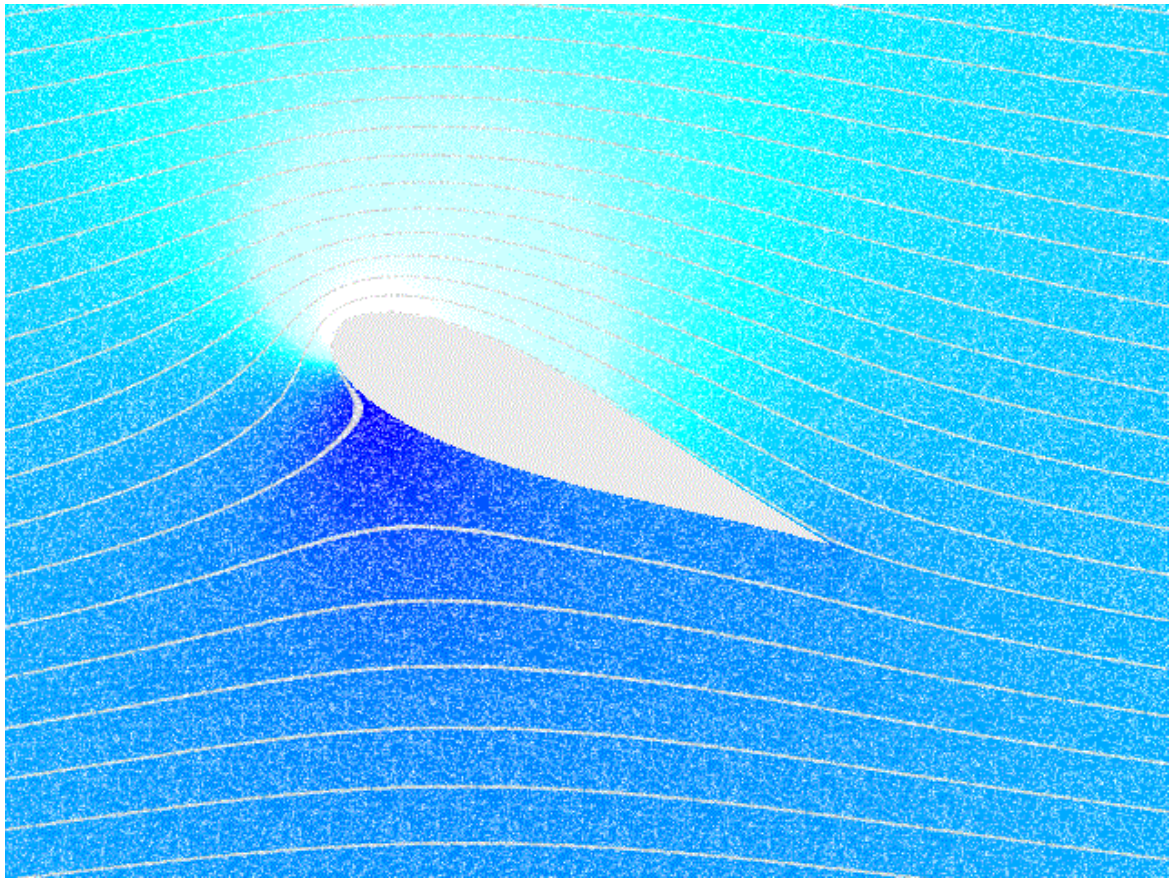
Self-Consistent Velocity Matching of Probability Flows

Li et al.

Andrin Zoller & Nicola Witzig

01. 06 2024, Zurich

Mass Conserving Phenomena



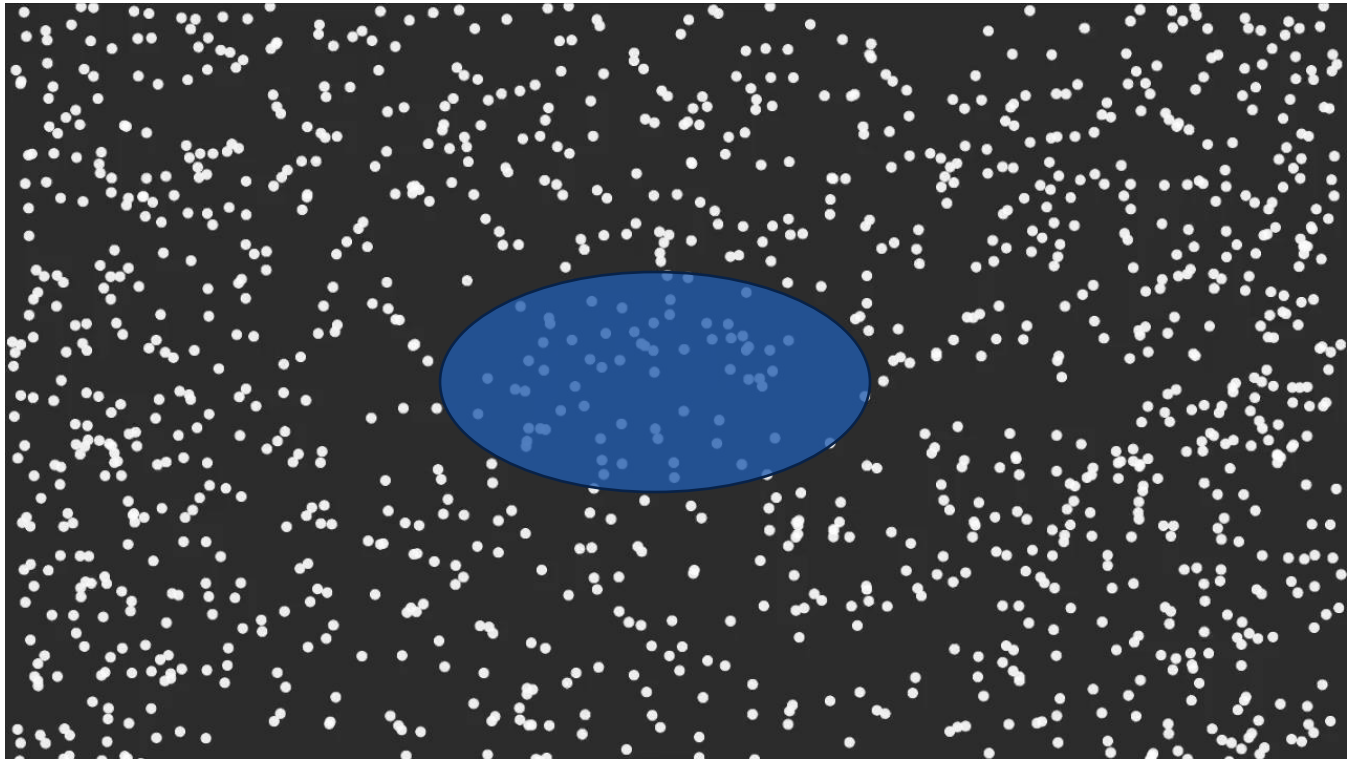
Sources: https://en.m.wikipedia.org/wiki/File:Flow_around_a_wing.gif

