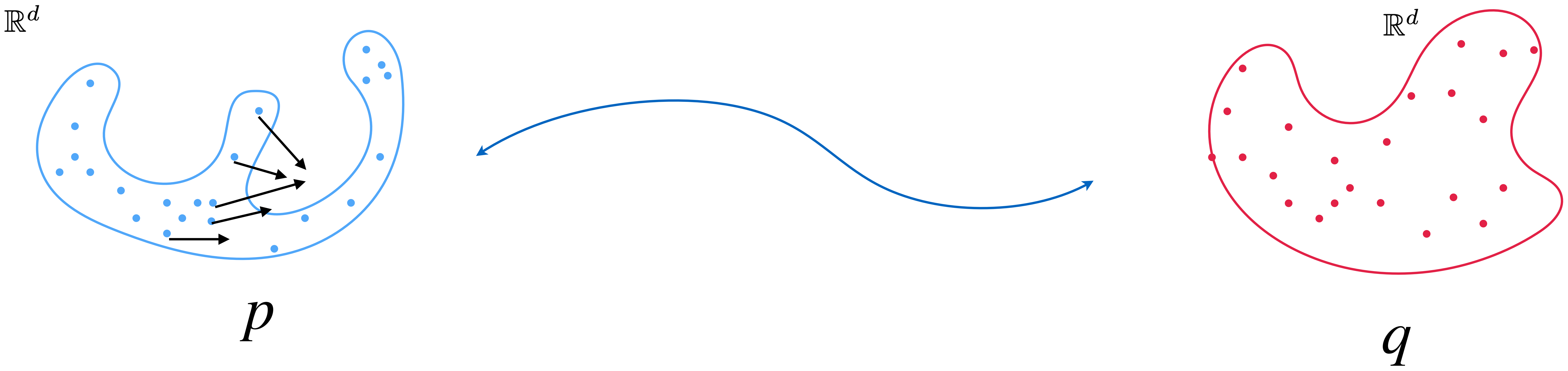
Computing high-dimensional optimal transport by flow neural networks. Cheng, **X.** arXiv:2305.11857. 2024

Roadmap

- Problem set-up
- Background
- Proposed algorithm
- Numerical examples
- Application: Improved density ratio estimation (DRE)

Estimate high-dimensional optimal transport

- Given two sets of d-dimensional samples $\{X_i\}_{i=1}^N \sim p$ and $\{\tilde{X}_j\}_{j=1}^M \sim q$
- Goal: (i) estimate $\mathscr{W}_2^2(p, q)$ and (ii) find **transport map** to match distributions



Idea: Use dynamic optimal transport formula with neural ODE, to handle high-dim data.

Between two arbitrary distributions

- Motivation: optimal transport, transfer learning, domain adaptation

$x_i \sim \rho_t$

$\rho_0 = p$

Handbag

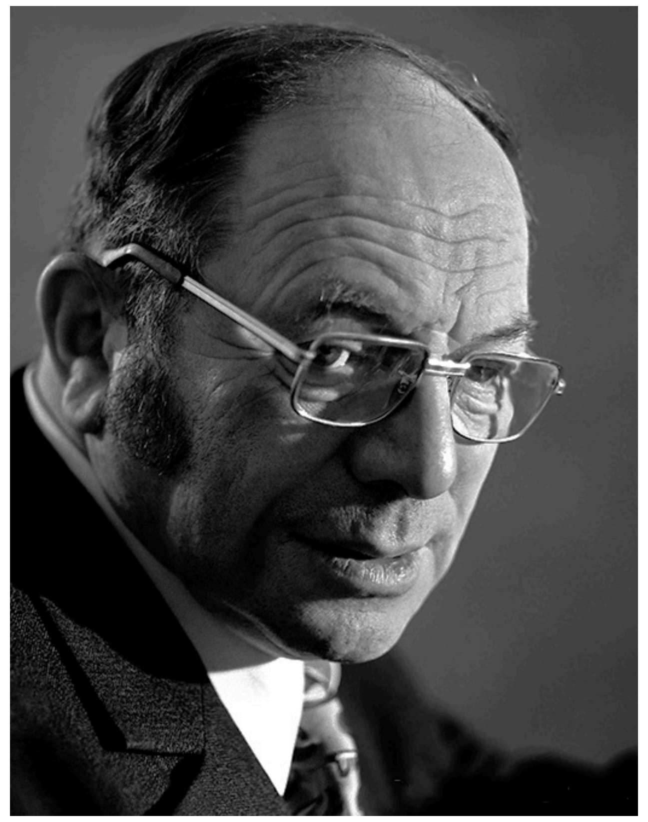
$\rho_1 = q$

Shoes

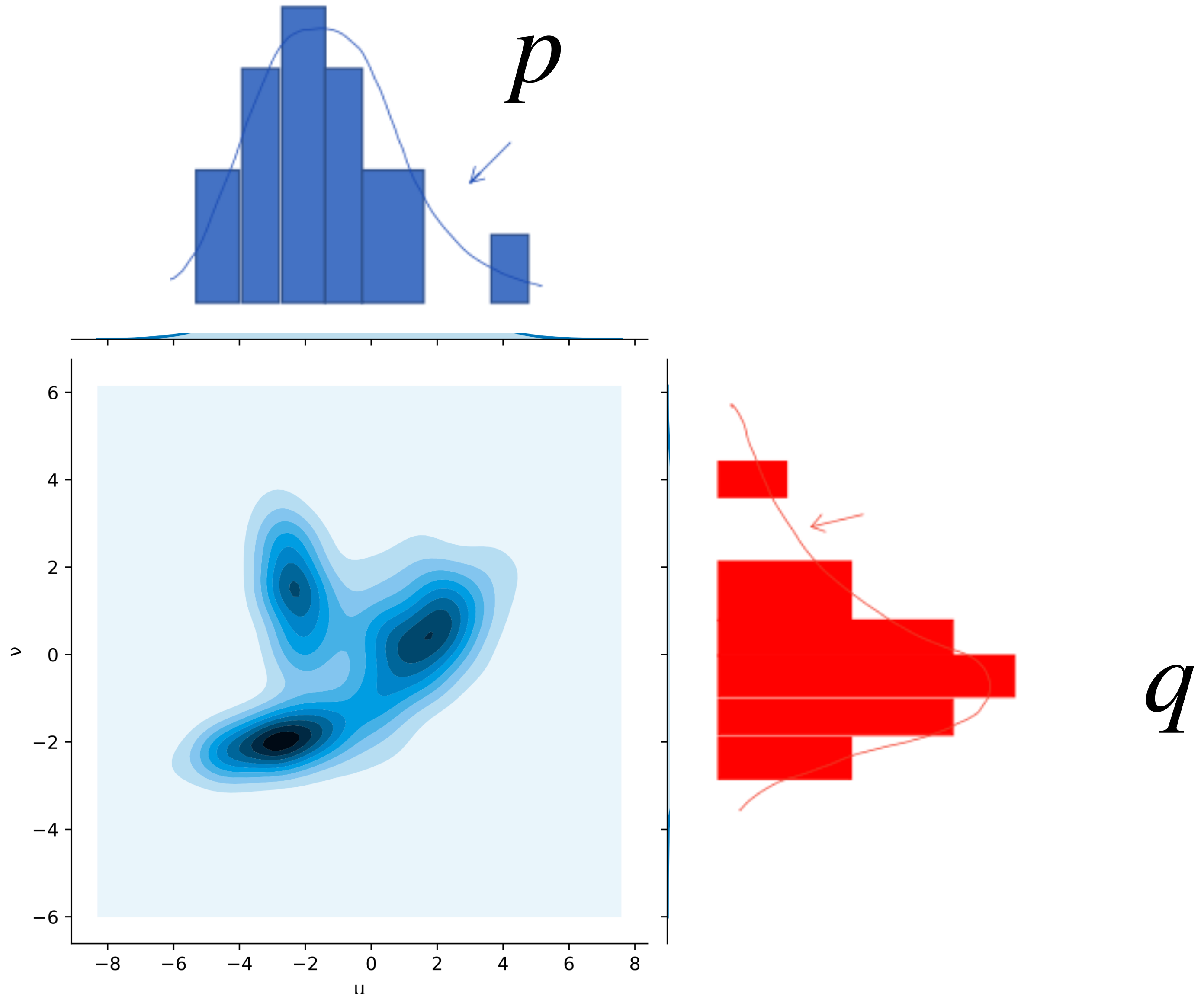
Wasserstein metric

- Distance function defined between *probability distributions* on a metric space: minimum cost of transporting probabilities
- Wasserstein-2 metric, Kantorovich

$$\mathscr{W}_2^2(p,q) = \min_{\gamma}\{\mathbb{E}_{(X,X')\sim\gamma} | \|X - X'\|_2^2 : \gamma \text{ has marginal distribution } p, q\}$$



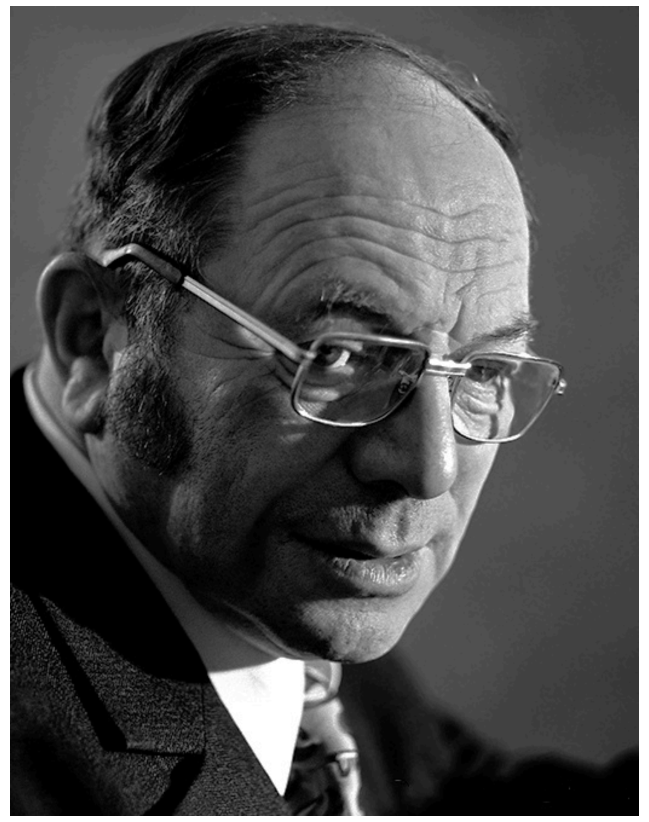
Kantorovich (1930)



