# Fluid Dynamic Models in Machine Learning

Daniel D. Lee

Yung-Kyun Noh

CORNELL TECH

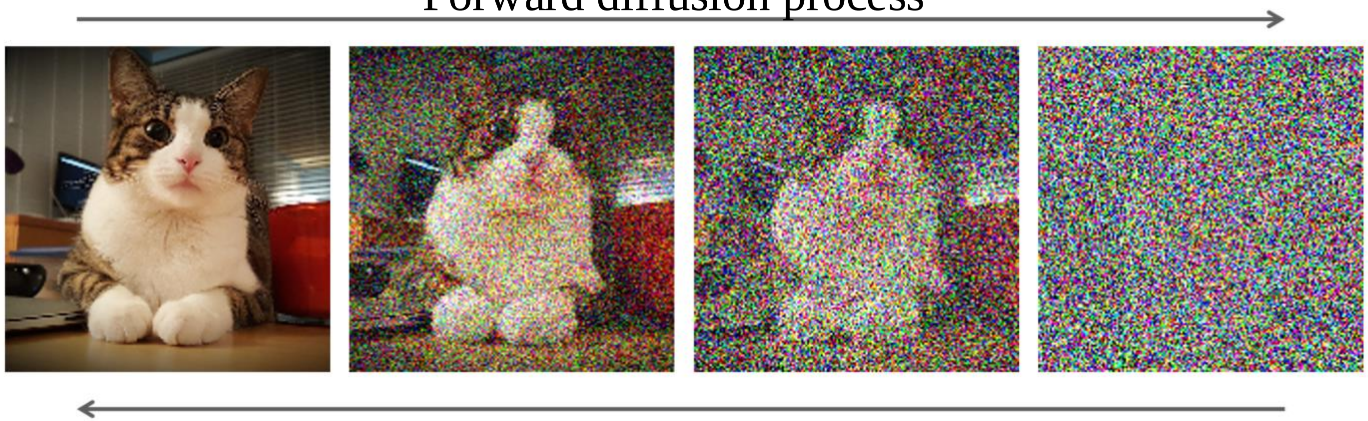
KIAS
KOREA INSTITUTE FOR ADVANCED STUDY

# Outline

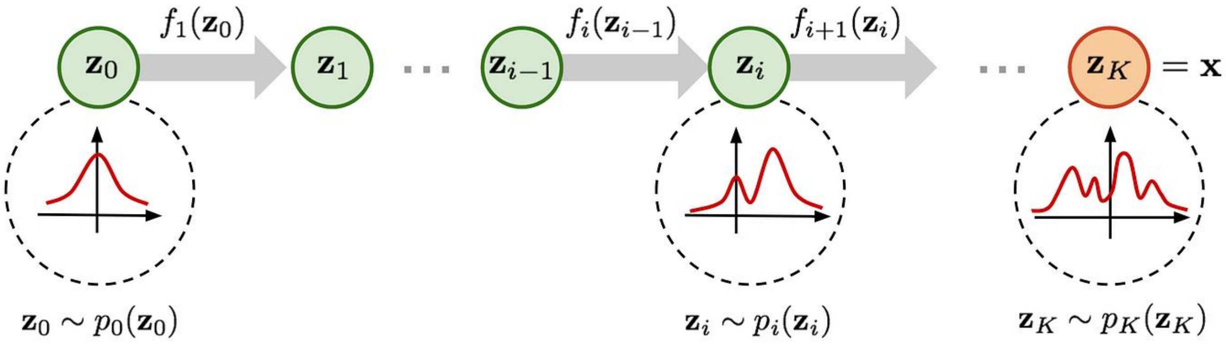
- Generative AI motivation
- Fluid dynamics and conservation laws
- Fokker-Planck equation and reverse diffusion
- Supervised learning and discriminant analysis
- Learning dynamics of neural weights

