Wasserstein Gradient Flow for Stochastic Prediction, Filtering, Learning and Control

Abhishek Halder

Department of Applied Mathematics
University of California, Santa Cruz
Santa Cruz, CA 95064

Joint work with Kenneth F. Caluya (UC Santa Cruz),
Tryphon T. Georgiou (UC Irvine), Walid Krichene (Google)

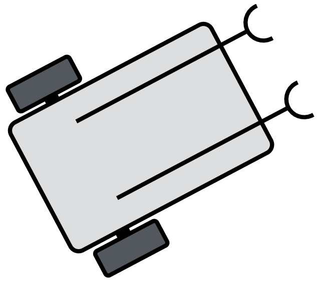
Controls, Autonomy and Robotics Seminar, UT Austin, TX, Nov. 18, 2020

Overarching Theme

Systems-control theory for densities

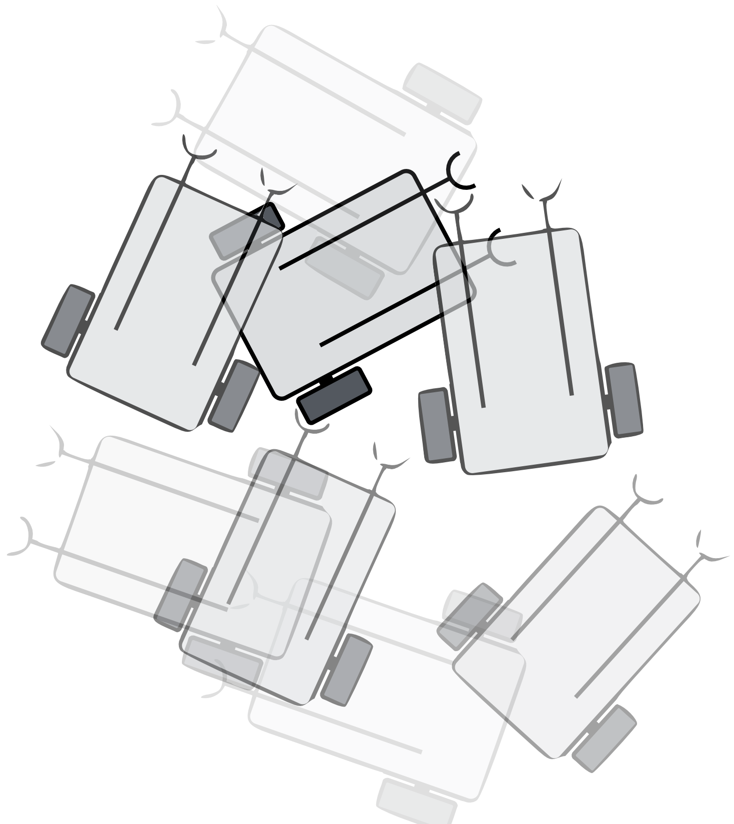
What is density?

Probability Density Fn.



$$\boldsymbol{x}(t) \in \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} \in \mathcal{X} \equiv \mathbb{R}^2 \times \mathbb{S}^1$$

Probability Density Fn.