Wasserstein metric

- Distance function defined between probability distributions on a metric space: minimum cost of transporting probabilities
- Wasserstein-2 metric, Kantorovich

$$\mathscr{W}_2^2(p,q) = \min_{\gamma}\{\mathbb{E}_{(X,X')\sim\gamma}\|\|X - X'\|^2 : \gamma \text{ has marginal distribution } p, q\}$$



Kantorovich (1930)

- Monge: Pushforward operator (**transport map**) $T : \mathbb{R}^d \to \mathbb{R}^d$:

$$T_\sharp P(A) = P(T^{-1}(A))$$

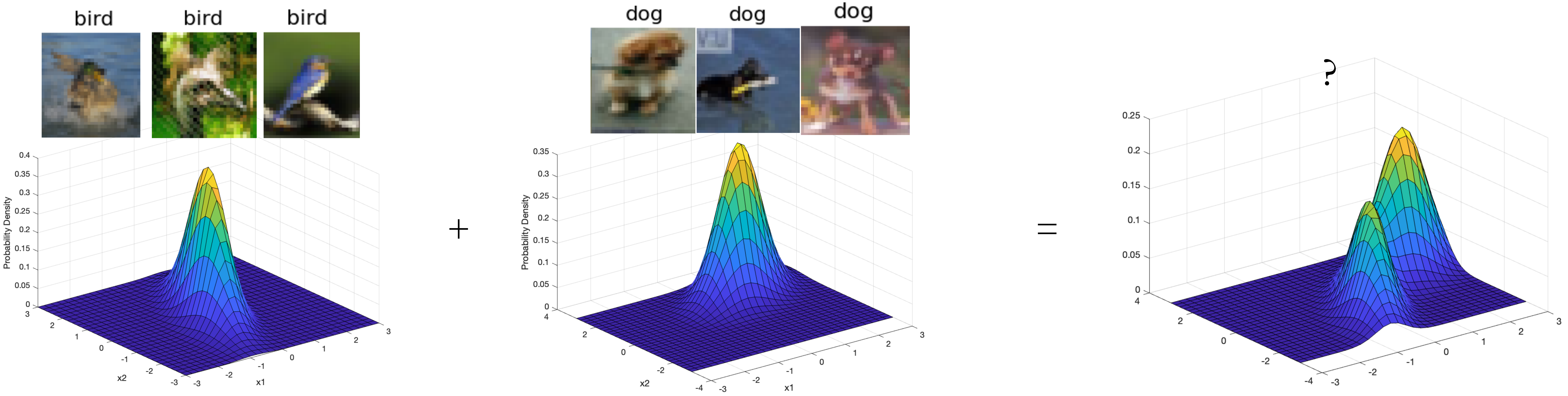
$$\mathscr{W}_2^2(p,q) = \min_{T:T_\sharp p = q} \mathbb{E}_{X\sim p}\|X - T(X)\|_2^2$$

- Brenier Theorem (1991) Monge = Kantorovich under regularity cond.

Monge (1781)

Space of distributions

- We are familiar with “vector spaces” but “distribution space” is tricky
- $p_1 + p_2$ is not a distribution
- $0.4p_1 + 0.6p_2$ is a distribution but we cannot do this to convert “noise” to “cat”



Dynamic view of density evolution

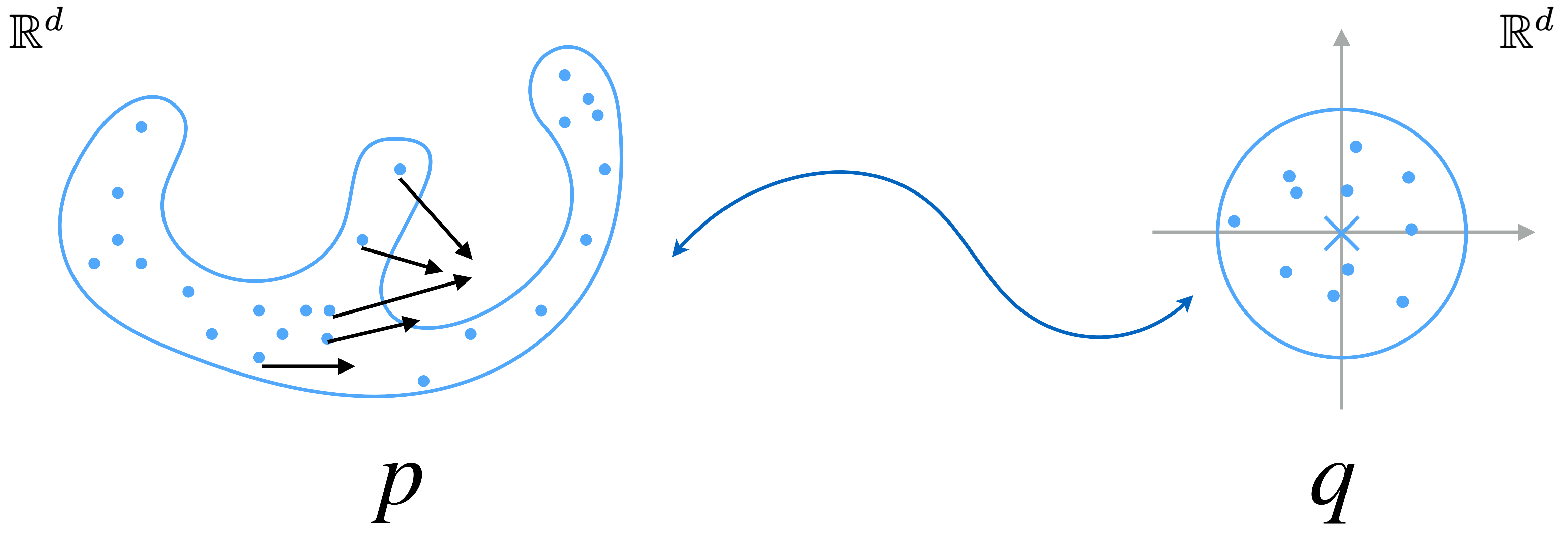
- Particles $X(0) \sim p$, push particles by velocity field $v(\cdot, t) : \mathbb{R}^d \to \mathbb{R}^d$