Continuity equation

$$\partial_t p_t(x) = -\nabla \cdot (v_t p_t), \forall x, t \in [0, T],$$

Continuity equation

$$\partial_t p_t(x) = -\nabla \cdot (v_t p_t), \forall x, t \in [0, T],$$



Continuity equation

$$\partial_t p_t(x) = -\nabla \cdot (v_t p_t), \forall x, t \in [0, T],$$

$\Rightarrow \quad \int p_t(x)\,\mathrm{d}x$ is conserved

$\Rightarrow$ p_t and v_t are coupled, as the evolution of p_t is determined by v_t

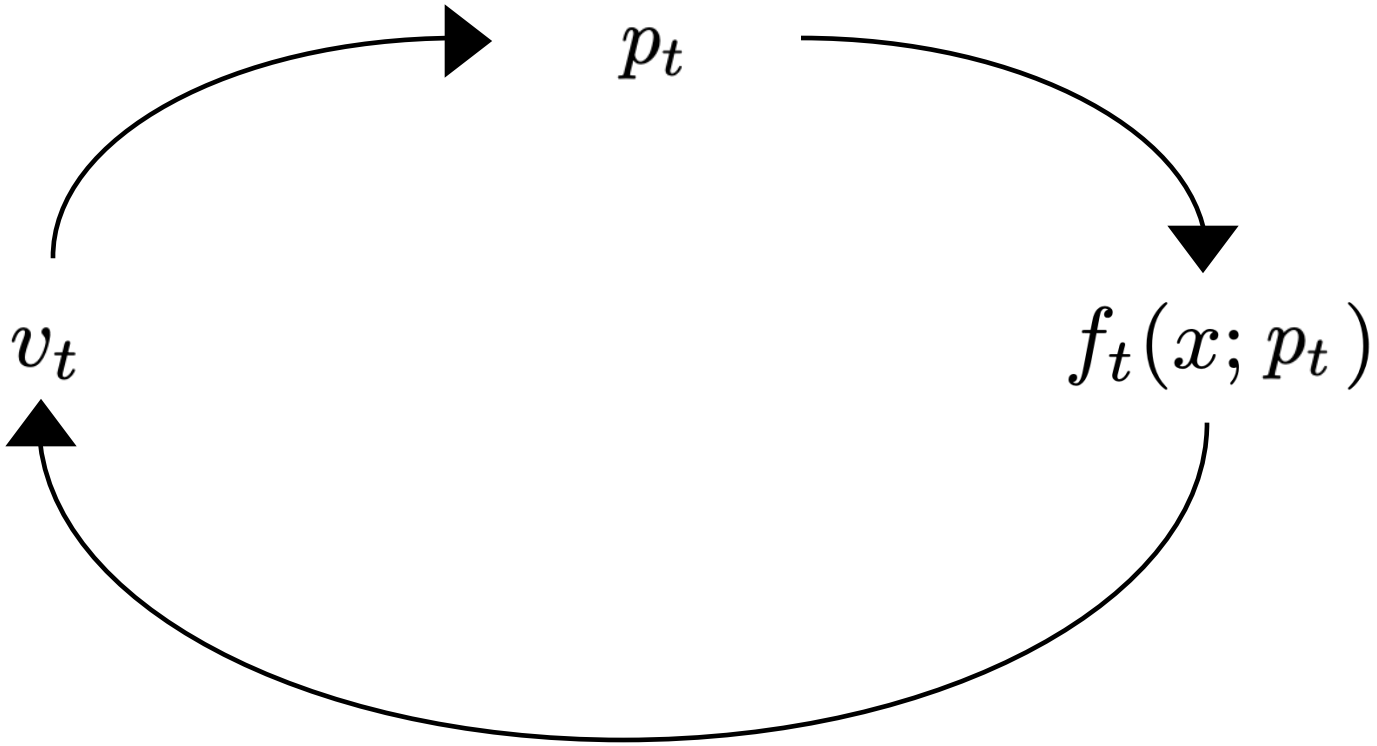
Continuity equation

$$\partial_t p_t(x) = -\nabla \cdot (v_t p_t), \forall x, t \in [0, T],$$

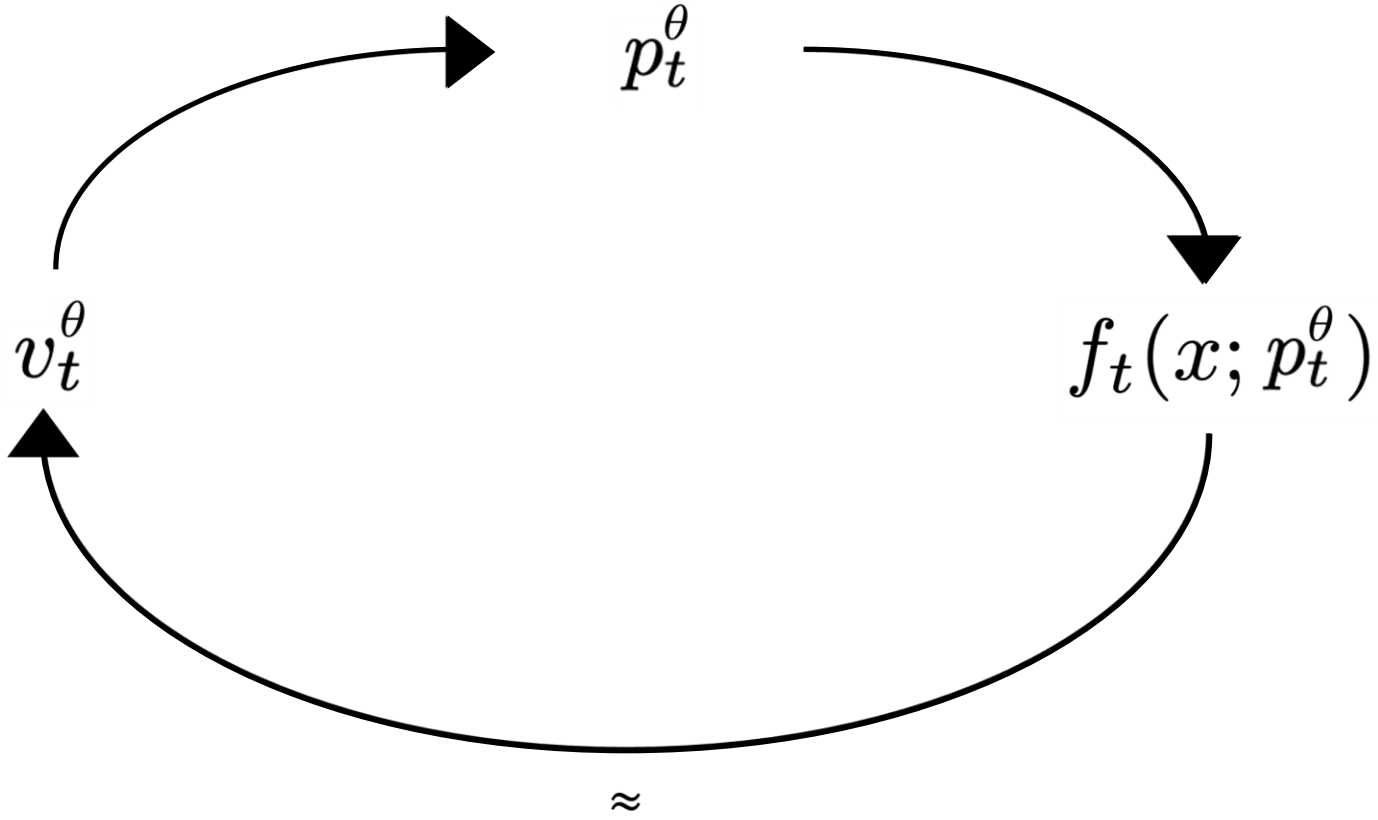
$$\partial_t p_t(x) = -\nabla \cdot (f_t(x; p_t) p_t), \forall x, t \in [0, T],$$