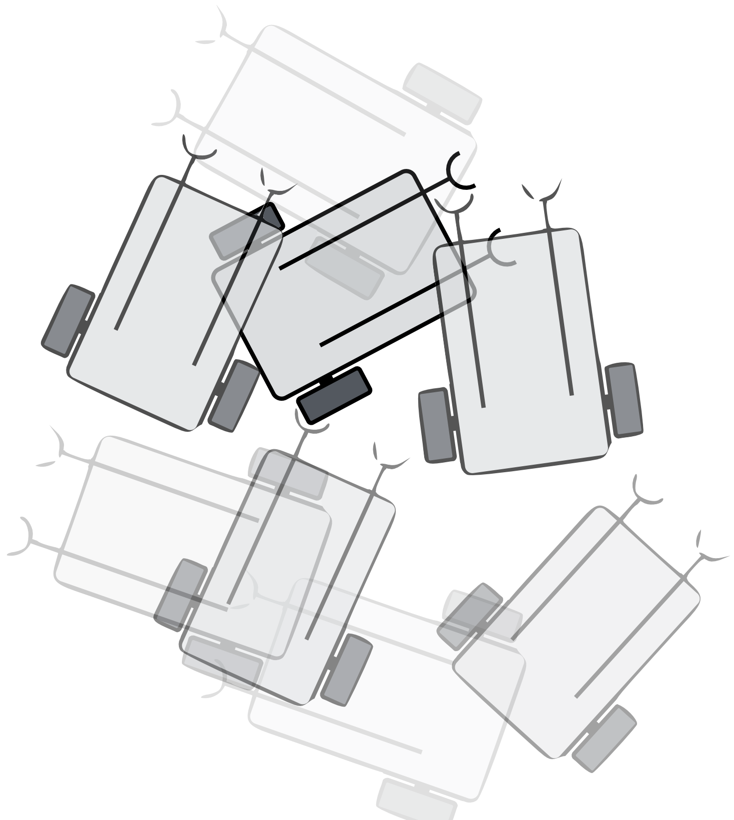


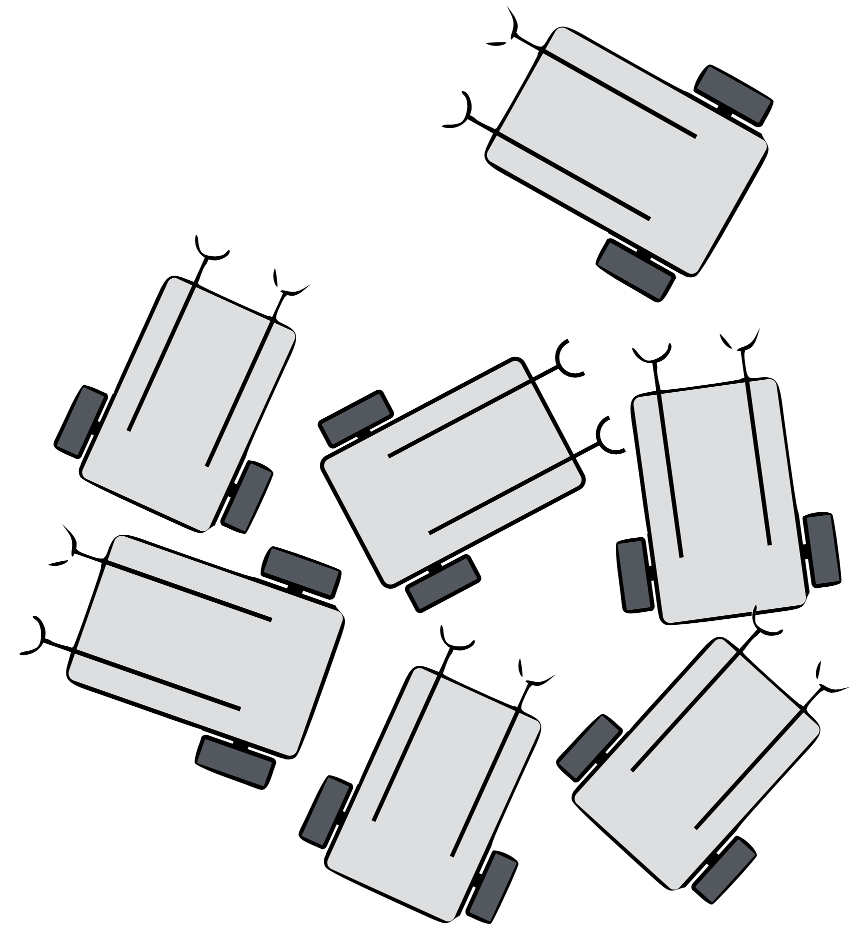
$$\boldsymbol{x}(t) \in \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} \in \mathcal{X} \equiv \mathbb{R}^2 \times \mathbb{S}^1$$

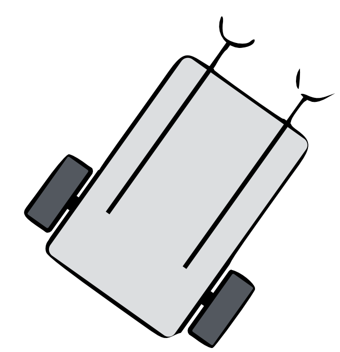
$$\rho\left(\boldsymbol{x}, t\right) : \mathcal{X} \times [0, \infty) \mapsto \mathbb{R}_{\geq 0}$$

$$\int_{\mathcal{X}} \rho \, \mathrm{d}\boldsymbol{x} = 1 \quad \text{for all } t \in [0, \infty)$$

Probability Density Fn.   Population Density Fn.







$$\boldsymbol{x}(t) \in \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} \in \mathcal{X} \equiv \mathbb{R}^2 \times \mathbb{S}^1$$

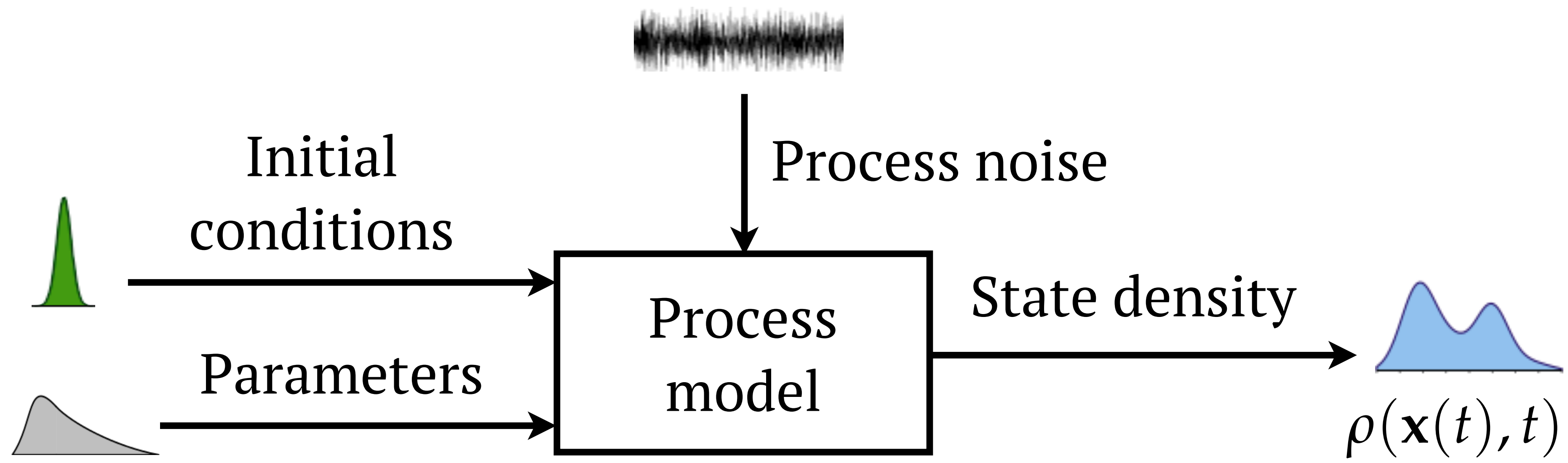
$$\rho\left(\boldsymbol{x}, t\right) : \mathcal{X} \times [0, \infty) \mapsto \mathbb{R}_{\geq 0}$$