$$\dot{x}(t) = v(x(t), t)$$

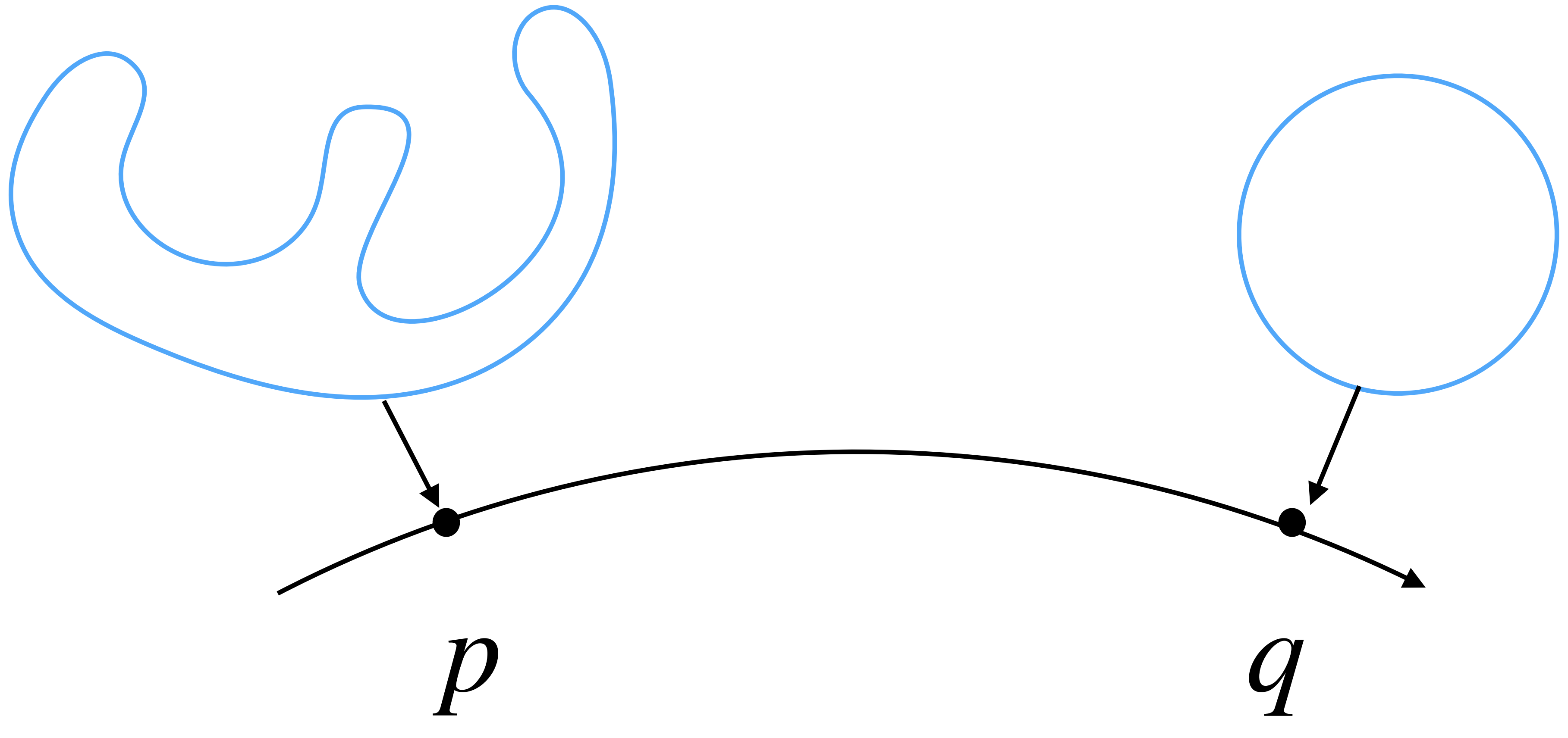
- Distributions $X(t) \sim \rho_t$

$$\partial_t \rho_t + \nabla \cdot (\rho_t v_t) = 0$$

continuity equation



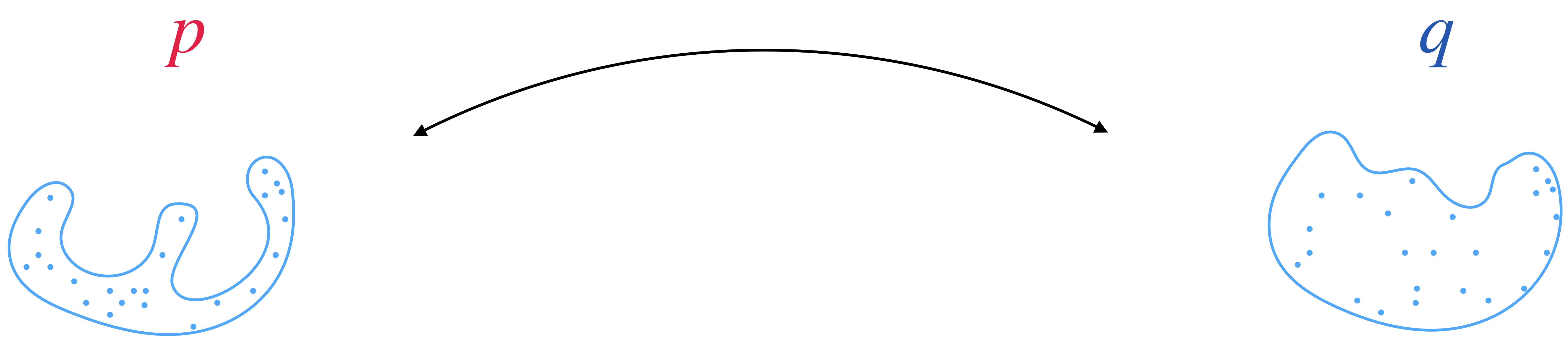
particle space



distribution space

Space of distributions

- More interestingly …



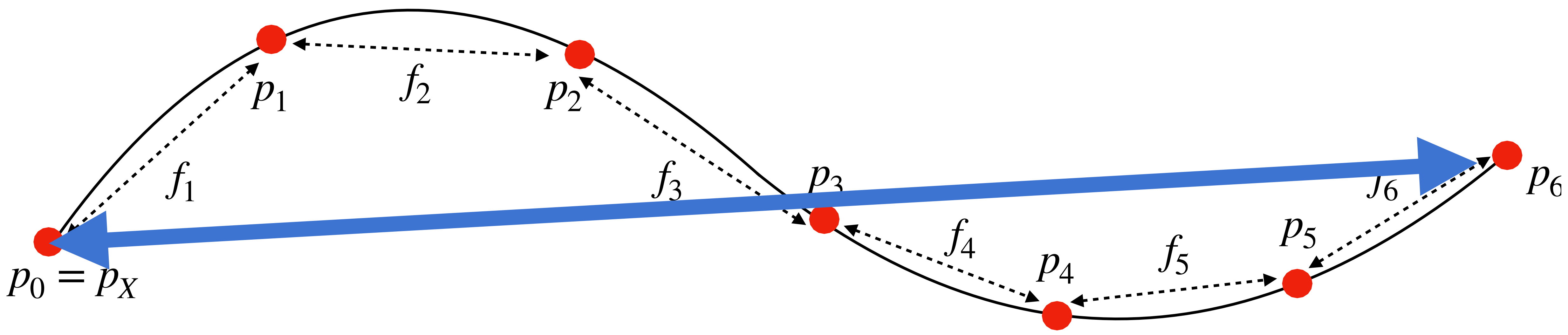
Dynamic formulation of Wasserstein

- Benamou-Brenier formula (2000) (Villani et al. 2009)
- **Optimal** velocity field leads to

$$\mathcal{W}_2^2(p, q) := \inf_{\rho, v} \int_0^1 \mathbb{E}_{x \sim \rho(\cdot, t)} \|v(x,t)\|^2 dt$$

$$s.t. \quad \partial_t \rho + \nabla \cdot (\rho v) = 0, \quad \rho(\cdot, 0) = p, \quad \rho(\cdot, 1) = q$$

$v(x, t)$

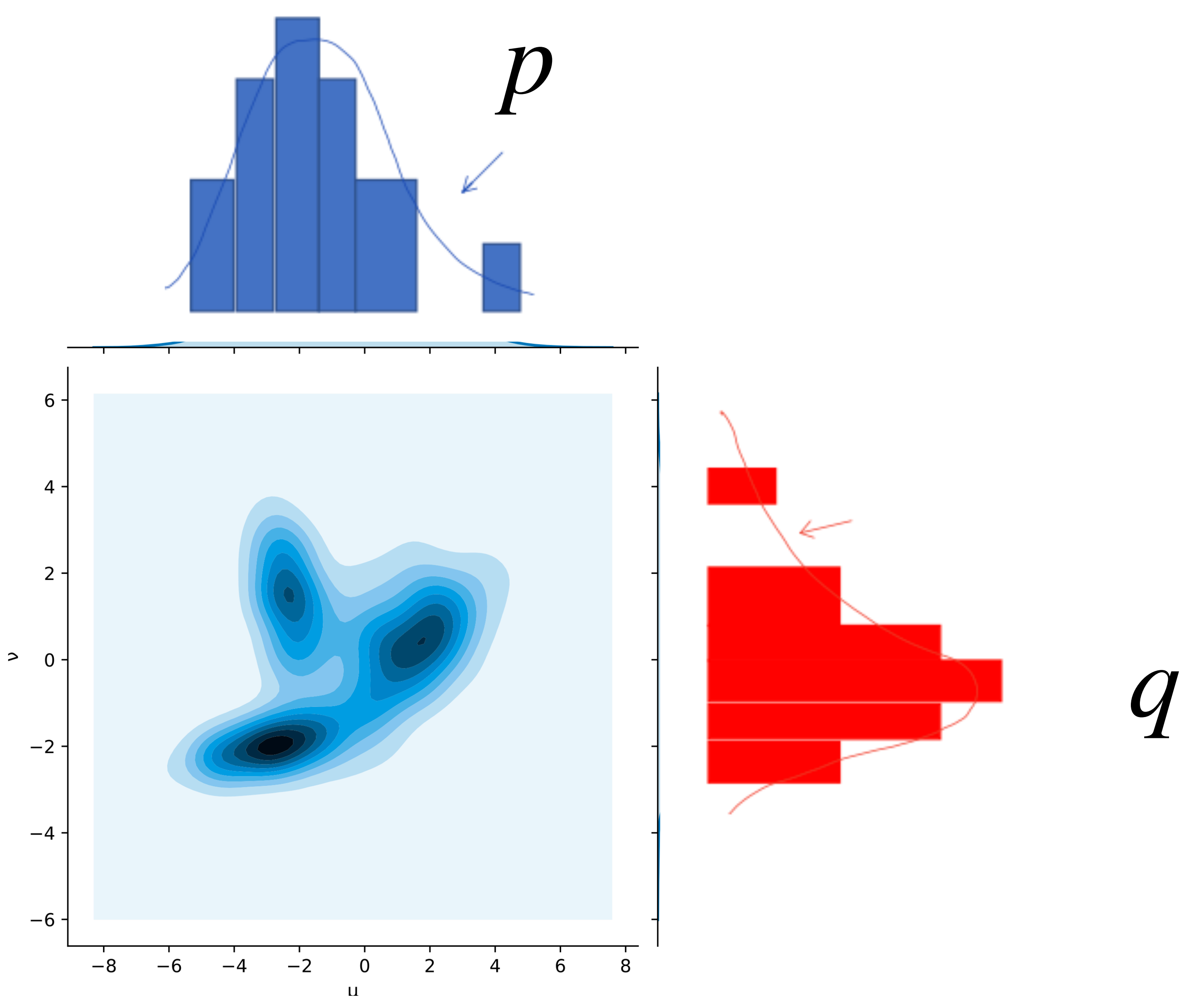


- Transport map: $T_0^t(x) = x + \int_0^t v^\star(x(s), s) ds,$ and $x(s) = T_0^s(x)$

Dynamic vs. Static Wasserstein

- Static: “one shot”

$$\mathscr{W}_2^2(p,q) = \min_{T:T_\sharp p=q} \mathbb{E}_{X\sim p}\|X - T(X)\|_2^2$$