# Success of Generative AI

Forward diffusion process

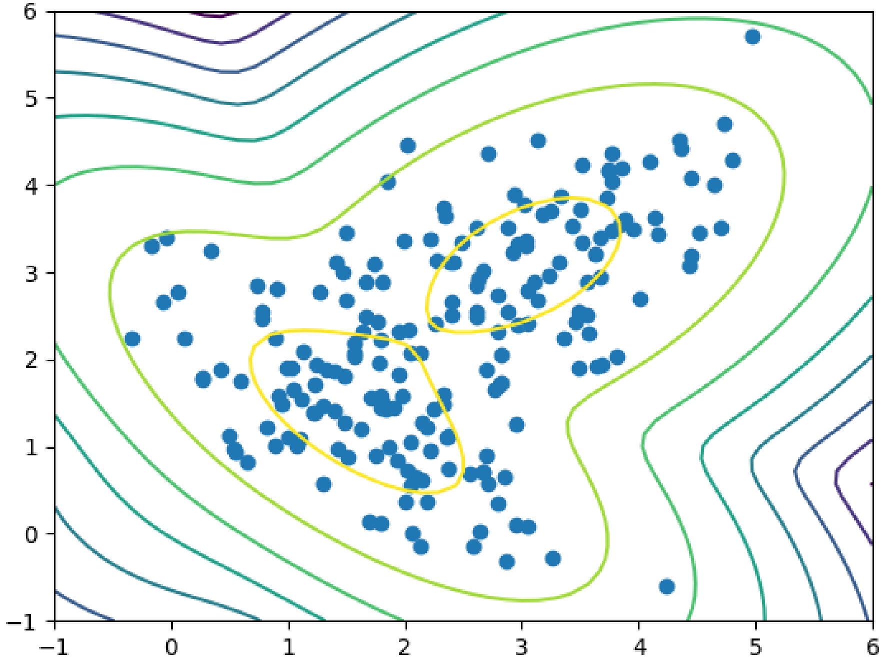


Reverse diffusion process




Normalizing flow models

# Density estimation



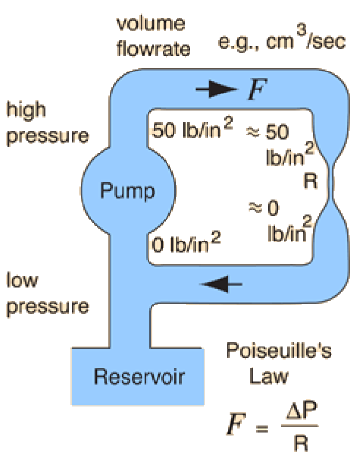

$$\vec{x} \in \mathbb{R}^N$$

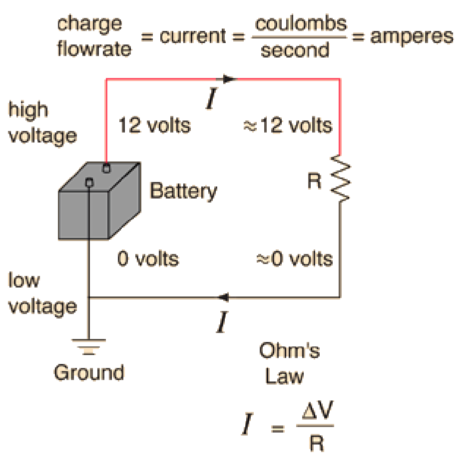
Probability density function

$$p(\vec{x}) \geq 0$$

- Many ML algorithms are methods for density estimation, e.g. the probability density of natural images in pixel space

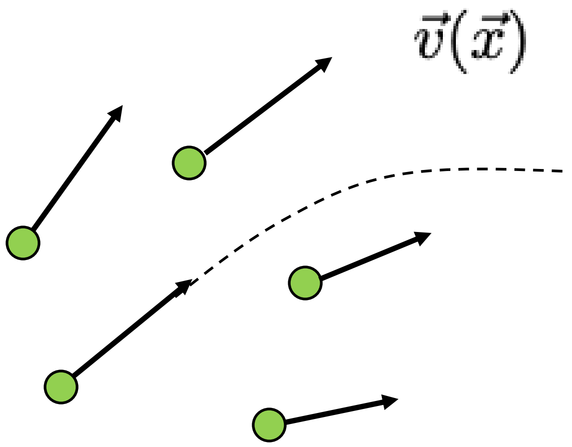
# Fluid models




- Continuum models of the motions of many particles, e.g. hydrodynamics and traffic models

# Flows




Velocity flow field

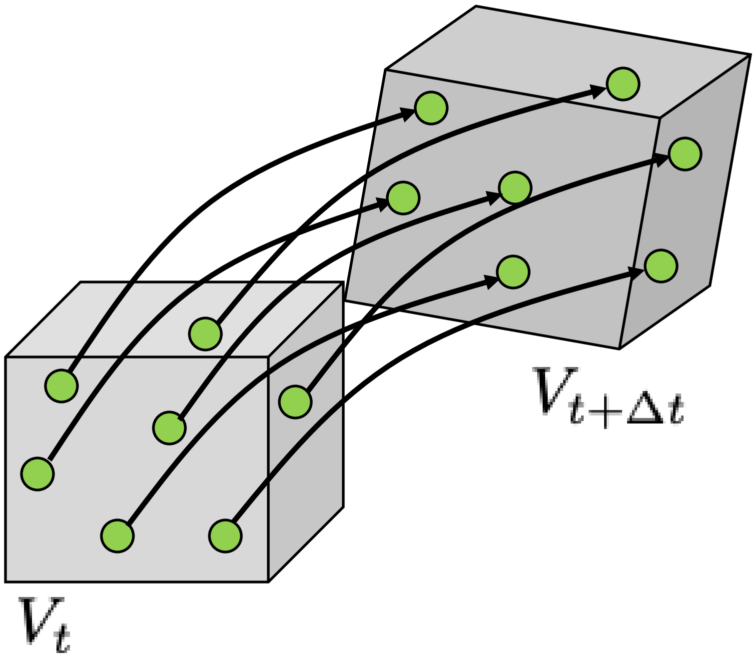
$\vec{v} \in \mathbb{R}^N$

Particle motion described by differential equation:

$$d\vec{x}_t = \vec{v}(\vec{x}_t)dt$$

- Dynamical motion of particles described by velocity flows and differential equations