- $f_t(x; p_t)$ is a given function depending on p_t
- Many problems fall into this class of PDEs: Wasserstein gradient flow, Fokker-Planck

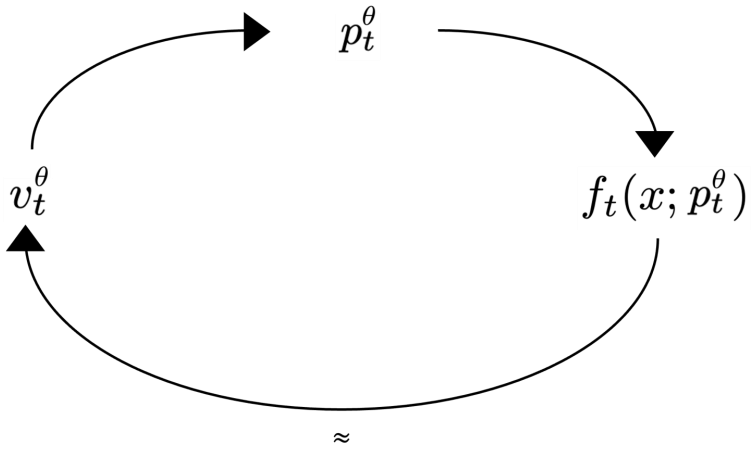
Self-consistency



Self-consistency



Self-consistency



Self-consistency loss: $$L(\theta) := \int_0^T \mathbf{E}_{X \sim \mu_t^\theta} \left[\left\| v_t^\theta(X) - f_t(X; p_t^\theta) \right\|^2 \right] \mathrm{d}t$$

μ_t^θ : probability measure with density function p_t^θ

Iterative Method (SCVM)

$$\theta_{k+1} := \theta_k - \eta \nabla_\theta|_{\theta=\theta_k} F(\theta, \theta_k),$$

$$F(\theta, \theta_k) := \int_0^T \mathbf{E}_{X \sim \mu_t^{\theta_k}} \left[\left\| v_t^\theta(X) - f_t(X; p_t^{\theta_k}) \right\|^2 \right] \mathrm{d}t.$$

$$L(\theta) := \int_0^T \mathbf{E}_{X \sim \mu_t^{\theta}} \left[\left\| v_t^\theta(X) - f_t(X; p_t^{\theta}) \right\|^2 \right] \mathrm{d}t$$

θ_k : Network weights at iteration k

θ : Variable of network weights

Iterative Method (SCVM)

$$\nabla L(\theta) = \int_0^T \int \nabla_\theta p_t^\theta(x) \left\| v_t^\theta(x) - f_t^\theta(x) \right\|^2 \mathrm{d}x \, \mathrm{d}t$$

$$+ 2 \int_0^T \int p_t^\theta(x) J_\theta v_t^\theta(x)^\top (v_t^\theta(x) - f_t^\theta(x)) \, \mathrm{d}x \, \mathrm{d}t$$

$$- 2 \int_0^T \int p_t^\theta(x) J_\theta f_t^\theta(x)^\top (v_t^\theta(x) - f_t^\theta(x)) \, \mathrm{d}x \, \mathrm{d}t.$$

$$\nabla_\theta F(\theta, \theta_k) = \int_0^T \int \nabla_\theta p_t^{\theta_k}(x) \left\| v_t^\theta(x) - f_t^{\theta_k}(x) \right\|^2 \mathrm{d}x \, \mathrm{d}t$$

$$+ 2 \int_0^T \int p_t^{\theta_k}(x) J_\theta v_t^\theta(x)^\top (v_t^\theta(x) - f_t^{\theta_k}(x)) \, \mathrm{d}x \, \mathrm{d}t$$

$$- 2 \int_0^T \int p_t^{\theta_k}(x) J_\theta f_t^{\theta_k}(x)^\top (v_t^\theta(x) - f_t^{\theta_k}(x)) \, \mathrm{d}x \, \mathrm{d}t.$$

$$f_t^\theta(x) := f_t(X; \mu_t^\theta)$$

θ_k : Network weights at iteration k

θ : Variable of network weights

Iterative Method (SCVM)

$$\nabla L(\theta) = \int_0^T \int \nabla_\theta p_t^\theta(x) \left\| v_t^\theta(x) - f_t^\theta(x) \right\|^2 \mathrm{d}x \, \mathrm{d}t$$

$$+ 2 \int_0^T \int p_t^\theta(x) J_\theta v_t^\theta(x)^\top (v_t^\theta(x) - f_t^\theta(x)) \, \mathrm{d}x \, \mathrm{d}t$$