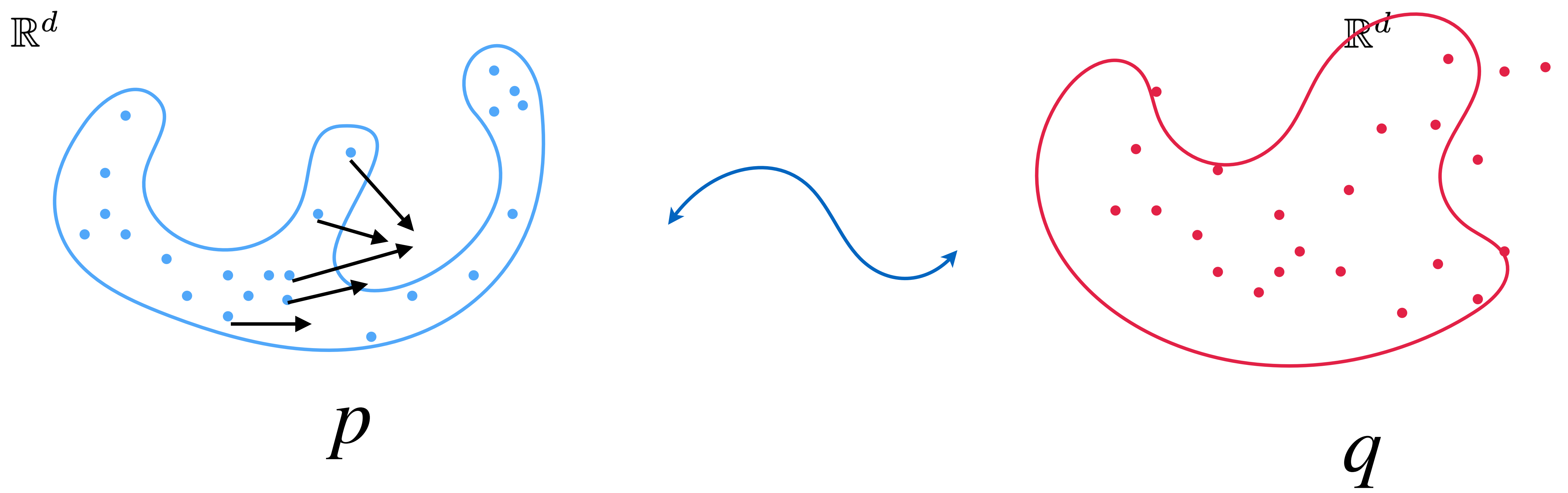


- Dynamic: trajectory

$$\mathscr{W}_2^2(p,q) :=$$

$$\inf_{\rho,v} \int_0^1 \mathbb{E}_{x\sim\rho(\cdot,t)}\|v(x,t)\|^2 dt$$

$$s.t. \quad \partial_t \rho + \nabla\cdot(\rho v) = 0, \quad \rho(\cdot,0) = p, \quad \rho(\cdot,1) = q$$



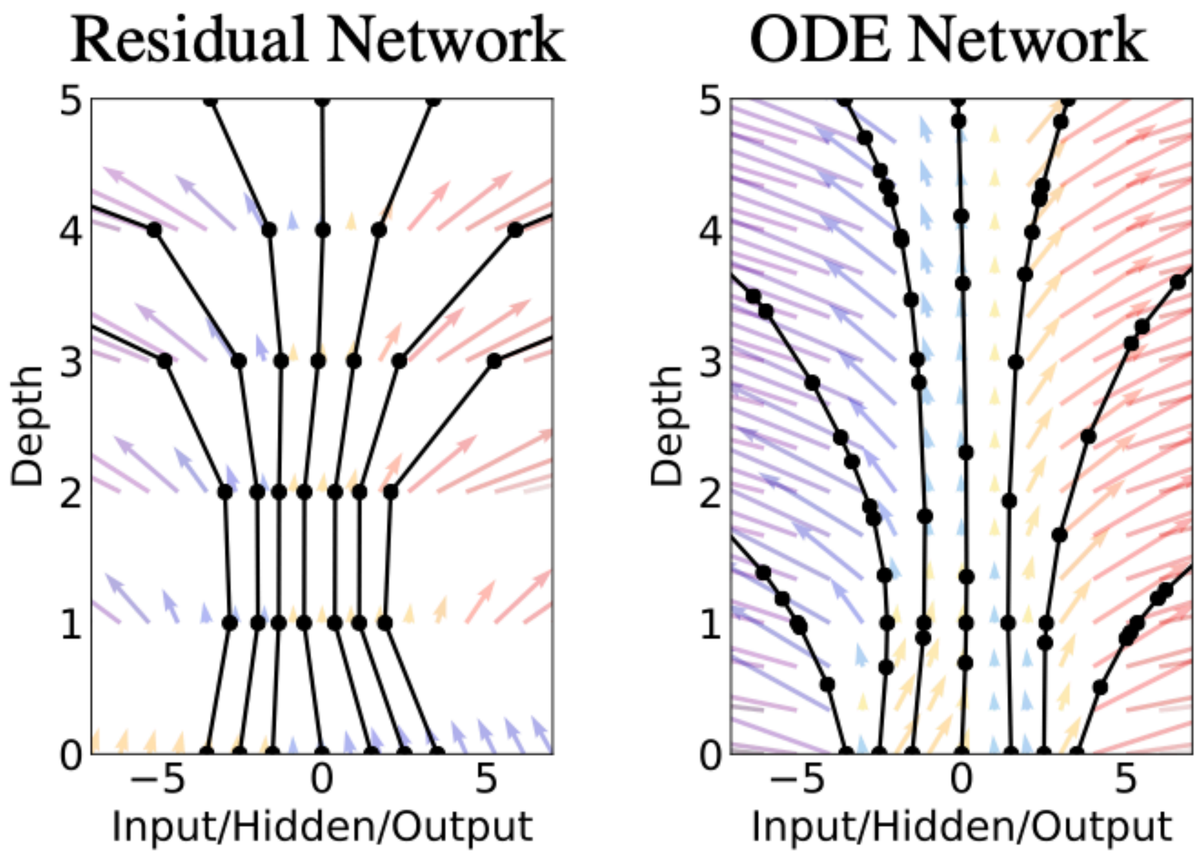
Continuous normalizing flow

- NeuralODE [Chen et al. 18], FFJORD (Grathwohl et al. 18)
- Particles $X(0) \sim p$, push particles by velocity field $v(\cdot, t) : \mathbb{R}^d \to \mathbb{R}^d$

$$\dot{x}(t) = v(x(t), t)$$

Can be parameterized by **free-form** neural networks

- Residual networks, recurrent neural network decoder: Euler discretization of a continuous transformation



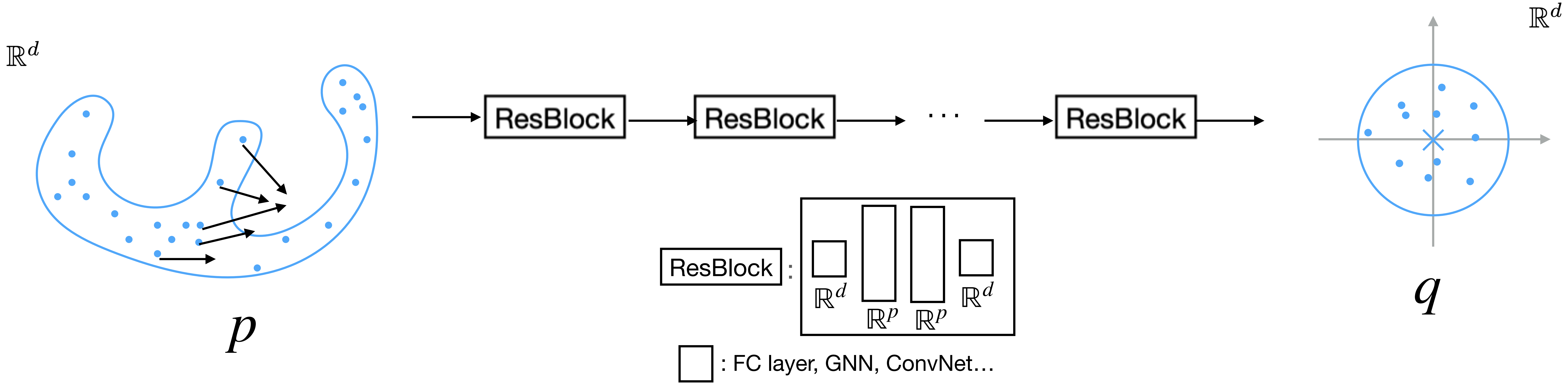
(He, Zhang, Ren, Sun 2015) (Chen, Rubanova et al. 2019)

Continuous normalizing flow

- Particles $X(0) \sim p$, push particles by velocity field $v(\,\cdot\,, t) : \mathbb{R}^d \to \mathbb{R}^d$

$$\dot{x}(t) = v(x(t), t)$$

- Implementation: Discretize into N blocks



Discrete normalizing flow

- Discrete-time version: $x_n = T_n(x_{n-1})$, T_n **invertible**

$$x_0 \xrightarrow{T_1} x_1 \xrightarrow{T_2} \cdots \xrightarrow{T_N} x_N$$

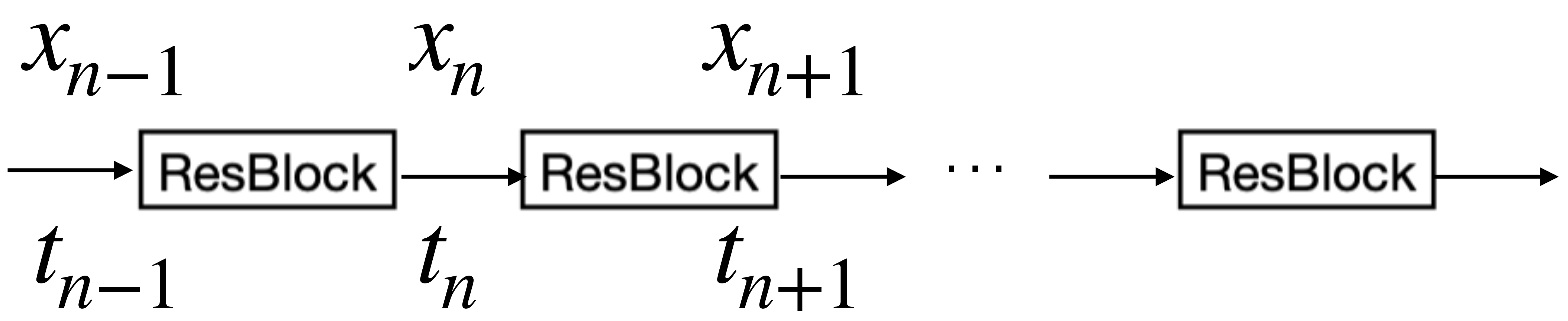
$$p \qquad\qquad\qquad\qquad q$$

Overall $T = T_N \circ \cdots \circ T_1$

- Earlier work (e.g., NICE [Dinh 15]) requires special network architectures may have limited representation power
- iResNet [Behrmann et al. 2019] utilizes extra computation (spectral normalization)