# Conservation of mass




Number of particles enclosed in moving volume

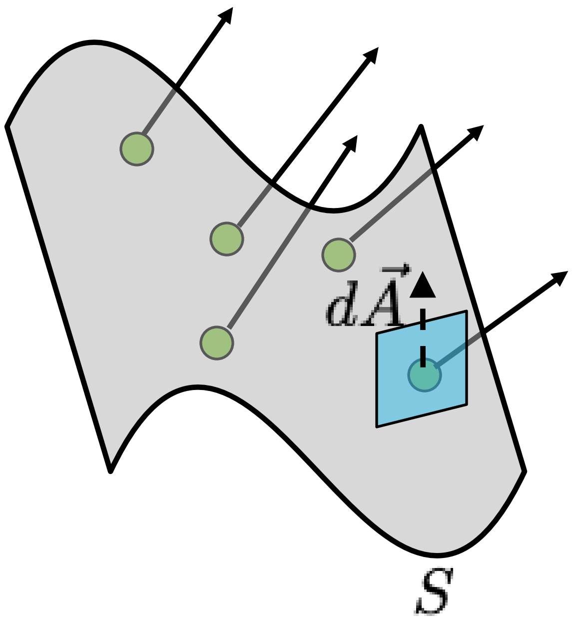
$$\frac{d}{dt}\int_{V_t} p(x)dx = 0$$

Total number of particles is conserved

$$\int_{-\infty}^{+\infty} p(x)dx = 1$$

- With no sources or sinks, particles are neither created nor destroyed

# Flux




Flux vector field is
current flow per unit area

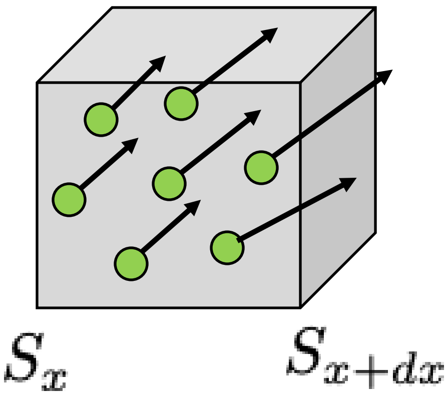
$$\vec{J}(x) = p(x)\vec{v}(x)$$

Particle flow per unit time
across surface

$$\iint_S \vec{J}(x) \cdot d\vec{A}$$

- Flux measures the rate of particle flow across surfaces

# Continuity equation

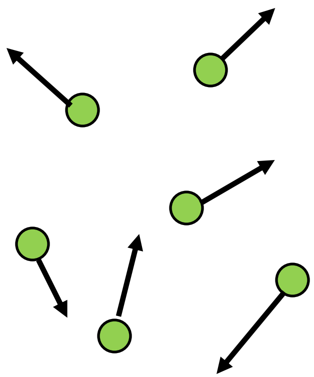



$$\frac{\partial p}{\partial t} + \nabla \cdot \vec{J} = 0$$

$$\nabla \cdot \vec{J} = \frac{J_x(x+dx) - J_x(x)}{dx} + \frac{J_y(y+dy) - J_y(y)}{dy} + \frac{J_z(z+dz) - J_z(z)}{dz}$$

- Green’s theorem relates total flux to temporal change in density within volume

# Diffusion



Random white noise $\vec{\eta}_t$

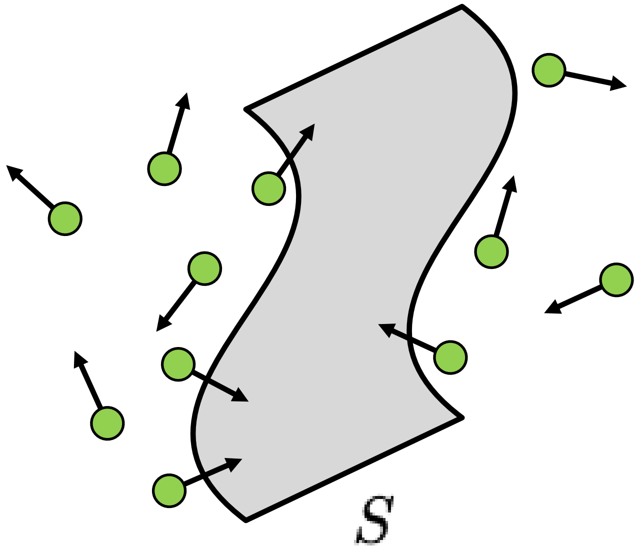
$$\langle \vec{\eta}_t \rangle = 0 \quad \langle \vec{\eta}_t \vec{\eta}_{t'}^\top \rangle = \delta_{tt'} I_{N \times N}$$

Stochastic differential equation:

$$d\vec{x}_t = \epsilon d\vec{\eta}_t$$

- Process of adding Brownian random noise to particle motion

# Diffusion equation




Flux from diffusion

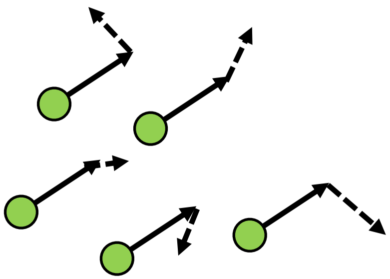
$$\vec{J}_D = -\frac{1}{2}\epsilon^2 \nabla p(\vec{x})$$

Diffusion equation:

$$\frac{\partial p}{\partial t} = -\nabla \cdot \vec{J}_D = \frac{1}{2}\epsilon^2 \nabla^2 p$$

- Diffusion causes net flow from regions of high density to low density

# Fokker-Planck equation



Drift and diffusion:

$$d\vec{x}_t = \vec{v}(\vec{x}_t)dt + \epsilon d\vec{\eta}_t$$

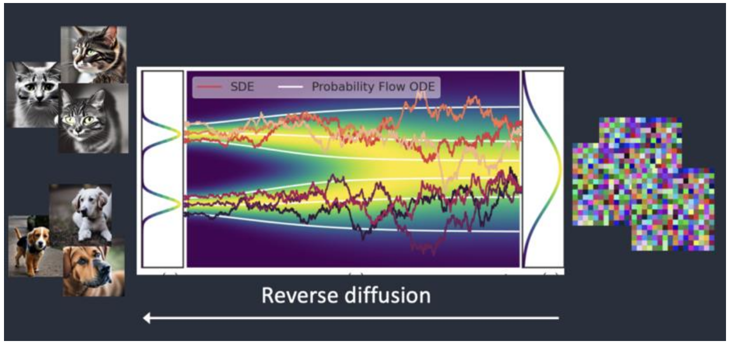
Fokker-Planck:

$$\frac{\partial p_t}{\partial t} = -\nabla \cdot \vec{J}_v - \nabla \cdot \vec{J}_D$$
$$= -\nabla \cdot (p_t \vec{v}_t) + \frac{1}{2}\epsilon^2 \nabla^2 p_t$$