$$\int_{\mathcal{X}} \rho \, \mathrm{d}\boldsymbol{x} = 1 \quad \text{for all } t \in [0, \infty)$$

Why care about densities?

Prediction Problem

Compute
joint state PDF
$\rho\left(\boldsymbol{x}, t\right)$



Trajectory flow:

$$\mathrm{d}\mathbf{X}(t) = \mathbf{f}(\mathbf{X}, t)\,\mathrm{d}t + \mathbf{g}(\mathbf{X}, t)\,\mathrm{d}\mathbf{w}(t), \quad \mathrm{d}\mathbf{w}(t) \sim \mathcal{N}(0, \mathbf{Q}\mathrm{d}t)$$

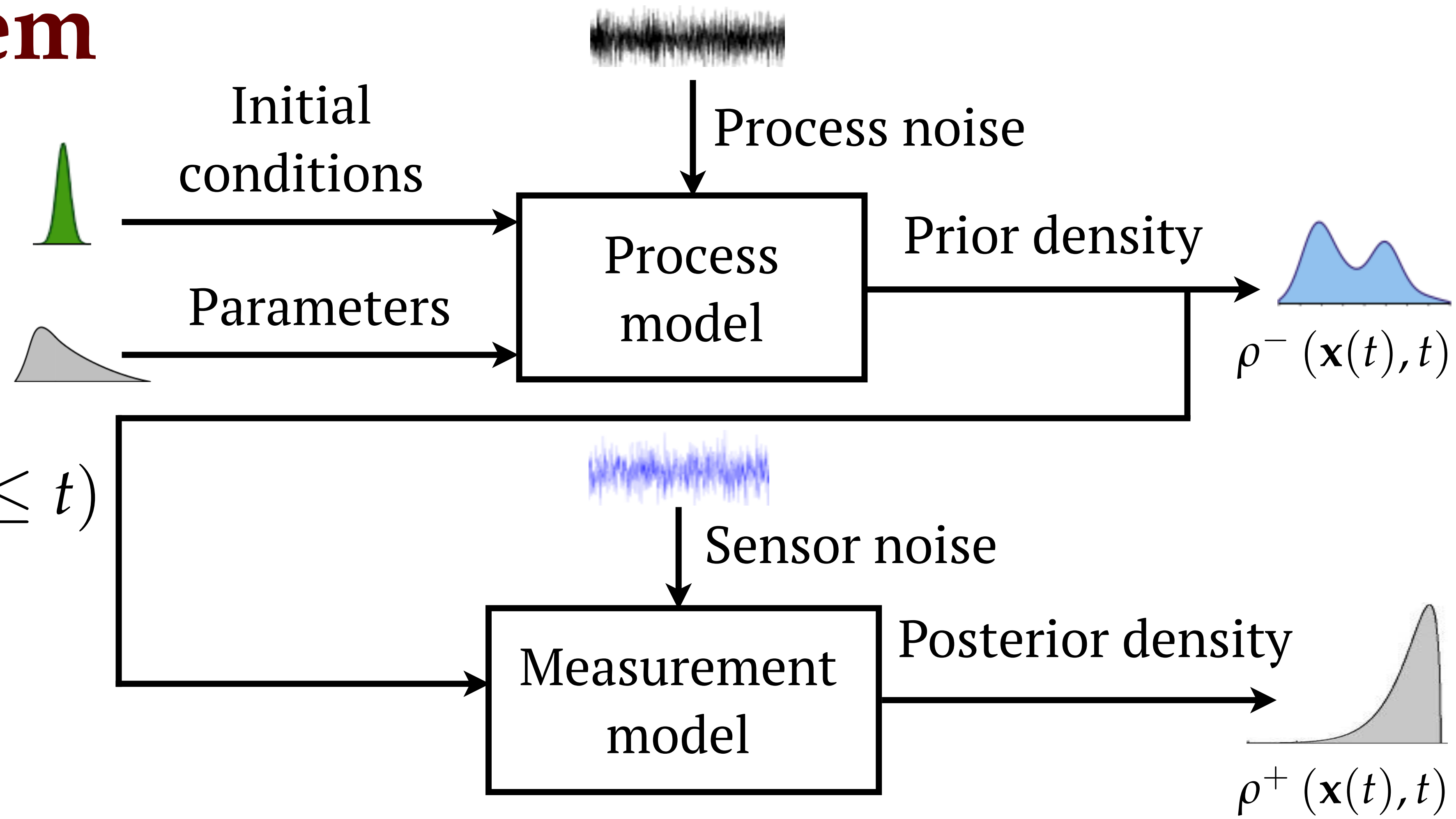
Density flow:

$$\frac{\partial \rho}{\partial t} = \mathcal{L}_{\mathrm{FP}}(\rho) := -\nabla \cdot (\rho \mathbf{f}) + \frac{1}{2} \sum_{i,j=1}^{n} \frac{\partial^2}{\partial x_i \partial x_j} \left(\left(\mathbf{g}\mathbf{Q}\mathbf{g}^{\top} \right)_{ij} \rho \right)$$

Filtering Problem

Compute conditional joint state PDF

$$\rho^{+} := \rho\left(x, t \mid z(s), 0 \leq s \leq t\right)$$



Trajectory flow:

$$\mathrm{d}\mathbf{X}(t) = \mathbf{f}(\mathbf{X}, t)\,\mathrm{d}t + \mathbf{g}(\mathbf{X}, t)\,\mathrm{d}\mathbf{w}(t), \qquad \mathrm{d}\mathbf{w}(t) \sim \mathcal{N}(0, \mathbf{Q}\mathrm{d}t)$$

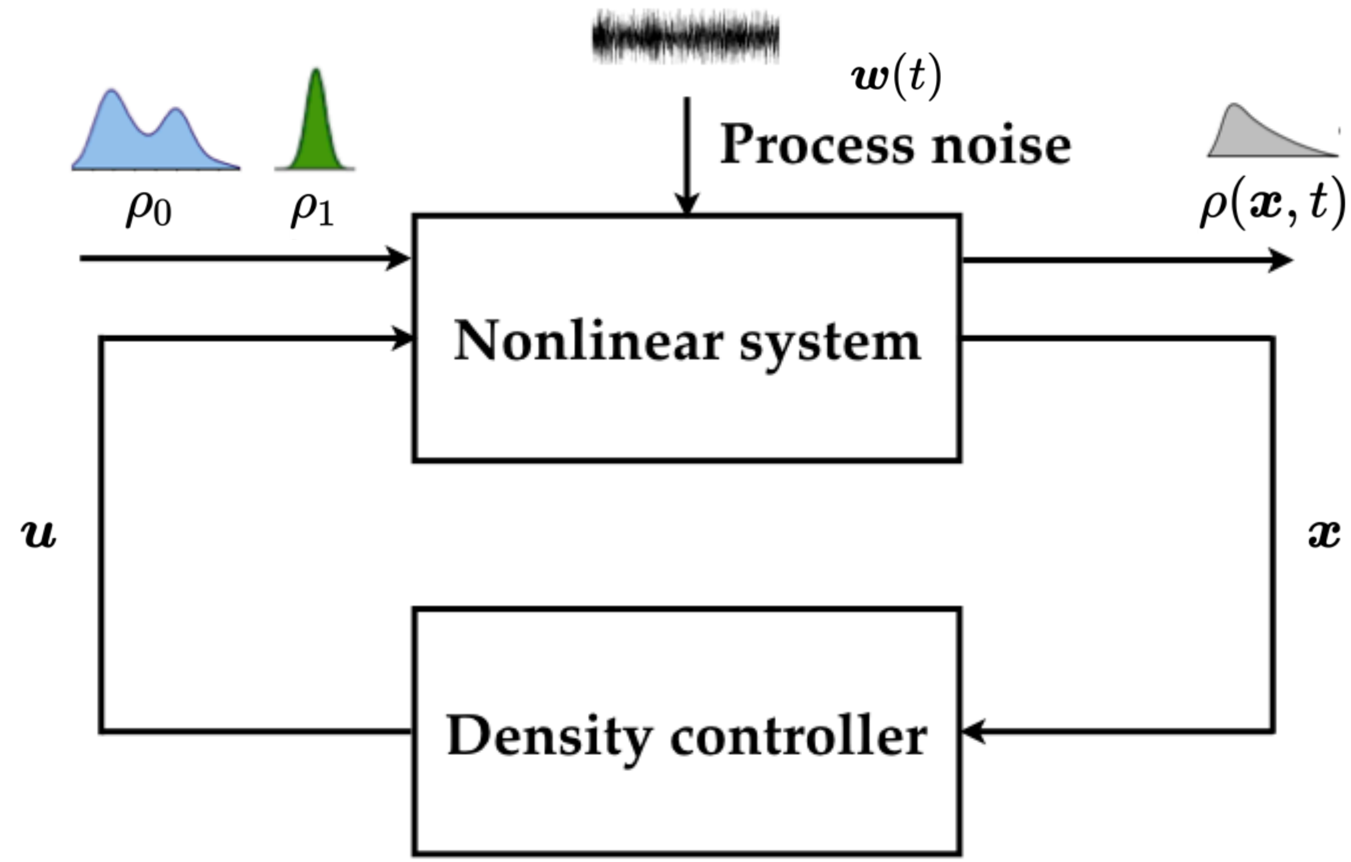
$$\mathrm{d}\mathbf{Z}(t) = \mathbf{h}(\mathbf{X}, t)\,\mathrm{d}t + \mathrm{d}\mathbf{v}(t), \qquad \mathrm{d}\mathbf{v}(t) \sim \mathcal{N}(0, \mathbf{R}\mathrm{d}t)$$