Neural ODE: Invertibility

- Invertibility of each block is ensured by continuity of (neural ODE)



- Forward

$$T_n(x_{n-1}) = x_{n-1} + \int_{t_{n-1}}^{t_n} \boxed{v(x(\tau), \tau)} d\tau, \quad x(t_{n-1}) = x_{n-1}$$

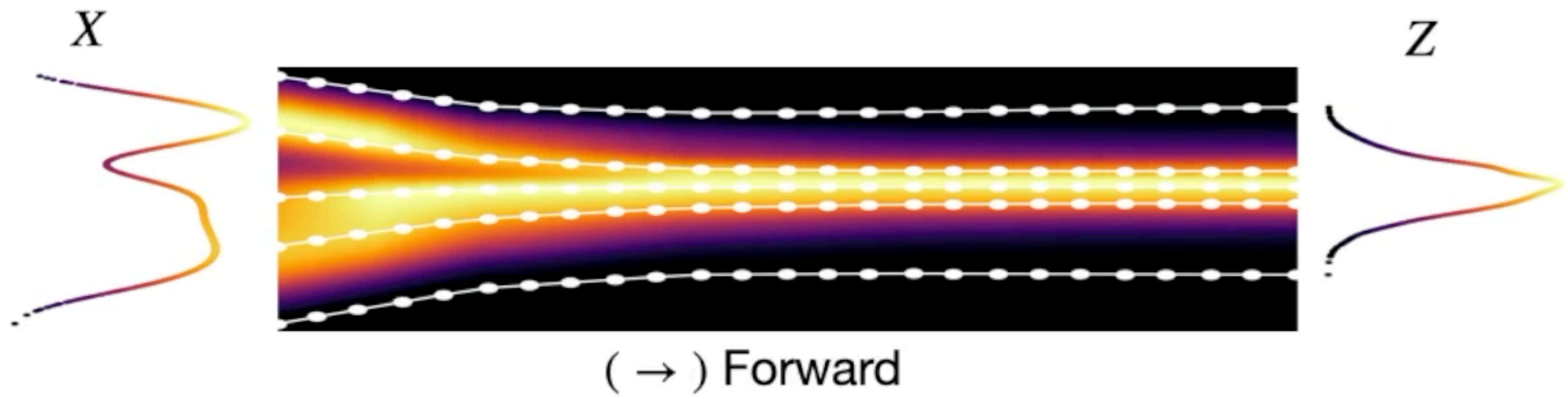
parameterized by neural networks;

numerical integral

- Reverse

$$T_n^{-1}(x_n) = x_n - \int_{t_{n-1}}^{t_n} v(x(\tau), \tau) d\tau, \quad x(t_n) = x_n$$

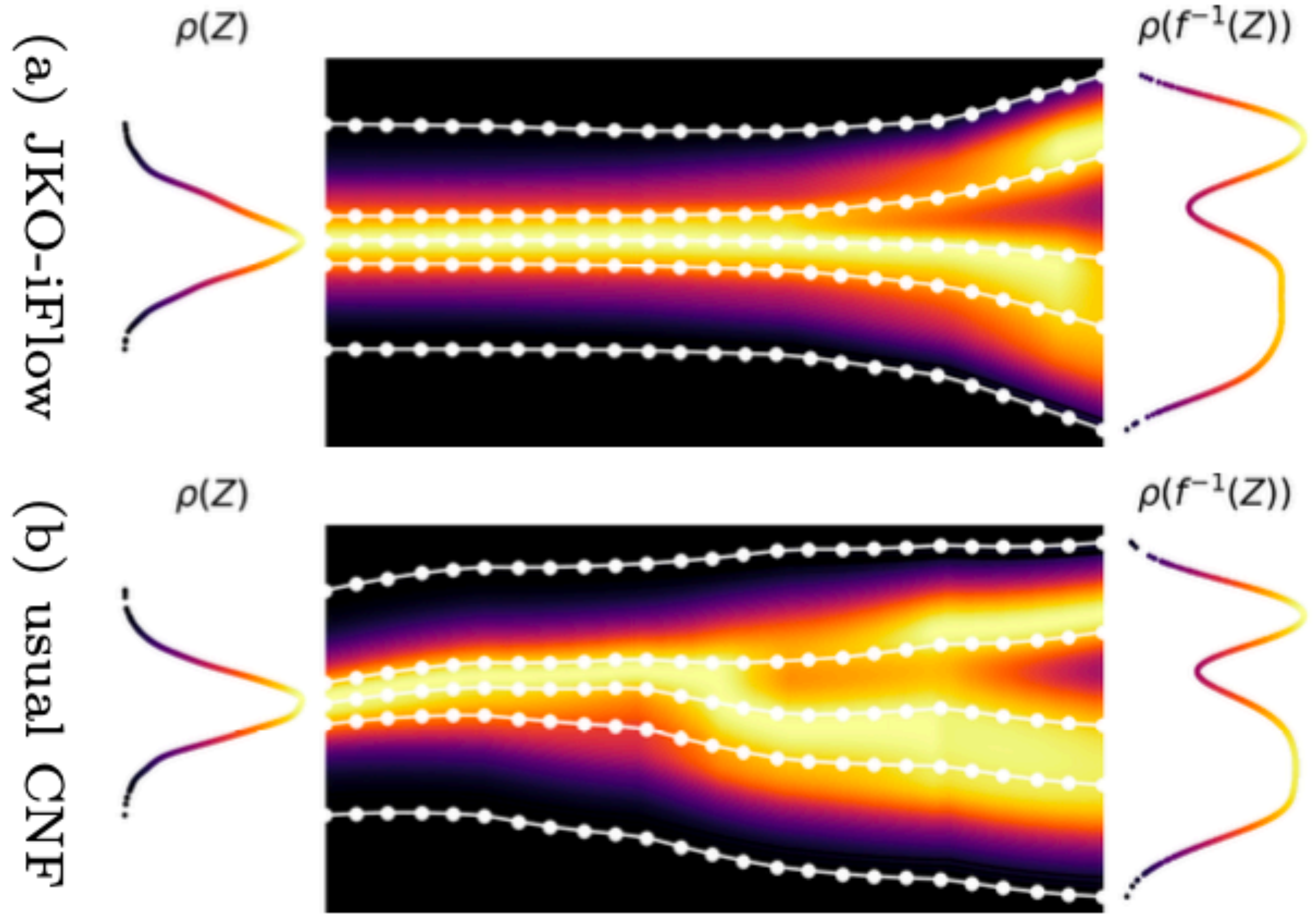
Example



$G(\rho) = \mathrm{KL}(\rho \| f_z)$

Velocity field is not unique

p



q

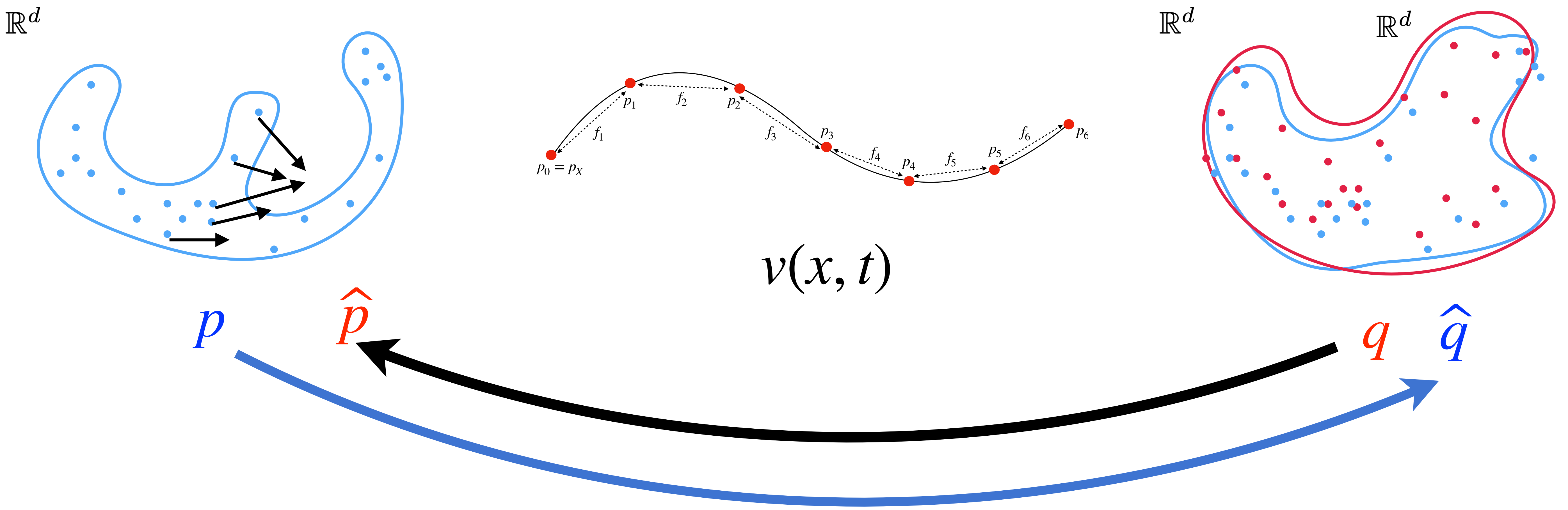
Optimal transform corresponds to “minimum energy” velocity field.