- Continuity equation with drift velocity and diffusion

# Reverse diffusion

$$\frac{\partial p_t}{\partial t} = -\nabla \cdot (p_t \vec{v}_t) + \frac{1}{2}\epsilon^2 \nabla^2 p_t$$



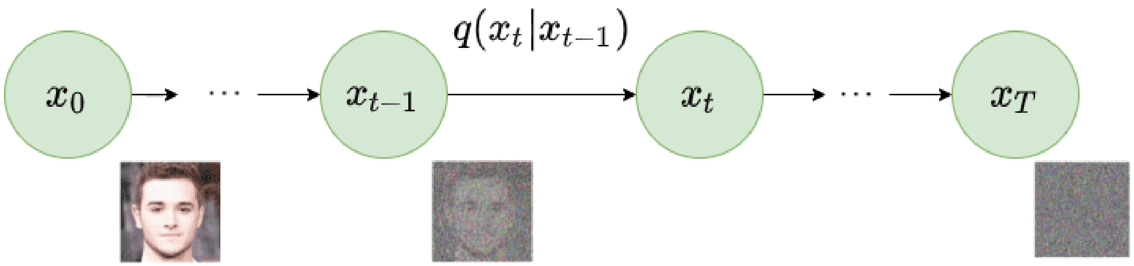

Score function: $\vec{v}_t = \nabla \log p_t$

Reverse diffusion Laplacian term:

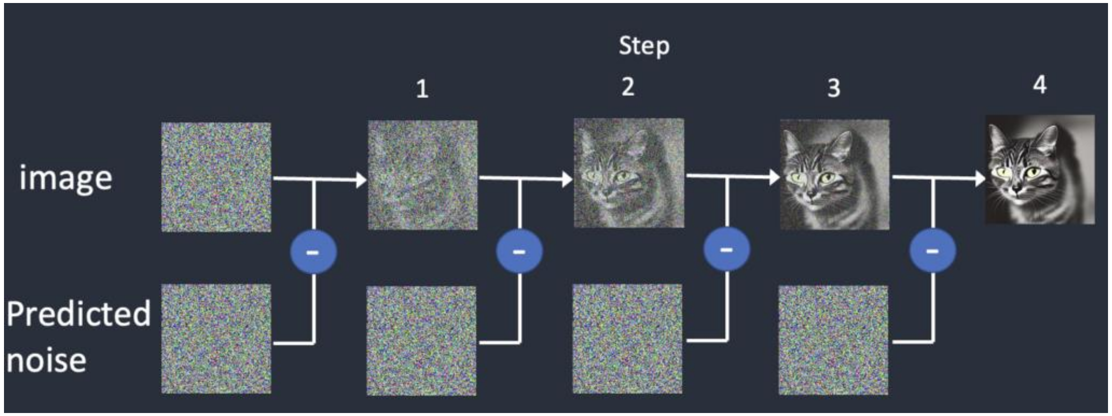
$$-\nabla \cdot (p_t \vec{v}_t) = -\nabla^2 p_t$$

- Velocity flow field with score function will counteract diffusive flux

# Diffusion models




Ho, et. al. (2020)




- Learn score function with neural network from forward diffusion process

# Ornstein-Uhlenbeck process

Process with parameter: $0 \leq \sigma \leq 1$

Base distribution $P_{\sigma=0}(\vec{x}) \quad \longrightarrow \quad P_{\sigma=1}(\vec{x}) = \mathcal{N}(0, I)$

$$dx_t = -\frac{\sigma}{1+\sigma} x_t dt + \sqrt{\frac{2\sigma}{1+\sigma}} d\eta_t \qquad \sigma(t) = 1 - e^{-t}$$

$$P_\sigma(\vec{x}) = \int d\vec{x}' \, P_0(\vec{x}') \mathcal{N}_{\sigma^2}(\vec{x} - \sqrt{1-\sigma^2}\vec{x}')$$

- Forward Ornstein-Uhlenbeck process model

# Reverse diffusion

Score function: $$\nabla \log P_\sigma(\vec{x}) = -\frac{1}{\sigma^2}\vec{x} + \frac{\sqrt{1-\sigma^2}}{\sigma^2}\langle \vec{x}' | \vec{x} \rangle_\sigma$$