Density flow:

$$\mathrm{d}\rho^{+} = \left[\mathcal{L}_{\mathrm{FP}}\mathrm{d}t + \left(\mathbf{h}(\mathbf{x}, t) - \mathbb{E}_{\rho^{+}}\{\mathbf{h}(\mathbf{x}, t)\}\right)^{\top}\mathbf{R}^{-1}\left(\mathrm{d}\mathbf{z}(t) - \mathbb{E}_{\rho^{+}}\{\mathbf{h}(\mathbf{x}, t)\}\mathrm{d}t\right)\right]\rho^{+}$$

Control Problem

Steer joint state PDF via feedback control over finite time horizon



$$\underset{u \in \mathcal{U}}{\text{minimize}} \quad \mathbb{E}\left[\int_0^1 \|\boldsymbol{u}\|_2^2 \, \mathrm{d}t\right]$$

subject to

$$\mathrm{d}\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, t)\, \mathrm{d}t + \boldsymbol{g}(\boldsymbol{x}, t)\, \mathrm{d}\boldsymbol{w},$$

$$\boldsymbol{x}(t=0) \sim \rho_0, \quad \boldsymbol{x}(t=1) \sim \rho_1$$

Neural Network Learning Problem

Consider fully connected NN

Think "layers" as interacting population of neurons

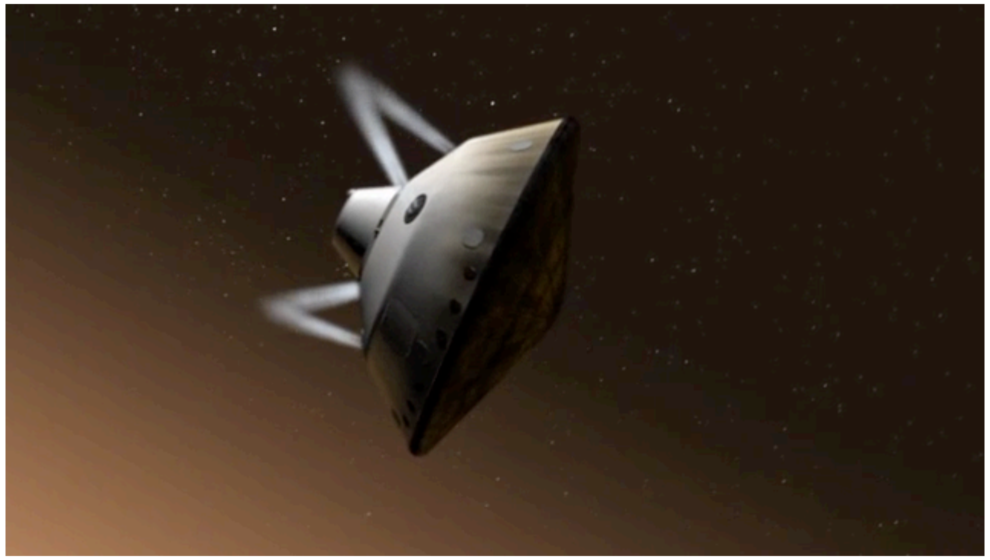
Mean field learning problem: $$\inf_{\rho \in \mathcal{P}_2(\mathbb{R}^p)} R\left(\int \Phi(\boldsymbol{x}, \boldsymbol{\theta})\rho(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}\right)$$

PDF dynamics:

$$\frac{\partial \rho}{\partial t} = -\nabla^{W} R\left(\int \Phi\rho\right) = \nabla \cdot \left(\rho \nabla \frac{\delta}{\delta \rho} R\left(\int \Phi\rho\right)\right)$$