Roadmap

- Problem set-up
- Background
- Proposed algorithm
- Numerical examples
- Application: Improved density ratio estimation (DRE)

Algorithm

Matching distributions, not individual data points



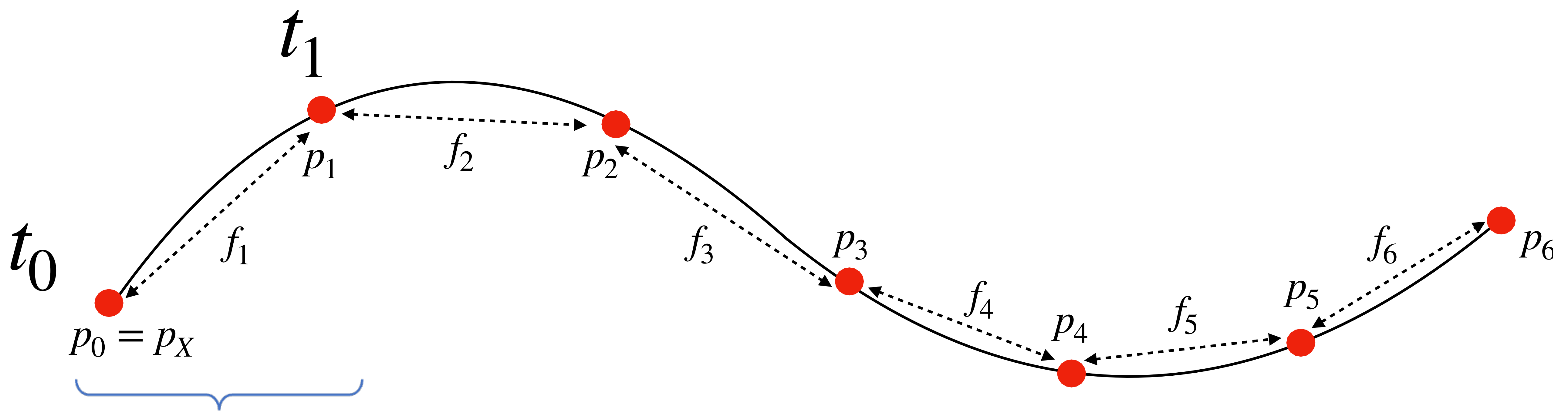
- Cast the problem as Learning velocity field $v(\cdot, t) : \mathbb{R}^d \rightarrow \mathbb{R}^d, t \in [0,1]$
- We do not know p and q, only observe through samples
- Relax terminal constraints using $\mathrm{KL}(q \| \hat{q})$ and $\mathrm{KL}(p \| \hat{p})$
- Due to symmetry: consider both directions

Algorithm

(Cont.)

- Solve the following problem

$$\text{Find } v(x,t) \text{ to minimize } \underbrace{\int_0^1 \mathbb{E}_{x\sim\rho(t)} \|v(x,t)\|_2^2 dt}_{\text{estimate using } \textbf{time discretization} \text{ and finite sample}} + \frac{\gamma}{2}\mathrm{KL}(p\|\hat{p}) + \frac{\gamma}{2}\mathrm{KL}(q\|\hat{q})$$

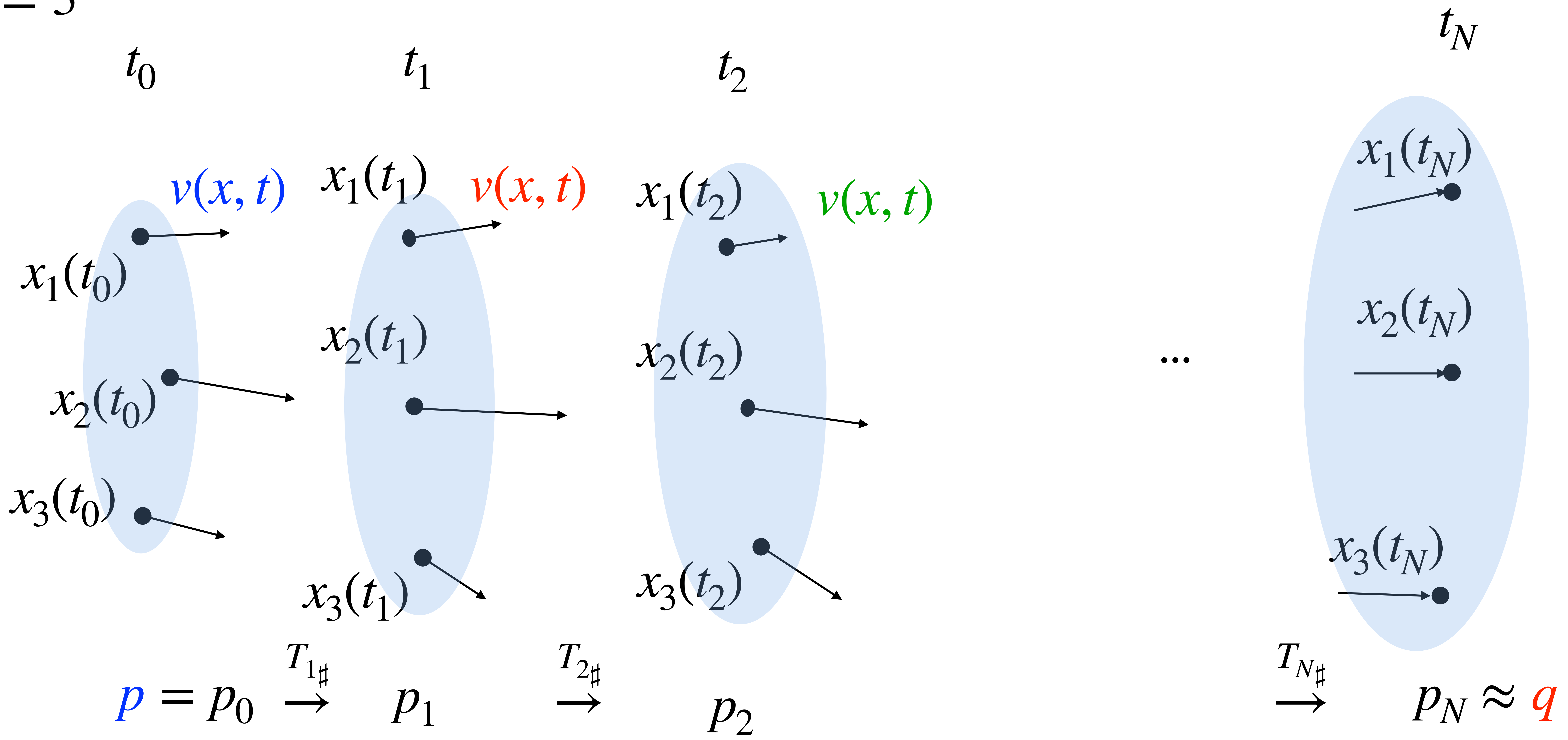


$$\text{For given } v, \int_{t_0}^{t_1} \mathbb{E}_{x\sim\rho_t} \|v(x,t)\|_2^2 dt = W_2^2(\rho_{t_0}, \rho_{t_1}) \approx \frac{1}{N}\sum_{i=1}^{N} \|x_i(t_1) - x_i(t_0)\|_2^2$$

Can also use symmetry to push from the other end.

Time discretization

$m = 3$



Algorithm

(Cont.)

- Solve the following problem

$$\text{Find } v(x,t) \text{ to minimize } \int_0^1 \mathbb{E}_{x\sim\rho(t)}\|v(x,t)\|_2^2 dt + \underbrace{\frac{\gamma}{2}\mathrm{KL}(p\|\hat{p}) + \frac{\gamma}{2}\mathrm{KL}(q\|\hat{q})}_{\substack{\text{Estimate } \mathrm{KL}(q\|\hat{q}),\ \mathrm{KL}(p\|\hat{p}) \\ \text{by GAN-loss}}}$$

<u>Lemma (Training of logistic loss leads to KL divergence under perfect training)</u>

For logistic loss, for f_0 and f_1, let

$$\ell[\varphi] = \int \log(1 + e^{\varphi(x)}) f_0(x)dx + \int \log(1 + e^{-\varphi(x)}) f_1(x)dx.$$

Then the functional global minimizer is given by $\varphi^\star = \log(f_1/f_0)$.

Roadmap

- Problem set-up
- Background
- Proposed algorithm
- Numerical examples
- Application: Improved density ratio estimation (DRE)

Comparison with other methods

- Our approach: parametrizes flow by a neural ODE
- directly solves the Benamou-Brenier equation from finite samples
- avoiding any pre-computation of OT couplings

Table 2: OT benchmarks using Gaussian mixtures with increasing dimensions (columns). Metric values ($\mathcal{L}^2$-UVP, cos) are shown in cells, with lower $\mathcal{L}^2$-UVP and higher cos being better. The last three rows are from [Korotin et al., 2021b] for comparisons.