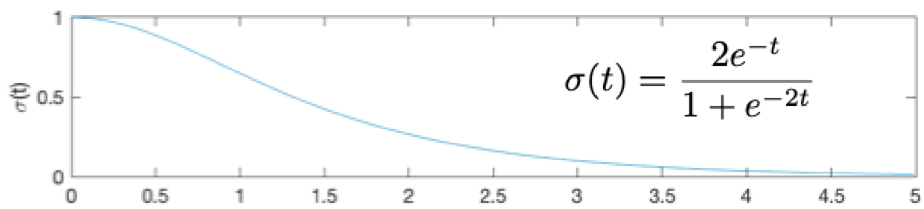
$$\langle \vec{x}' | \vec{x} \rangle_\sigma = \frac{1}{P_\sigma(\vec{x})} \int d\vec{x}' \, \vec{x}' P_0(\vec{x}') \mathcal{N}_{\sigma^2}(\vec{x} - \sqrt{1-\sigma^2}\vec{x}')$$

$$\frac{\partial P_\sigma}{\partial \sigma} = \frac{1}{\sigma} \nabla \cdot \left[ -P_\sigma \vec{x} + \frac{P_\sigma}{\sqrt{1-\sigma^2}} \langle \vec{x}' | \vec{x} \rangle_\sigma \right]$$

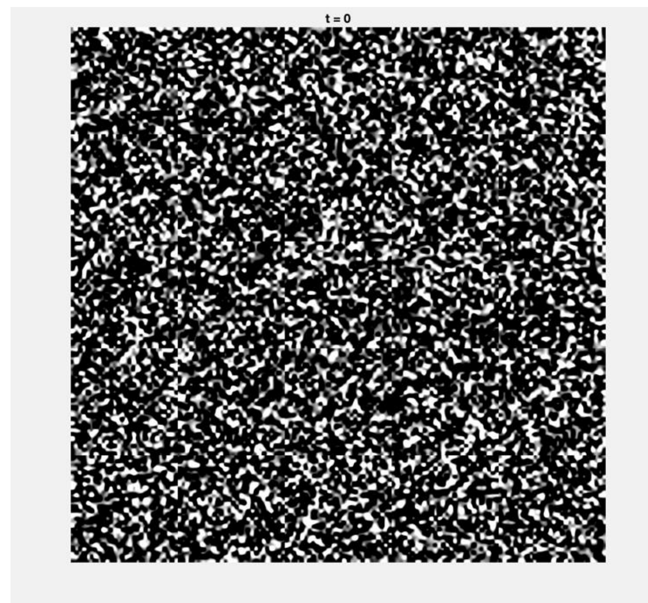
- Score function for reverse diffusion can be written as expectation of mixture distribution

# Reverse diffusion schedule

$$\frac{d\sigma}{dt} = -\sigma\sqrt{1-\sigma^2}$$




$$\vec{v}_t(\vec{x}) = -\sqrt{1-\sigma^2}\,\vec{x} + \langle \vec{x}' | \vec{x} \rangle_\sigma$$




- Alternative reverse diffusion schedule with slow initial descent

# Gauge freedom

$$\frac{\partial p_t}{\partial t} = -\nabla \cdot (p_t \vec{v}_t)$$

Helmholtz decomposition: $\vec{v}_t = \nabla \log p_t + \frac{1}{p_t} \nabla \times \vec{A}_t(\vec{x})$

Divergence-free term: $\nabla \cdot (\nabla \times \vec{A}_t) = 0$

- Freedom to add any divergence-free velocity-field from vector potential term

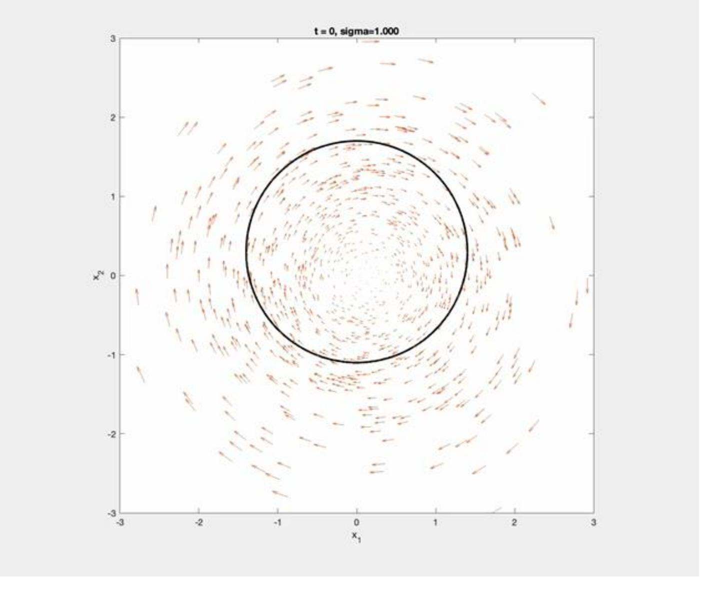
# Reverse diffusion with curl

Rotational velocity from score function:

$$\vec{v}_t = (I + A)\nabla \log p_t$$

Antisymmetric matrix gauge freedom:

$$A_{ij} = -A_{ji}$$



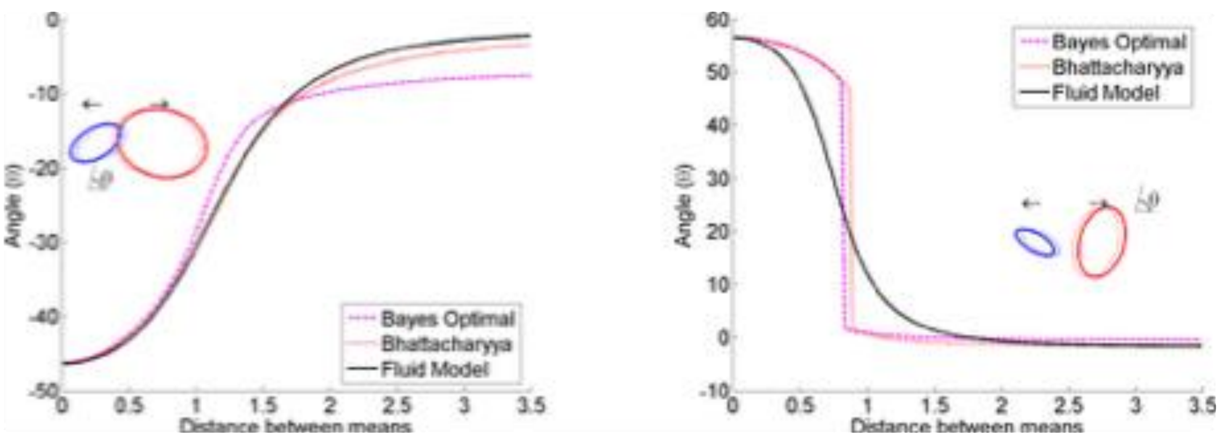

- Adding rotational dynamics to reverse diffusion onto a low-dimensional manifold

# Supervised learning

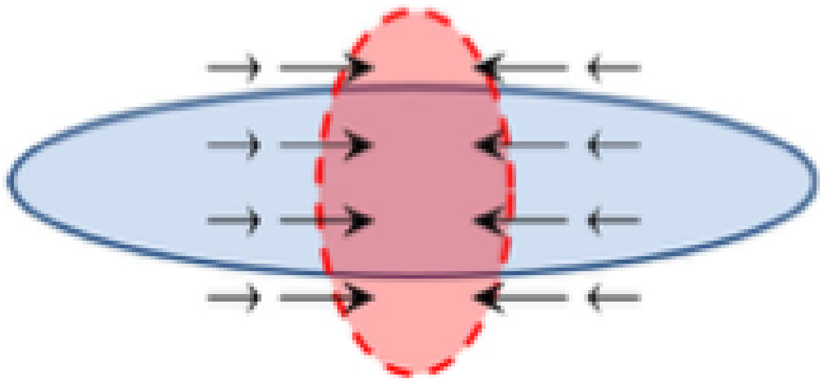
## Fluid Dynamic Models for Bhattacharyya-Based Discriminant Analysis

Yung-Kyun Noh, Jihun Hamm, Frank Chongwoo Park, *Fellow, IEEE*,
Byoung-Tak Zhang, and Daniel D. Lee, *Fellow, IEEE*

IEEE Trans. Pattern Analysis and Machine Intelligence, 2018



# Two interacting fluids

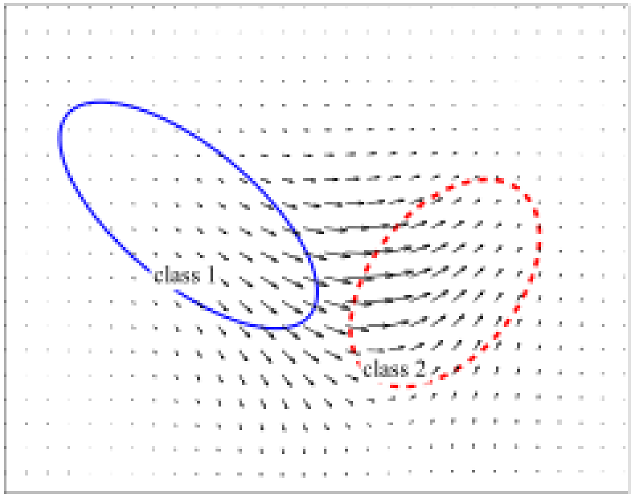


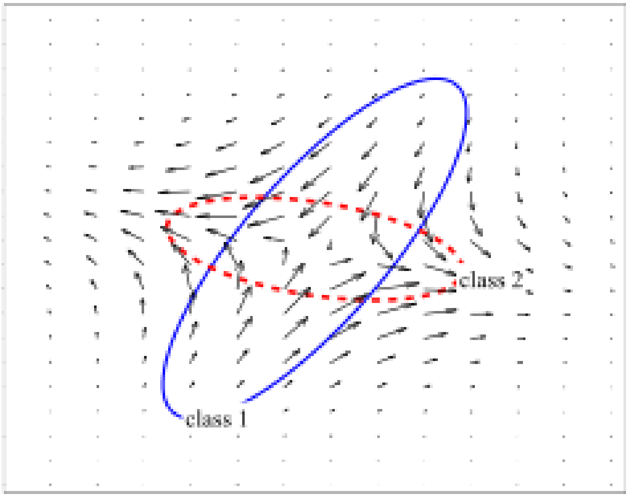
Energy based upon
Bhattacharyya coefficient:

$$U(p_1, p_2) = \int \sqrt{p_1(\vec{x})p_2(\vec{x})}\, d\vec{x}$$

- What is the induced flow from minimizing the potential energy between two fluids?