PDFs in Mars Entry-Descent-Landing

Prediction problem



Predict heating rate uncertainty

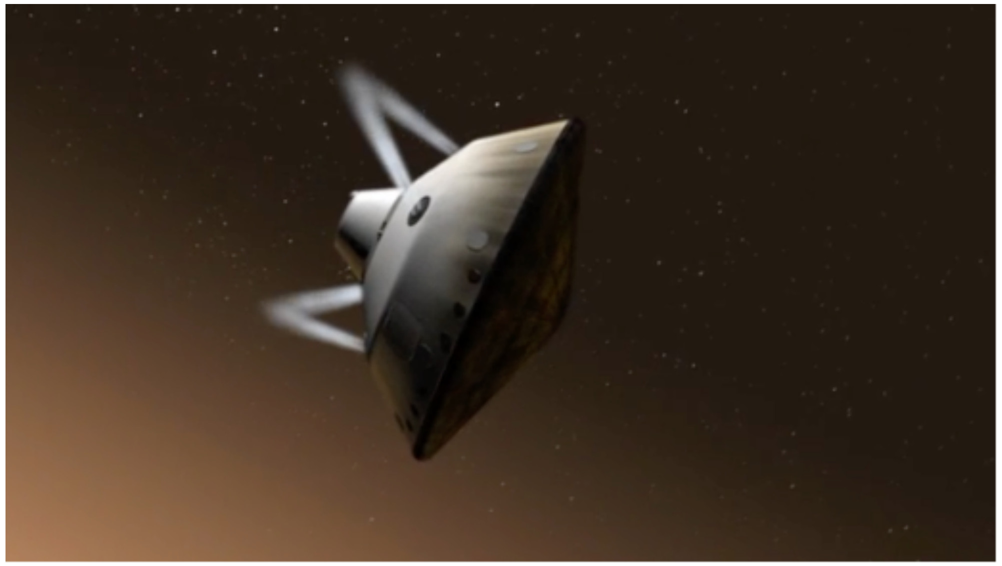
Filtering problem

Control problem

Learning problem

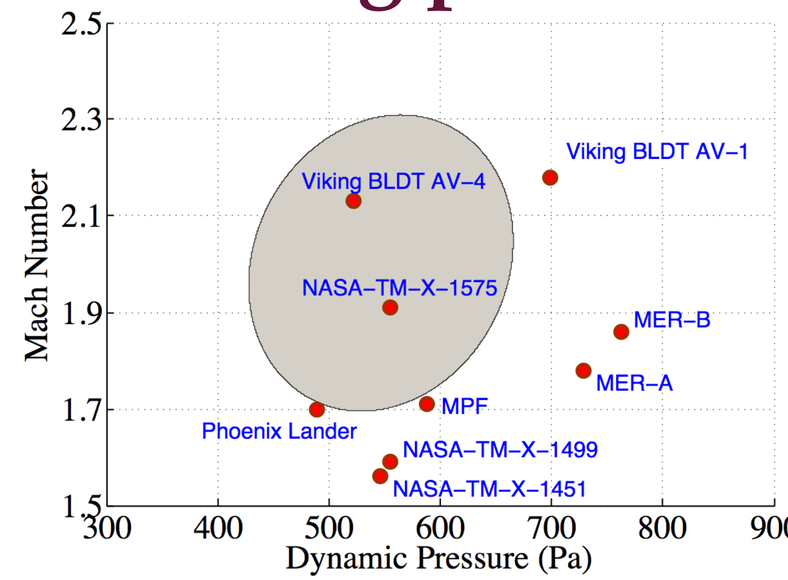
PDFs in Mars Entry-Descent-Landing

Prediction problem



Predict heating rate uncertainty

Filtering problem



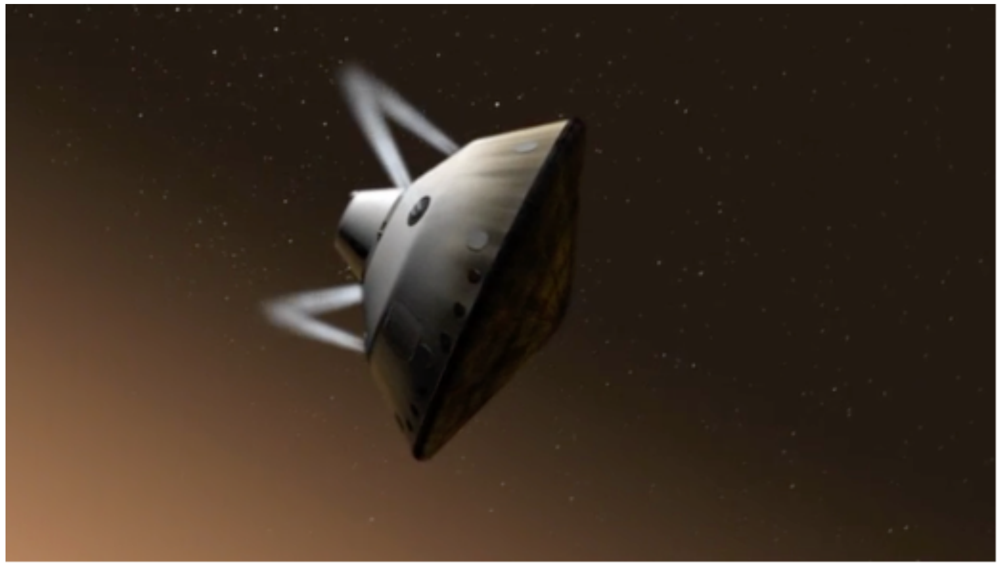