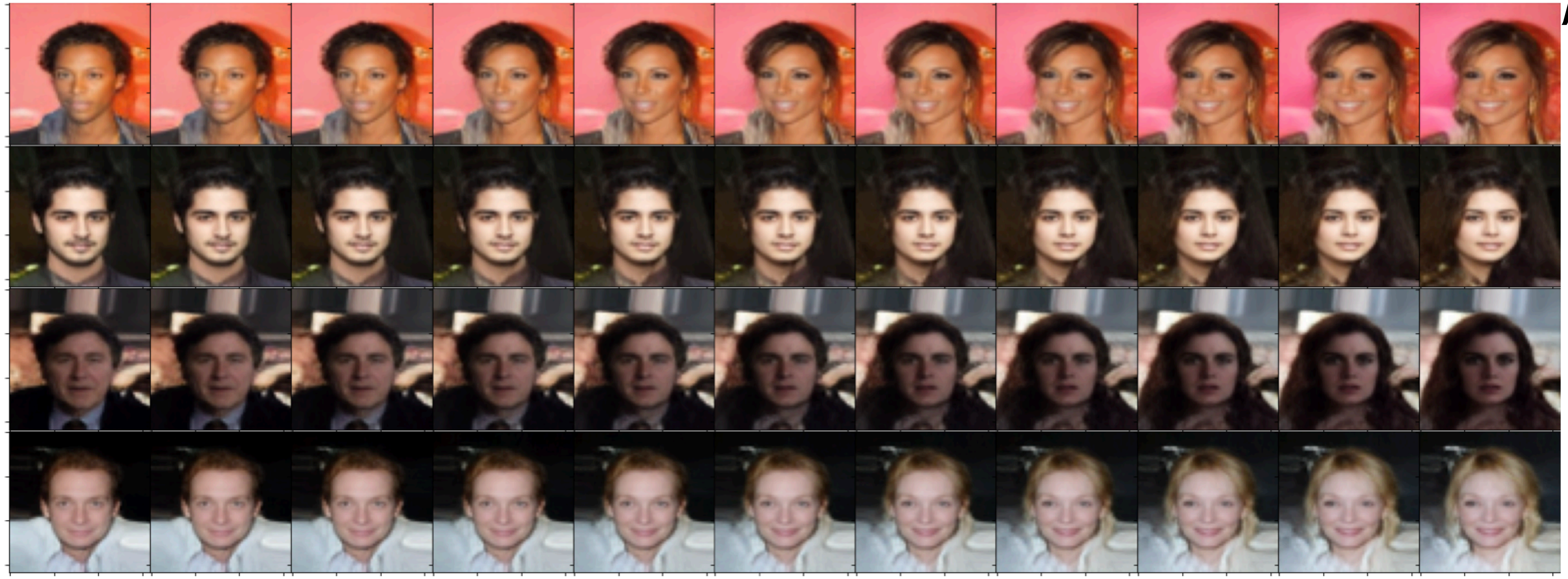
	Data dimension	**32**	**64**	**128**	**256**
	Q-flow (Ours)	**(3.27, 0.99)**	**(4.00, 0.98)**	**(2.12, 0.99)**	**(1.97, 0.99)**
Flow-matching based	OTCFM [Tong et al., 2024]	(3.74, 0.99)	(4.64, 0.97)	(2.78, 0.99)	(3.02, 0.98)
static OT	MMv1 [Taghvaei and Jalali, 2019]	(6.9, 0.98)	(8.1, 0.97)	(2.2, 0.99)	(2.6, 0.99)
	MMv2 [Fan et al., 2021]	(5.3, 0.99)	(10.1, 0.96)	(3.2, 0.99)	(2.7, 0.99)
	W2 [Korotin et al., 2021c]	(6.0, 0.99)	(7.2, 0.97)	(2.0, 1.00)	(2.7, 1.00)

Numerical example

- Using learned $\hat{v}(x, t)$ on new test sample, can perform "style transformation"

$x_i \sim \rho_t$

$\rho_0 = p$



$\rho_1 = q$

(b) CelebA male $\rightarrow$ female

Image size

64-by-64

Comparison on CelebA64 images

FID score

	Q-flow (ours)	OTCFM [Tong et al., 2024]	Re-flow [Liu et al., 2023]	MM:R [Makkuva et al., 2020]	Disco GAN [Kim et al., 2017]	Cycle GAN [Zhu et al., 2017]	NOT [Korotin et al., 2023]
Handbag → shoes	**12.34**	15.96	25.92	33.04	22.42	16.00	13.77
CelebA male → female	**9.66**	9.76	20.24	12.34	35.64	17.74	13.23

Image size: 64-by-64, dim = 4096

Latent space dim = 768

Roadmap

- Problem set-up
- Background
- Proposed algorithm
- Numerical examples
- Application: Improved density ratio estimation (DRE)

Application: Improved density ratio estimation

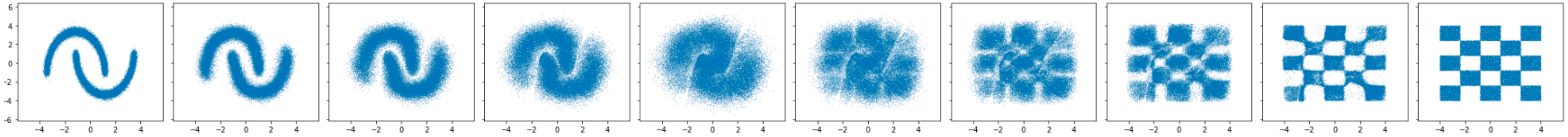
- Given **finite samples** from unknown p and q
- Density ratio estimation (DRE) $\log(p/q)$
- Idea: "infinitesimal density ratio estimation" (Choi, Meng, Song, Ermon, 2022)

$$\log\left(\frac{p}{q}\right) = \log\left(\frac{p}{p_1}\right) + \cdots + \log\left(\frac{q}{p_N}\right)$$

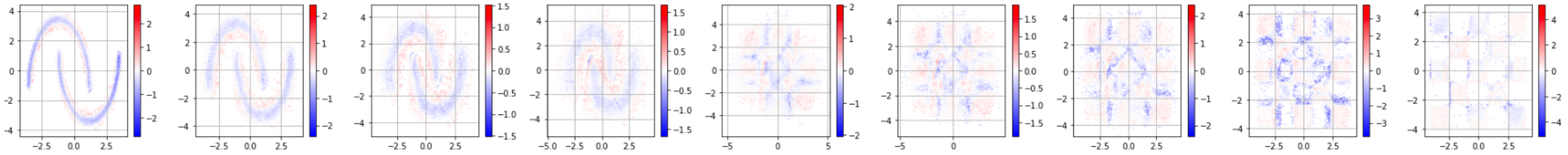
- Training by GAN applied to **transport data** over consecutive time grids

(Rhodes, Xu, Gutmann 2020)