$$- 2 \int_0^T \int p_t^\theta(x) J_\theta f_t^\theta(x)^\top (v_t^\theta(x) - f_t^\theta(x)) \, \mathrm{d}x \, \mathrm{d}t.$$

$$f_t^\theta(x) := f_t(X; \mu_t^\theta)$$

$$\nabla_\theta F(\theta, \theta_k) = \int_0^T \int \cancelto{0}{\nabla_\theta p_t^{\theta_k}(x)} \left\| v_t^\theta(x) - f_t^{\theta_k}(x) \right\|^2 \mathrm{d}x \, \mathrm{d}t$$

$$+ 2 \int_0^T \int p_t^{\theta_k}(x) J_\theta v_t^\theta(x)^\top (v_t^\theta(x) - f_t^{\theta_k}(x)) \, \mathrm{d}x \, \mathrm{d}t$$

$$- 2 \int_0^T \int p_t^{\theta_k}(x) J_\theta \cancelto{0}{f_t^{\theta_k}(x)}^\top (v_t^\theta(x) - f_t^{\theta_k}(x)) \, \mathrm{d}x \, \mathrm{d}t.$$

θ_k : Network weights at iteration k

θ : Variable of network weights

Iterative Method (SCVM)

$$\nabla L(\theta) = \int_0^T \int \nabla_\theta p_t^\theta(x) \left\| v_t^\theta(x) - f_t^\theta(x) \right\|^2 \mathrm{d}x \, \mathrm{d}t$$

$$+ 2 \int_0^T \int p_t^\theta(x) J_\theta v_t^\theta(x)^\top (v_t^\theta(x) - f_t^\theta(x)) \, \mathrm{d}x \, \mathrm{d}t$$

$$- 2 \int_0^T \int p_t^\theta(x) J_\theta f_t^\theta(x)^\top (v_t^\theta(x) - f_t^\theta(x)) \, \mathrm{d}x \, \mathrm{d}t.$$

$$\nabla_\theta F(\theta, \theta_k) = 2 \int_0^T \int p_t^{\theta_k}(x) J_\theta v_t^\theta(x)^\top (v_t^\theta(x) - f_t^{\theta_k}(x)) \, \mathrm{d}x \, \mathrm{d}t$$

Biased gradient estimator

$$f_t^\theta(x) := f_t(X; \mu_t^\theta)$$

θ_k : Network weights at iteration k

θ : Variable of network weights

Parametrization

Flow map: $\Phi_t : \mathbf{R}^d \to \mathbf{R}^d$

Velocity field: $v_t : \mathbf{R}^d \to \mathbf{R}^d$

Probability density: $p_t : \mathbf{R}^d \to \mathbf{R}$

Parametrization - Normalizing Flows

Parametrized by the model:

Flow map: $\Phi_t(x)$

Retrieved by derivation:

Density: $$\log p_t(x) = \log p_0^*(\Phi_t^{-1}(x)) + \log \left| \det J\Phi_t^{-1}(x) \right|$$

Velocity field: $$v_t(x) = \partial_t \Phi_t(\Phi_t^{-1}(x))$$

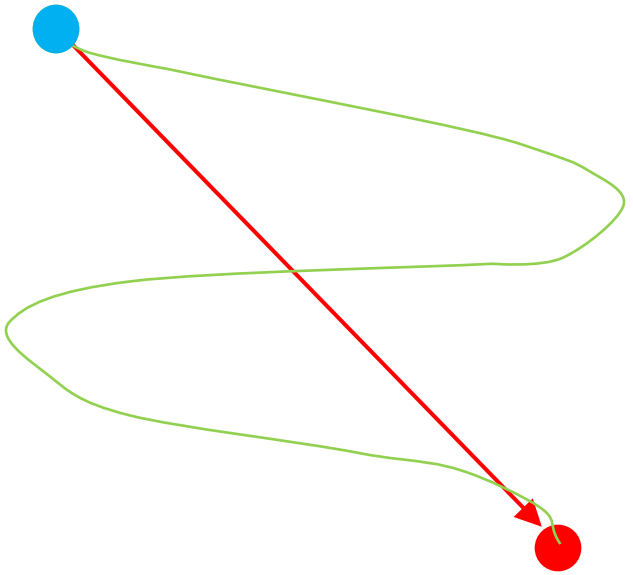
Parametrization - Normalizing Flows

$$\Phi_t(x_0) = x$$

$$\Phi_t(x_0) = starting\ point + path\ of\ particle\ over\ t\ timesteps$$

$$v_t(x) = \frac{dx}{dt}$$

$$\Phi_t(x_0) = x_0 + \int_0^t \frac{dx}{ds}\, ds$$



Parametrization - Normalizing Flows

Definitions:

$$v_t(x) = \frac{dx}{dt}$$

$$\Phi_t(x_0) = x_0 + \int_0^t \frac{dx}{ds}\, ds$$

$$\Phi_t^{-1}(x) = x_0$$

Proof:

$$\partial_t \Phi_t(x_0) = \partial_t x_0 + \partial_t \int_0^T \frac{dx}{ds}\, ds$$

$$\partial_t \Phi_t(x_0) = 0 + \partial_t \int_0^t \frac{dx}{ds}\, ds$$

$$\partial_t \Phi_t(x_0) = \frac{dx}{dt}$$

$$\partial_t \Phi_t(x_0) = v_t(x)$$

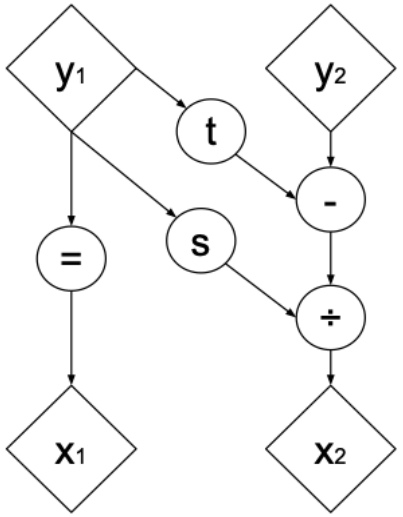
$$\partial_t \Phi_t\left(\Phi_t^{-1}(x)\right) = v_t(x)$$

Parametrization - Normalizing Flows

$$\log p_t(x) = \log p_0^*(\Phi_t^{-1}(x)) + \log \left| \det J\Phi_t^{-1}(x) \right|$$

$$v_t(x) = \partial_t \Phi_t(\Phi_t^{-1}(x))$$

Parametrization - Normalizing Flows



(a) Forward propagation

Coupling Layer

$$y_{1:d} = y_1$$
$$y_{d+1:D} = y_2$$
$$x_{1:d} = x_1$$
$$x_{d+1:D} = x_2$$

(a) Forward propagation

$$y_{1:d} = x_{1:d}$$
$$y_{d+1:D} = x_{d+1:D} \odot \exp\left(s(x_{1:d})\right) + t(x_{1:d})$$