Estimate state to deploy parachute

Control problem

Learning problem

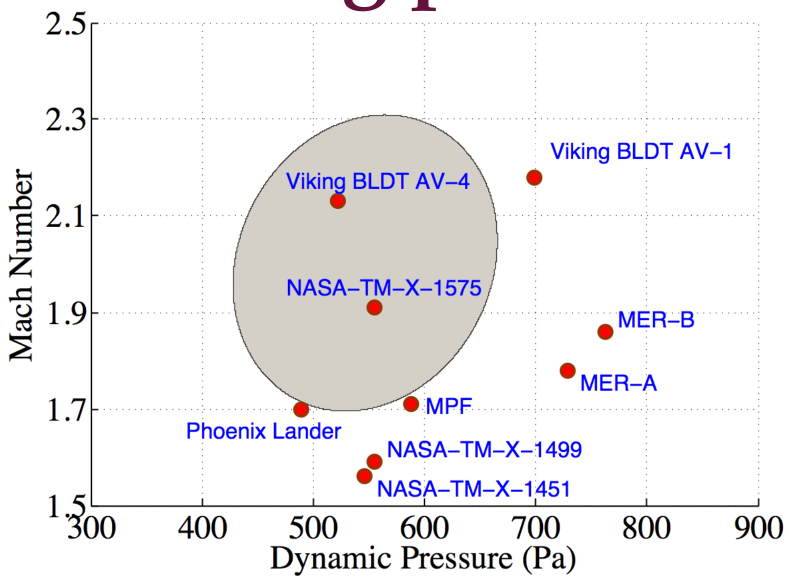
PDFs in Mars Entry-Descent-Landing

Prediction problem



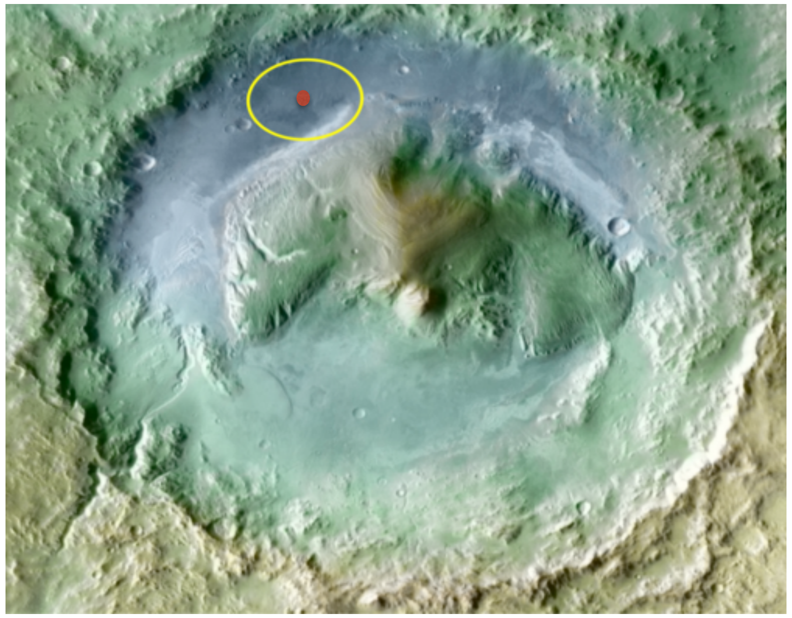
Predict heating rate uncertainty

Filtering problem



Estimate state to deploy parachute

Control problem



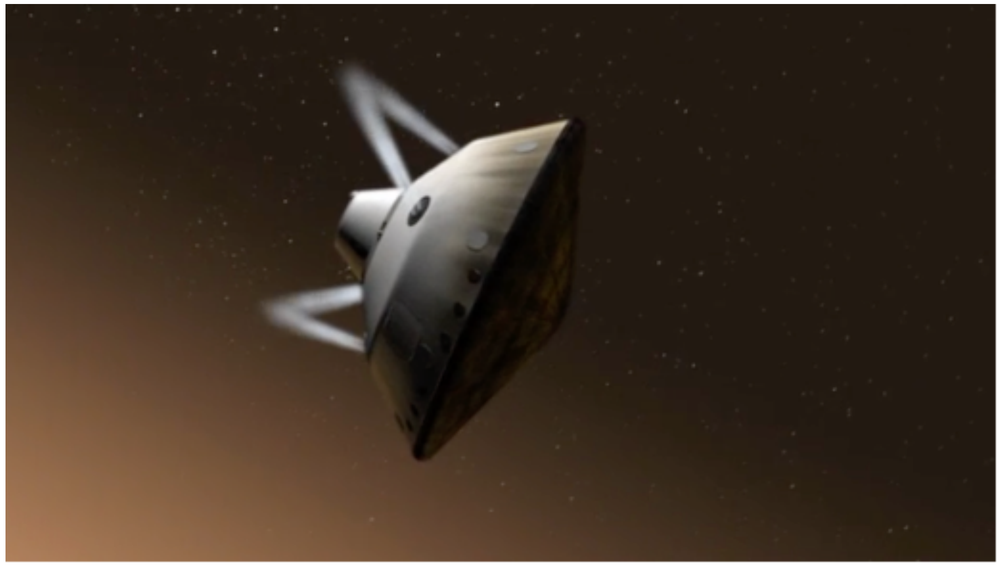
Gale Crater (4.49S, 137.42E)