# Flow fields




Force field on fluids: $F_2(\vec{x}) = -\frac{1}{2} p_2 \nabla \sqrt{\frac{p_1}{p_2}}$

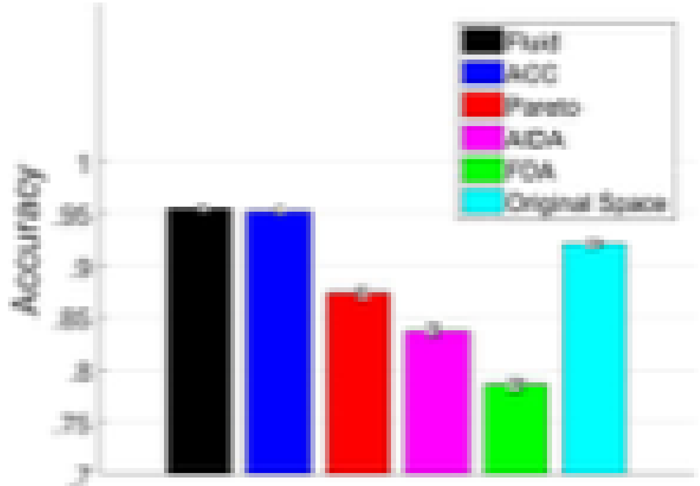
Newton's second law: $F_1(\vec{x}) = -F_2(\vec{x})$

- Induced flow field derived from equation of continuity

# Discriminant analysis

Flow correlation matrix:

$$\sum_c \langle F_c x_c^\top \rangle \langle x_c x_c^\top \rangle_{p_c} \langle x_c F_c^\top \rangle$$




- Optimal low-dimensional projection defined by principal eigenvectors of flow fields

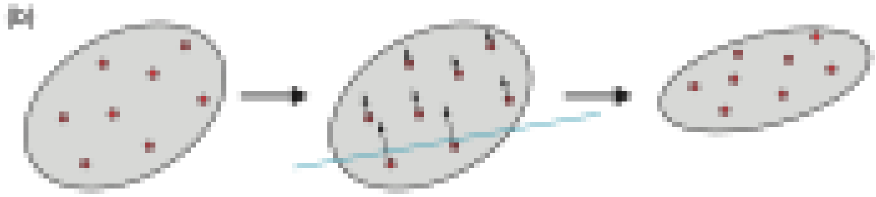
# Learning dynamics

## Learning via Gaussian Herding

**Koby Crammer**
Department of Electrical Enginering
The Technion
Haifa, 32000 Israel
koby@ee.technion.ac.il

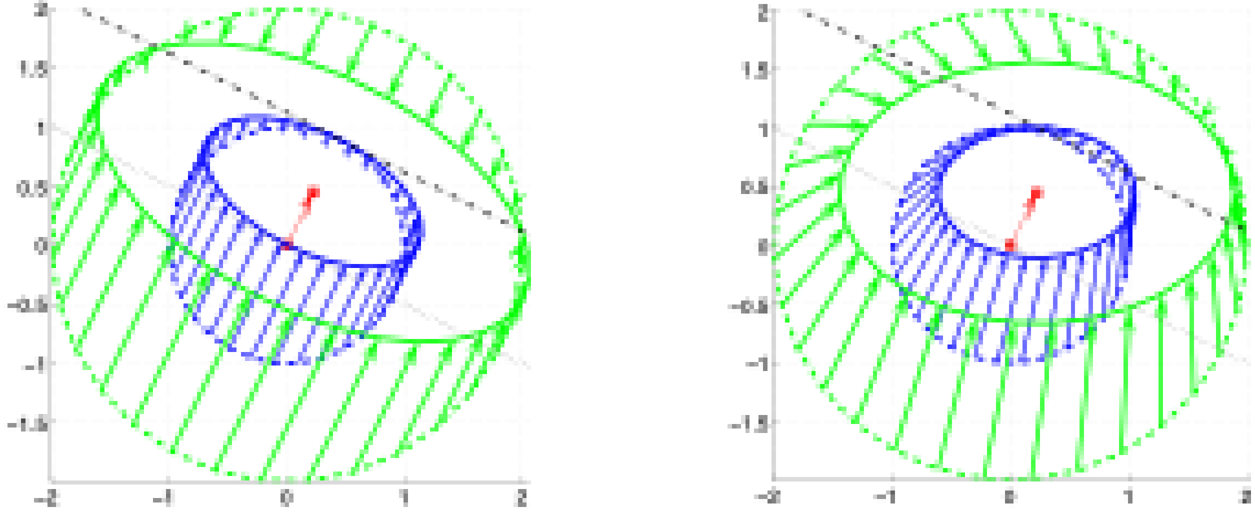
**Daniel D. Lee**
Dept. of Electrical and Systems Engineering
University of Pennsylvania
Philadelphia, PA 19104
ddlee@seas.upenn.edu