(b) Inverse propagation

$$x_{1:d} = y_{1:d}$$
$$x_{d+1:D} = \left(y_{d+1:D} - t(y_{1:d})\right) \odot \exp\left(-s(y_{1:d})\right)$$

Source: https://arxiv.org/abs/1605.08803 [Dinh et al. 2016]

Parametrization - Normalizing Flows

Jacobian:

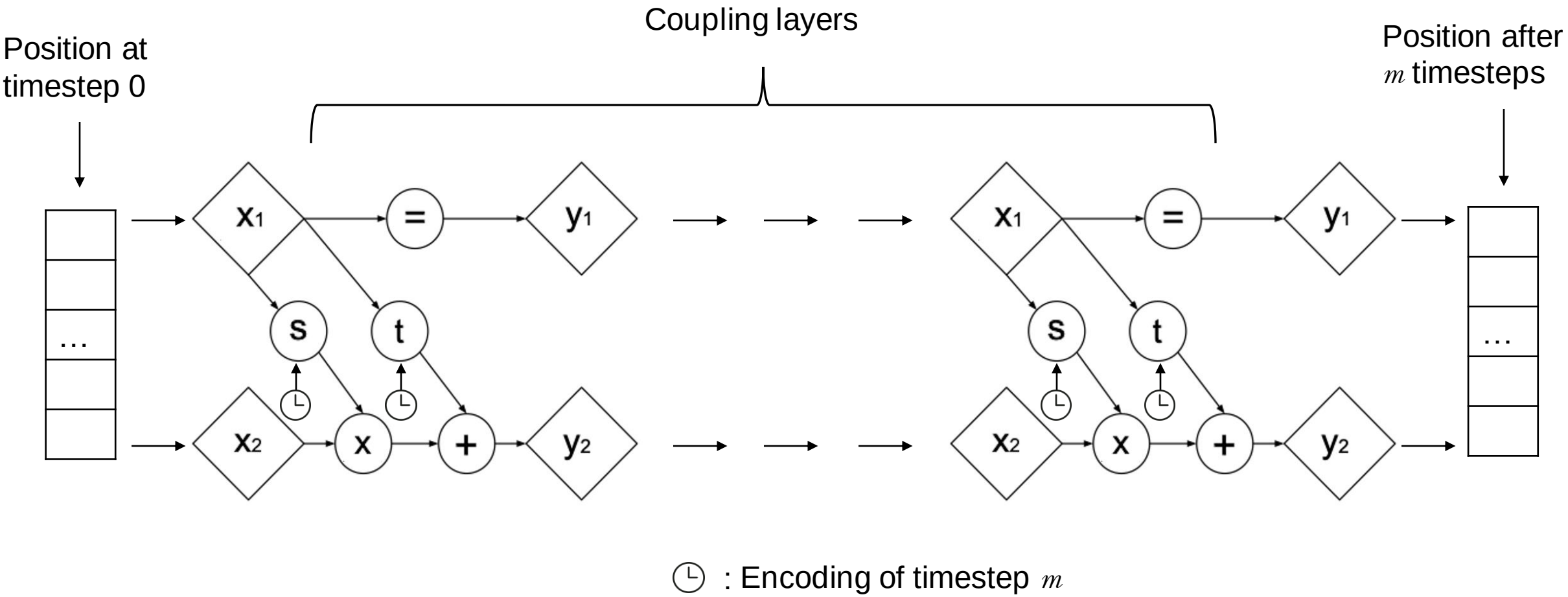
$$\frac{\partial y}{\partial x^T} = \begin{bmatrix} \mathbb{I}_d & 0 \\ \frac{\partial y_{d+1:D}}{\partial x_{1:d}^T} & \operatorname{diag}\left(\exp\left[s\left(x_{1:d}\right)\right]\right) \end{bmatrix}$$

Determinant:

$$\exp\left[\sum_j s\left(x_{1:d}\right)_j\right]$$

Source: https://arxiv.org/abs/1605.08803 [Dinh et al. 2016]

Parametrization - Normalizing Flows



Source: https://arxiv.org/abs/1605.08803 [Dinh et al. 2016]

Parametrization - Neural ODE

Parametrized by the model:

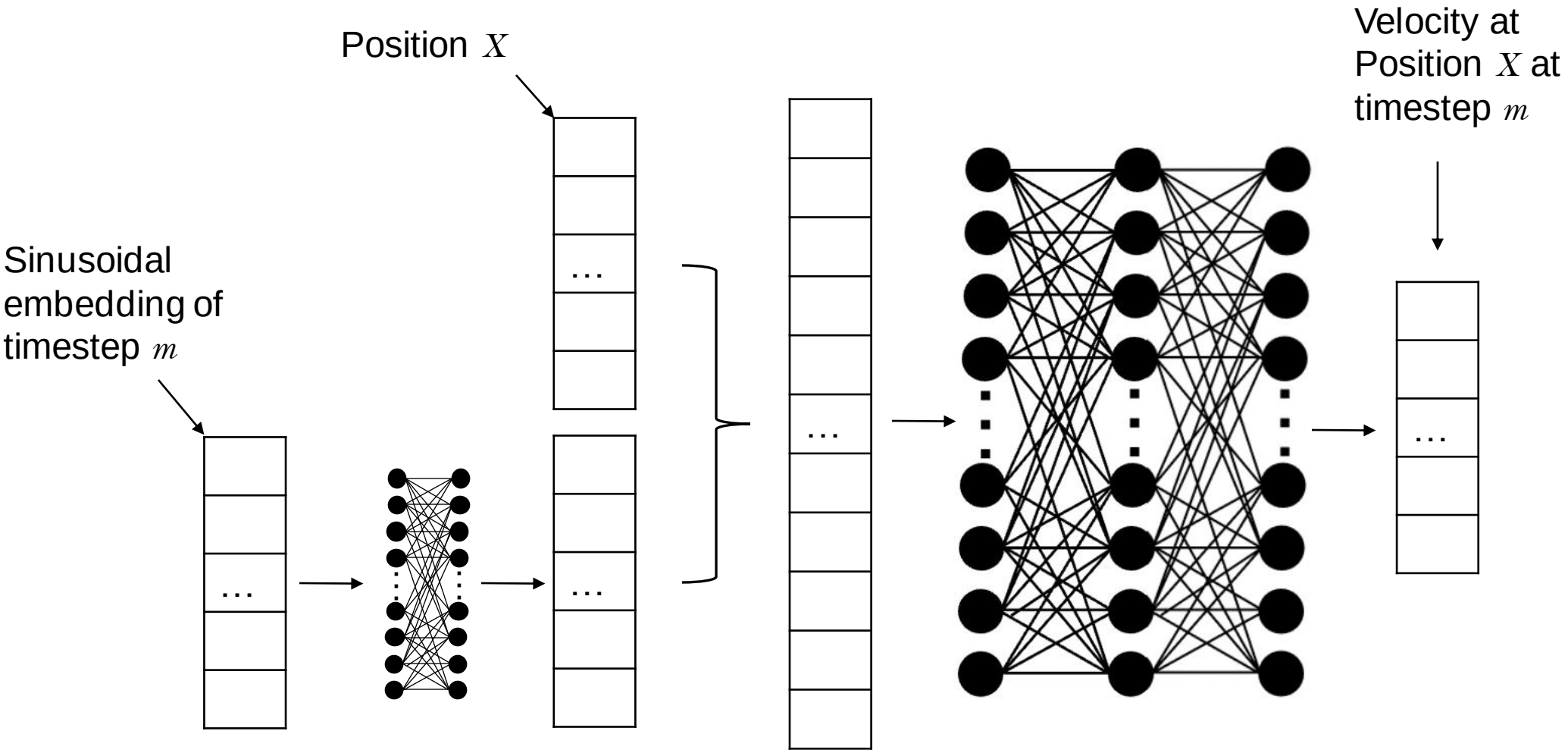
Velocity field: $v_t(x)$

Retrieved by derivation:

Flow map: $$\Phi_t(x) = x + \int_0^t v_s(\Phi_s(x))\,\mathrm{d}s$$

Density: $$\log p_t(\Phi_t(x)) = \log p_0^*(x) - \int_0^t \nabla \cdot v_s(\Phi_s(x))\,\mathrm{d}s$$

Parametrization - Neural ODE



Parametrization

Normalizing Flows

- Models the flow map
- Infeasable in higher dimensions

Neural ODE

- Models the velocity field
- Uses numerical integration

Both inherently conserve mass

Integration by parts trick [Hyvärinen and Dayan, 2005]

Recall:

$$F(\theta, \theta_k) := \int_0^T \mathbf{E}_{X \sim \mu_t^{\theta_k}} \left[\left\| v_t^\theta(X) - f_t(X; p_t^{\theta_k}) \right\|^2 \right] \mathrm{d}t$$

Middle term:

$$\mathbf{E}_{X \sim \mu_t^{\theta_k}} \left[v_t^\theta(X)^\top f_t(X; p_t^{\theta_k}) \right]$$

For f_t of this form:

$$f_t(x; \mu_t) = b_t(x) - D_t(x) \nabla \log p_t(x)$$

Density for NODE parametrization:

$$\log p_t(\Phi_t(x)) = \log p_0^*(x) - \int_0^t \boldsymbol{\nabla} \cdot v_s(\Phi_s(x))$$

Integration by parts trick [Hyvärinen and Dayan, 2005]

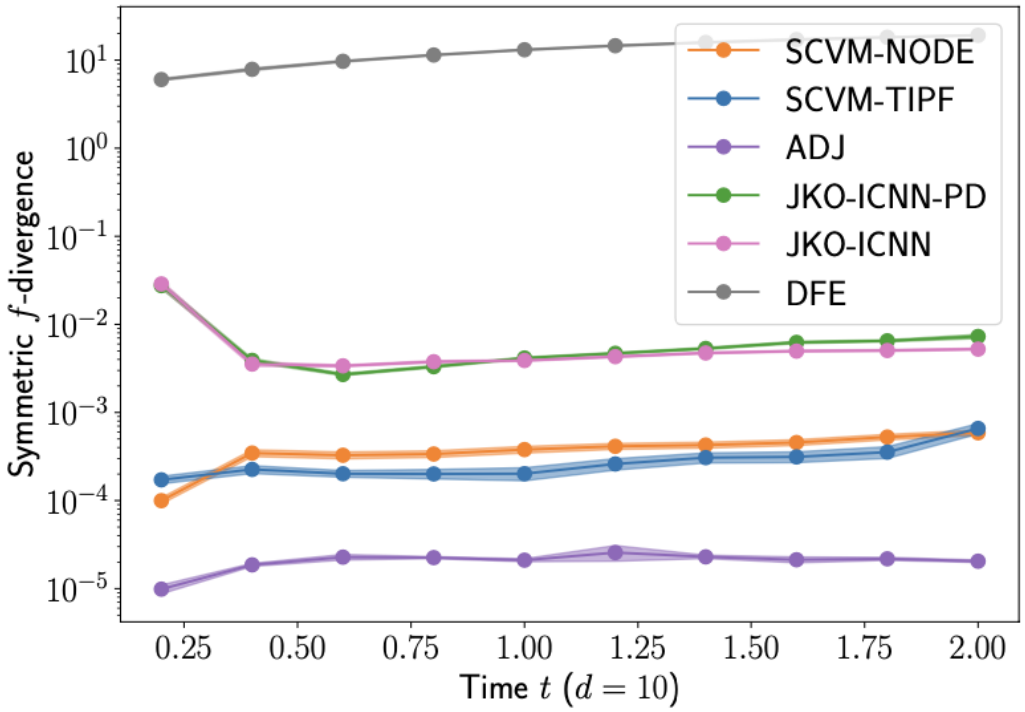
Trick:

$$\mathbf{E}_{X\sim\mu_t^{\theta_k}}\left[v_t^\theta(X)^\top f_t(X;\mu_t^{\theta_k})\right] = \mathbf{E}_{X\sim\mu_t^{\theta_k}}\left[v_t^\theta(X)^\top b_t(X) + \boldsymbol{\nabla}\cdot\left(D_t^\top(X)v_t^\theta(X)\right)\right]. \tag{10}$$

Condition: f_t depends on the density of μ_t only through the score $\nabla \log p_t$