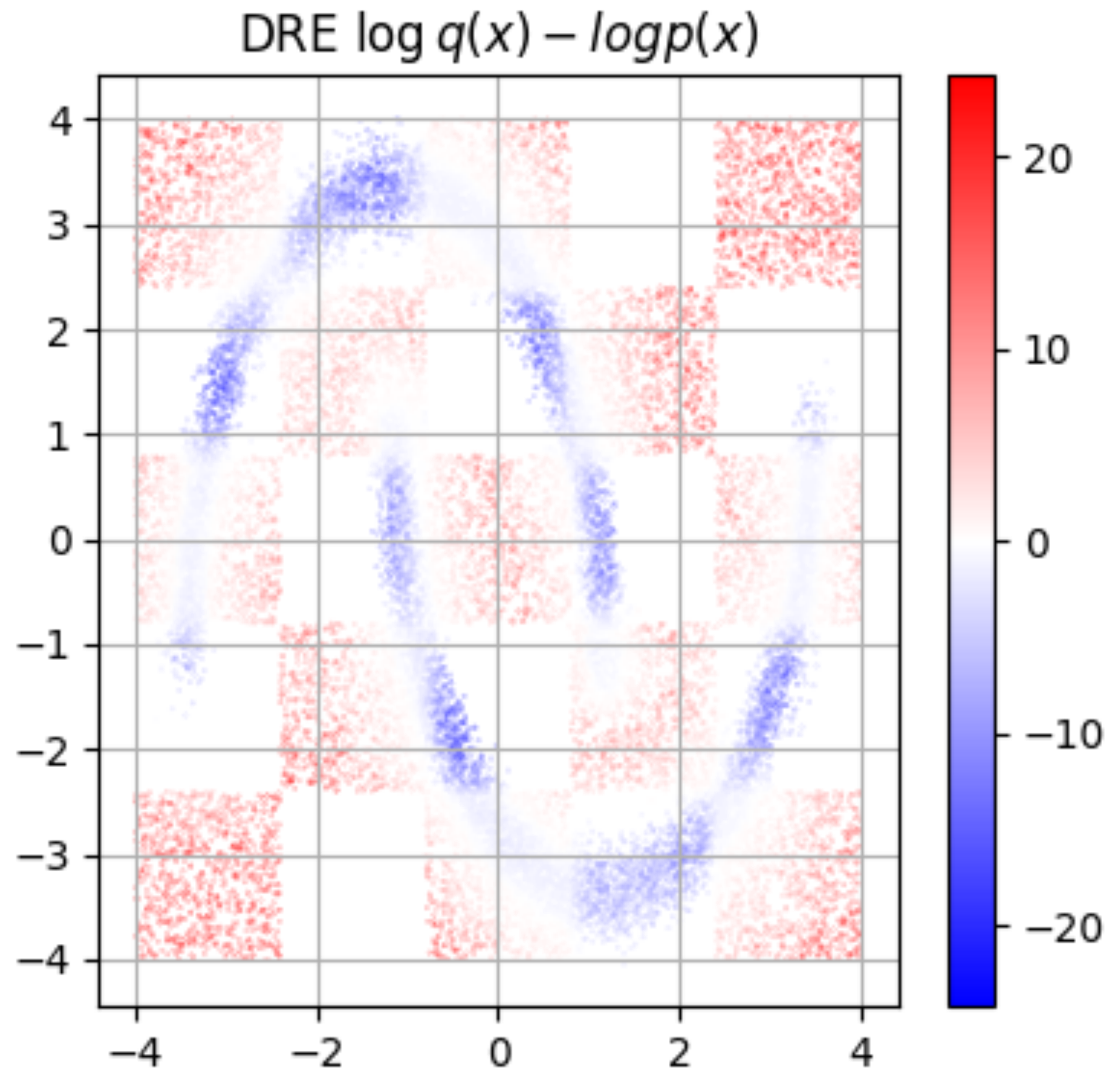
Example



(a) Trajectory from P to Q

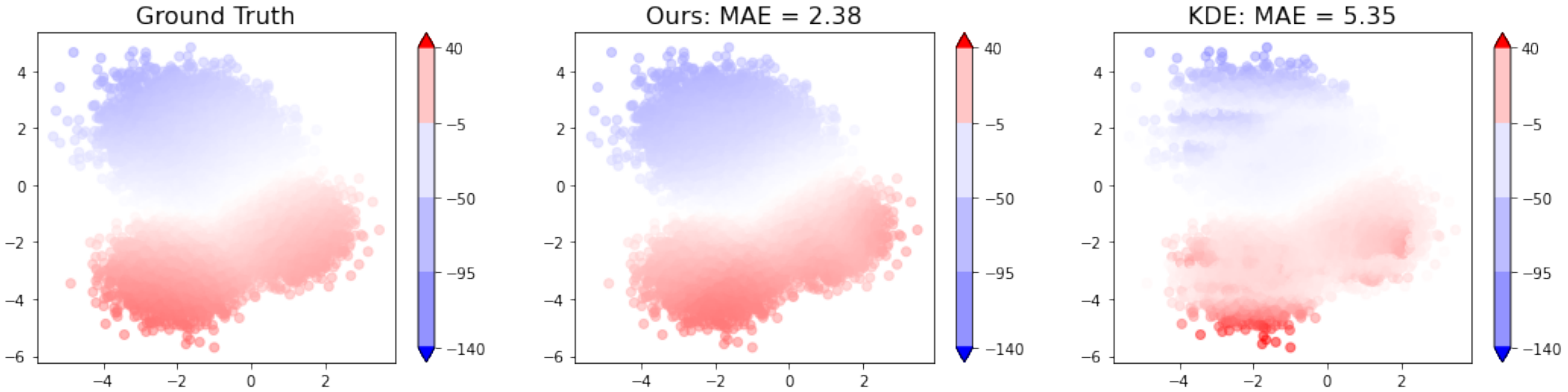


(b) Estimated log-ratio between $P_{t_{k-1}}$ and P_{t_k} by the trained flow-ratio net.



Example: Comparison

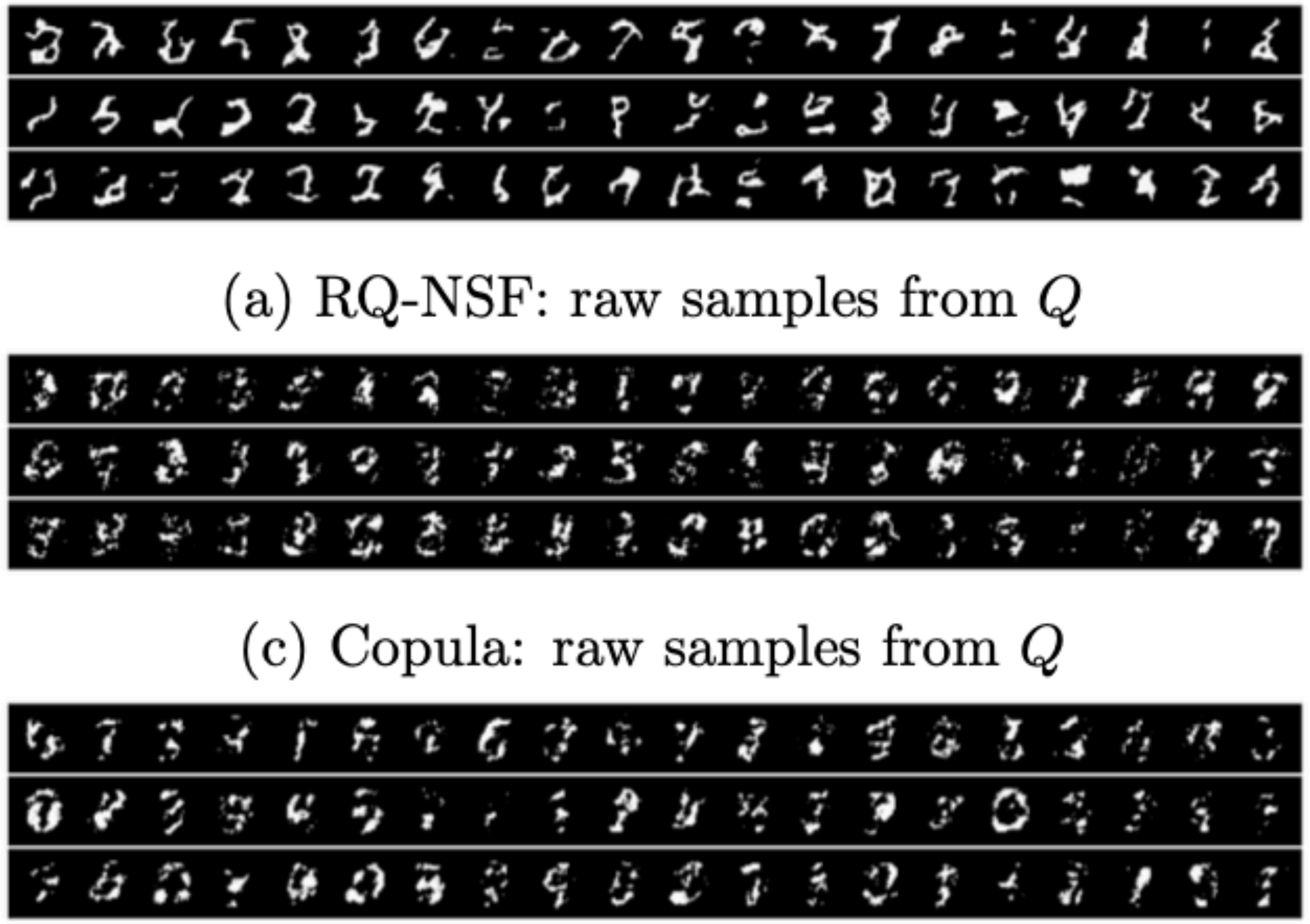
- Density ratio between two Gaussian mixtures
- $p = \frac{1}{3}\mathcal{N}(\begin{bmatrix}-2\\2\end{bmatrix}, 0.75I_2) + \frac{1}{3}\mathcal{N}(\begin{bmatrix}-1.5\\1.5\end{bmatrix}, 0.25I_2) + \frac{1}{3}\mathcal{N}(\begin{bmatrix}-1\\1\end{bmatrix}, 0.75I_2)$
- $q = \frac{1}{2}\mathcal{N}(\begin{bmatrix}0.75\\-1.5\end{bmatrix}, 0.5I_2) + \frac{1}{2}\mathcal{N}(\begin{bmatrix}-2\\-3\end{bmatrix}, 0.5I_2)$



Comparison for DRE estimation

Table 1: DRE performance on the energy-based modeling task for MNIST, reported in BPD and lower, is better. Results for DRE-∞ are from [Choi et al., 2022], and results for one ratio and TRE are from [Rhodes et al., 2020].

Choice of Q	RQ-NSF				Copula				Gaussian			
Method	Ours	DRE-∞	TRE	1 ratio	Ours	DRE-∞	TRE	1 ratio	Ours	DRE-∞	TRE	1 ratio
BPD ($\downarrow$)	**1.05**	1.09	1.09	1.09	**1.14**	1.21	1.24	1.33	**1.31**	1.33	1.39	1.96



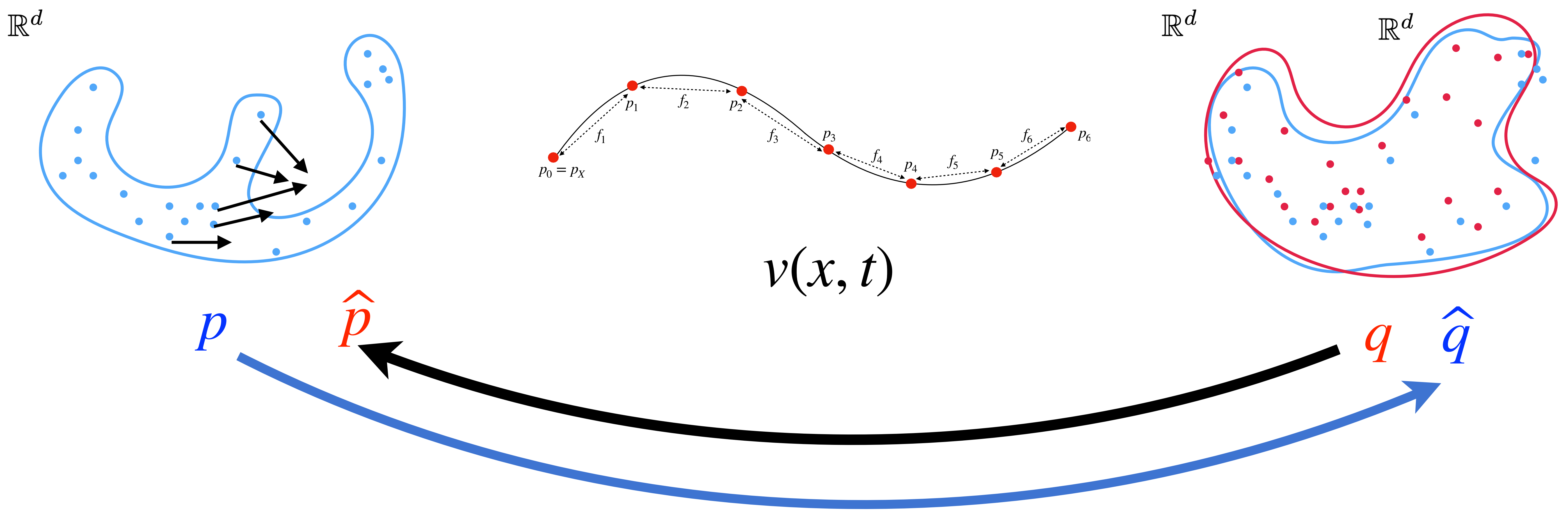
(a) RQ-NSF: raw samples from Q

(c) Copula: raw samples from Q

(e) Gaussian: raw samples from Q

Summary

- Compute optimal transport (OT) using **dynamic** formula
- Parametrizes flow by a **neural ODE**
- directly solves the **Benamou-Brenier equation** from finite samples
- avoiding pre-computation of OT couplings



Computing high-dimensional optimal transport by flow neural networks. Cheng, **X.** arXiv:2305.11857. 2024