NeurIPS, 2010



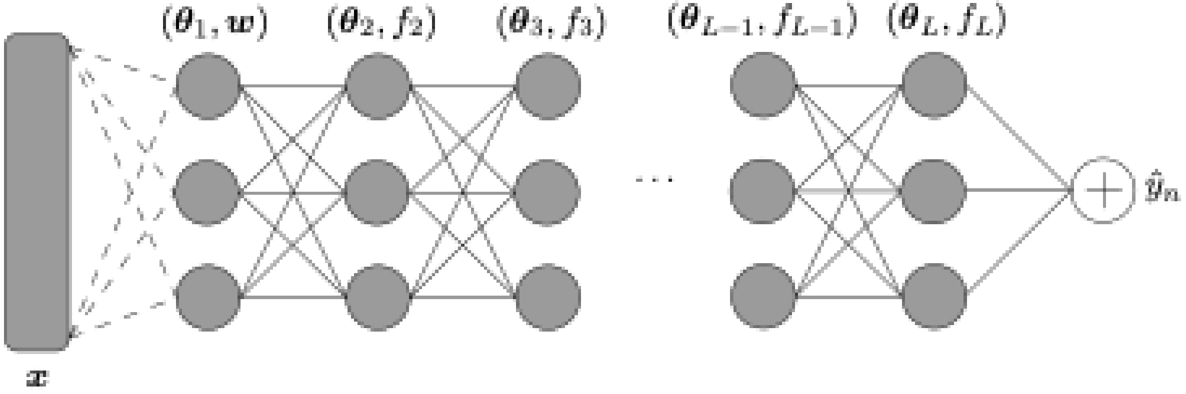
# Online learning

$$w_{i+1} = \arg\min_{w} \frac{1}{2}|w - w_i|^2 + \lambda \ell((x_i, y_i), w)$$




- Tracking flow of Gaussian weight distribution during online learning

# Multilayer neural networks




$$\hat{y}(\vec{x}) = \int p(\vec{w})\sigma(\vec{x}, \vec{w})d\vec{w}$$

- Analyzing weight distributions during learning in multilayer neural networks

# SGD dynamics

$$w_i^{k+1} = w_i^k - \eta(y_k - \hat{y}(x_k))\nabla_w \sigma(x_k; w)$$

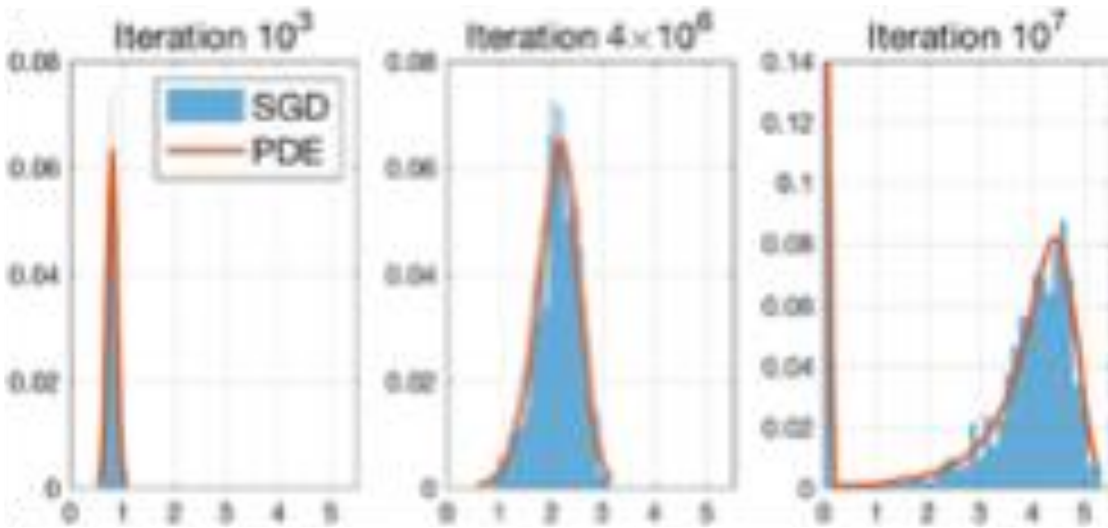
$$\frac{\partial p_t(\vec{w})}{\partial t} = -\nabla \cdot [p_t \nabla \Psi(\vec{w}; p_t)]$$

$$\Psi(\vec{w}; p_t) = V(\vec{w}) + \int U(\vec{w}, \vec{w}')p(\vec{w}')d\vec{w}'$$

Mei, Montanari, Nguyen (PNAS 2018)

- Mean-field flow-based analysis of stochastic gradient descent

# Mean field predictions

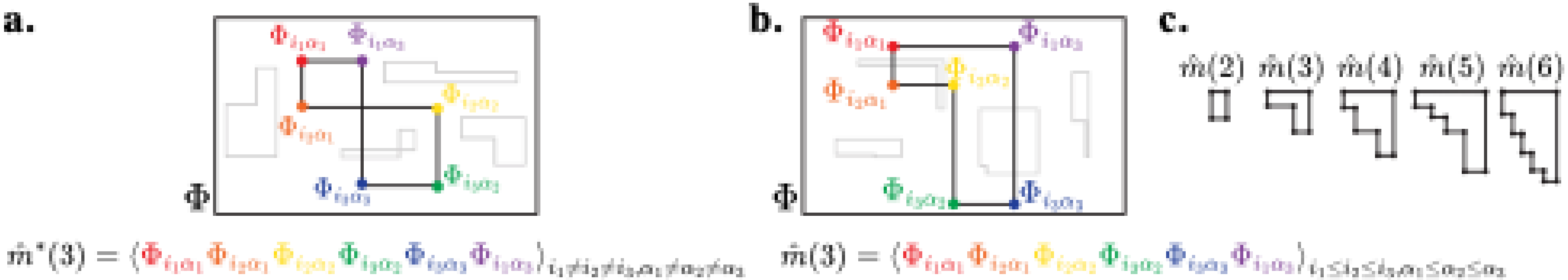



$$v_t = -\nabla\Psi(\vec{w}; p_t)$$

Mei, Montanari, Nguyen (PNAS 2018)

- PDE accurately predicts convergence of stochastic gradient descent

# Population estimates




Chun, Chung, Lee (preprint, 2024)

$$k(x, x') = \int dw \, p(w) \phi(x, w) \phi(x', w)$$

$$T_k f = \int dx \, p(x) k(\cdot, x) f(x)$$

- Estimating kernel integral operator properties from finite measurement matrices

# Summary

- Continuum fluid descriptions in AI
- Fokker-Planck equation describes time evolution of density functions
- Reverse diffusion dynamics
- Fluid models for learning dynamics in supervised learning
- Continuum population estimates from finite networks and sampling