For example, with f_t to model the Fokker-Plank Equation

Experiments – Ornstein-Uhlenbeck process



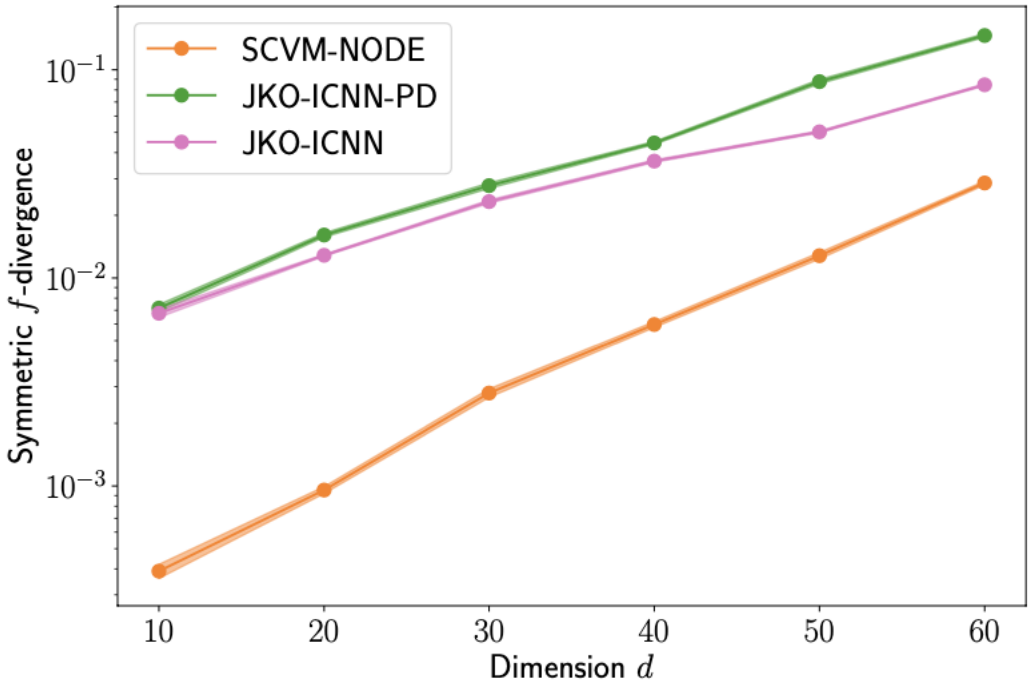
- The probability measure μ_t is known for any t in closed form if μ_0 is Gaussian
- Symmetric f-divergence
- SCVM-TIPF more accurate than SCVM-NODE (direct density access)
- Low training time
- Higher dimensions

Symmetric f-divergence:

$$D_f(\rho_1 \parallel \rho_2) := \mathbf{E}_{X\sim\rho_2}\left[(\log \rho_1(X) - \log \rho_2(X))^2/2\right]$$

$$\mathbf{Sym}D_f(\rho_1, \rho_2) := D_f(\rho_1 \parallel \rho_2) + D_f(\rho_2 \parallel \rho_1)$$

Experiments – Ornstein-Uhlenbeck process



- The probability measure μ_t is known for any t in closed form if μ_0 is Gaussian
- Symmetric f-divergence
- SCVM-TIPF more accurate than SCVM-NODE (direct density access)
- Low training time
- Higher dimensions

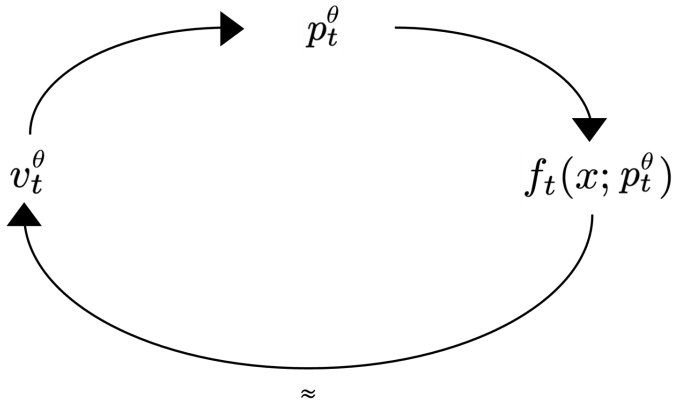
Symmetric f-divergence:

$$D_f(\rho_1 \parallel \rho_2) := \mathbf{E}_{X\sim\rho_2}\left[(\log \rho_1(X) - \log \rho_2(X))^2/2\right]$$

$$\mathbf{Sym}D_f(\rho_1, \rho_2) := D_f(\rho_1 \parallel \rho_2) + D_f(\rho_2 \parallel \rho_1)$$

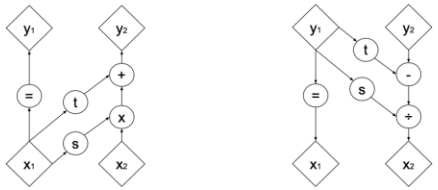
Conclusion

- Self-consistency:



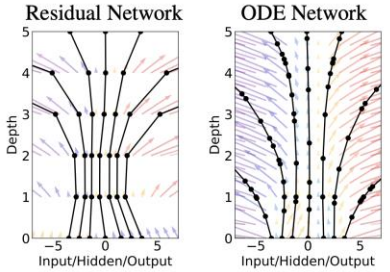
- Parametrization:

Normalizing Flows



Source: https://arxiv.org/abs/1605.08803
[Dinh et al. 2016]

Neural ODE



Source: https://arxiv.org/pdf/1806.07366
[Chen et al. 2019]

$$\theta_{k+1} := \theta_k - \eta \nabla_\theta|_{\theta=\theta_k} F(\theta, \theta_k),$$

- SCVM:

$$F(\theta, \theta_k) := \int_0^T \mathbf{E}_{X \sim \mu_t^{\theta_k}} \left[\left\| v_t^\theta(X) - f_t(X; p_t^{\theta_k}) \right\|^2 \right] \mathrm{d}t.$$

Conclusion

$$\theta_{k+1} := \theta_k - \eta \nabla_\theta|_{\theta=\theta_k} F(\theta, \theta_k),$$

- SCVM:

$$F(\theta, \theta_k) := \int_0^T \mathbf{E}_{X \sim \mu_t^{\theta_k}} \left[\left\| v_t^\theta(X) - f_t(X; p_t^{\theta_k}) \right\|^2 \right] \mathrm{d}t.$$

Conclusion

$$\theta_{k+1} := \theta_k - \eta \nabla_\theta|_{\theta=\theta_k} F(\theta, \theta_k),$$

- SCVM:

$$F(\theta, \theta_k) := \int_0^T \mathbf{E}_{X \sim \mu_t^{\theta_k}} \left[\left\| v_t^\theta(X) - f_t(X; p_t^{\theta_k}) \right\|^2 \right] \mathrm{d}t.$$

  - Discretization free
  - Lower training time
  - Scales well to higher dimensions

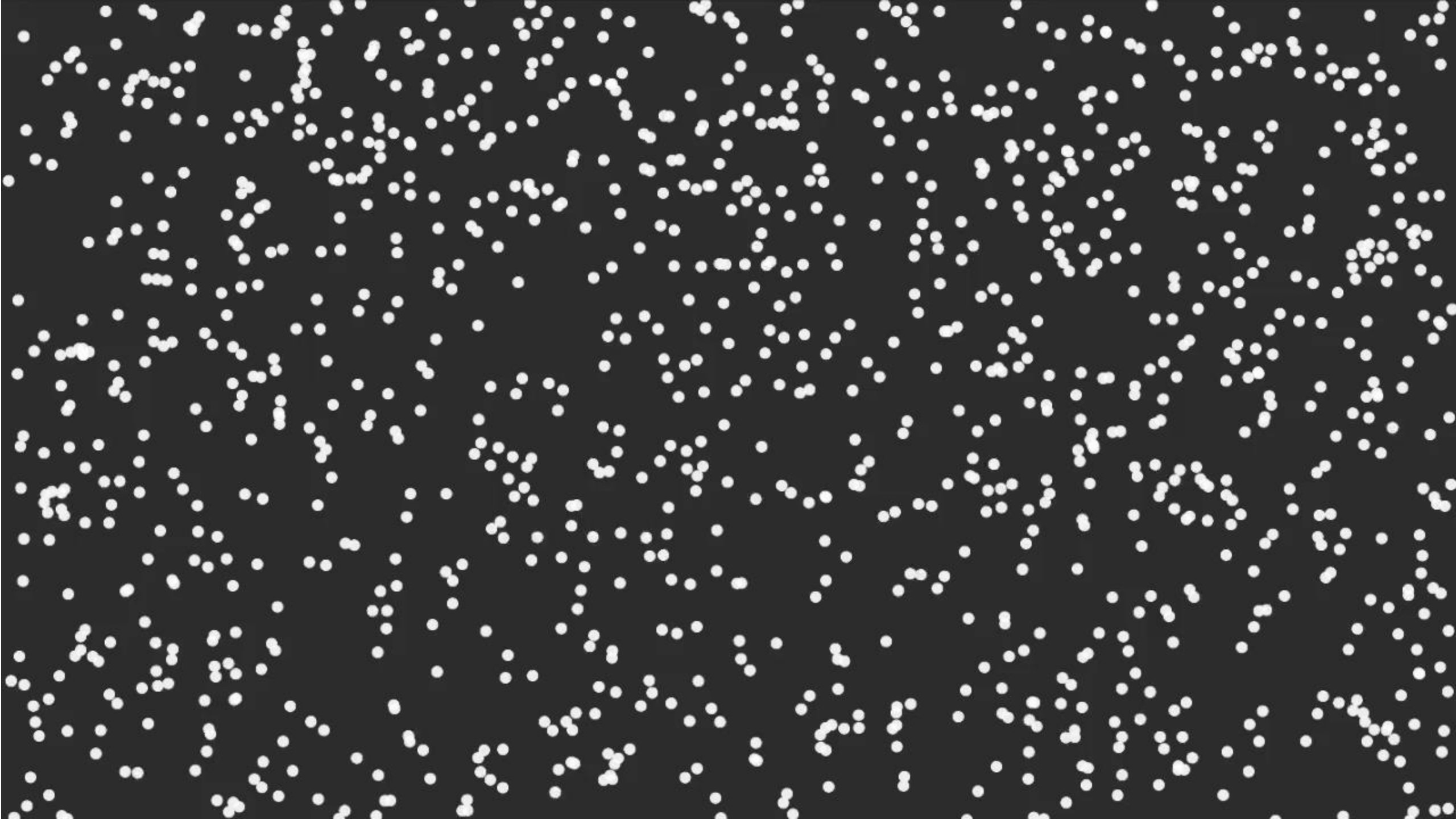
Conclusion

$$\theta_{k+1} := \theta_k - \eta \nabla_\theta|_{\theta=\theta_k} F(\theta, \theta_k),$$

- SCVM:

$$F(\theta, \theta_k) := \int_0^T \mathbf{E}_{X \sim \mu_t^{\theta_k}} \left[\left\| v_t^\theta(X) - f_t(X; p_t^{\theta_k}) \right\|^2 \right] \mathrm{d}t.$$

  - Discretization free
  - Lower training time
  - Scales well to higher dimensions
  - No theoretical guarantees



Additional Slides

B Integration-by-Parts Trick

This is a common trick used in score-matching literature [Hyvärinen and Dayan, 2005].

Proof of Proposition 3.1. Fix $t > 0$. The form of f_t in (4) is

$$f_t(x; \mu_t) = b_t(x) - D_t(x)\nabla \log p_t(x).$$

Hence

$$\mathbf{E}_{X\sim\mu_t^{\theta'}}\left[v_t^\theta(X)^\top f_t(X;\mu_t^{\theta'})\right] = \mathbf{E}_{X\sim\mu_t^{\theta'}}\left[v_t^\theta(X)^\top b_t(X)\right] - \mathbf{E}_{X\sim\mu_t^{\theta'}}\left[v_t^\theta(X)^\top D_t(X)\nabla \log p_t^{\theta'}(X)\right]$$

The second term can be written as

$$\begin{aligned}
\mathbf{E}_{X\sim\mu_t^{\theta'}}\left[v_t^\theta(X)^\top D_t(X)\nabla \log p_t^{\theta'}(X)\right] &= \int v_t^\theta(x)^\top D_t(x)\nabla \log p_t^{\theta'}(x)\,\mathrm{d}p_t^{\theta'}(x) \\
&= \int v_t^\theta(x)^\top D_t(x)\nabla p_t^{\theta'}(x)/p_t^{\theta'}(x)\cdot p_t^{\theta'}(x)\,\mathrm{d}x \\
&= \int v_t^\theta(x)^\top D_t(x)\nabla p_t^{\theta'}(x)\,\mathrm{d}x \\
&= -\int \boldsymbol{\nabla}\cdot\left(D_t(x)^\top v_t^\theta(x)\right)p_t^{\theta'}(x)\,\mathrm{d}x \\
&= -\mathbf{E}_{X\sim\mu_t^{\theta'}}\left[\boldsymbol{\nabla}\cdot\left(D_t(X)^\top v_t^\theta(X)\right)\right],
\end{aligned}$$

where we use integration-by-parts to get the second last equation and the assumption that v_t^θ, D_t are bounded and $p_t^\theta(x) \to 0$ as $\|x\| \to \infty$. □