Steer state PDF to achieve desired landing footprint accuracy

Learning problem

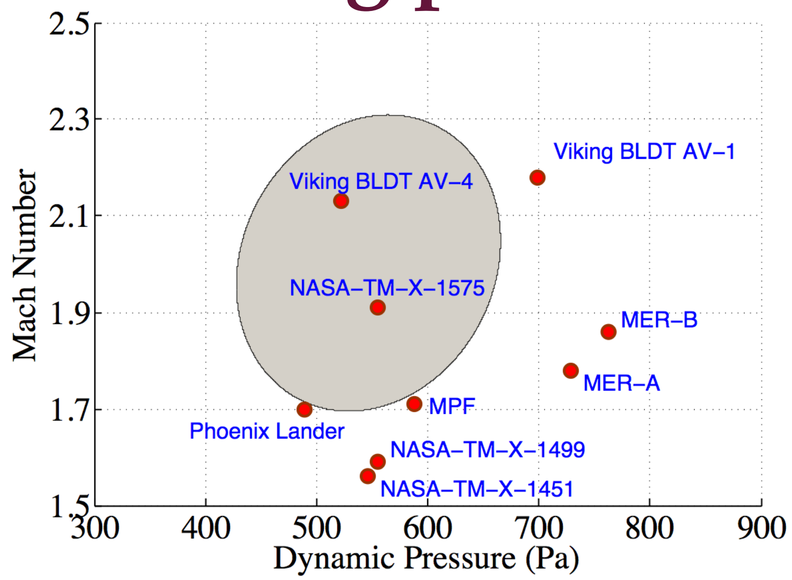
PDFs in Mars Entry-Descent-Landing

Prediction problem



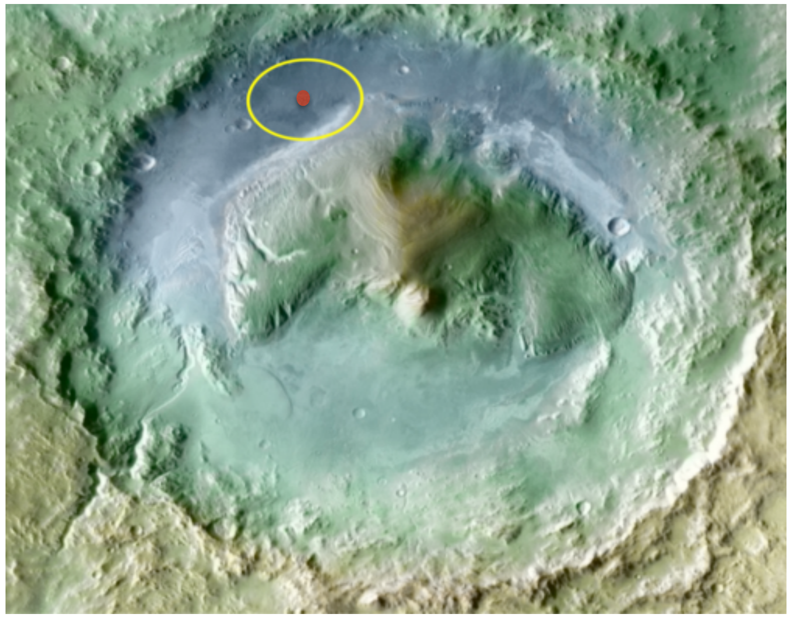
Predict heating rate uncertainty

Filtering problem



Estimate state to deploy parachute

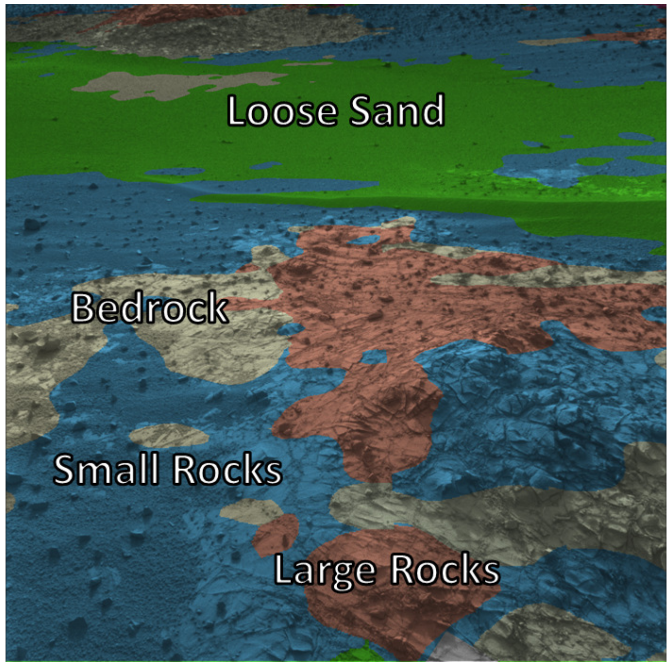
Control problem



Gale Crater (4.49S, 137.42E)

Steer state PDF to achieve desired landing footprint accuracy

Learning problem



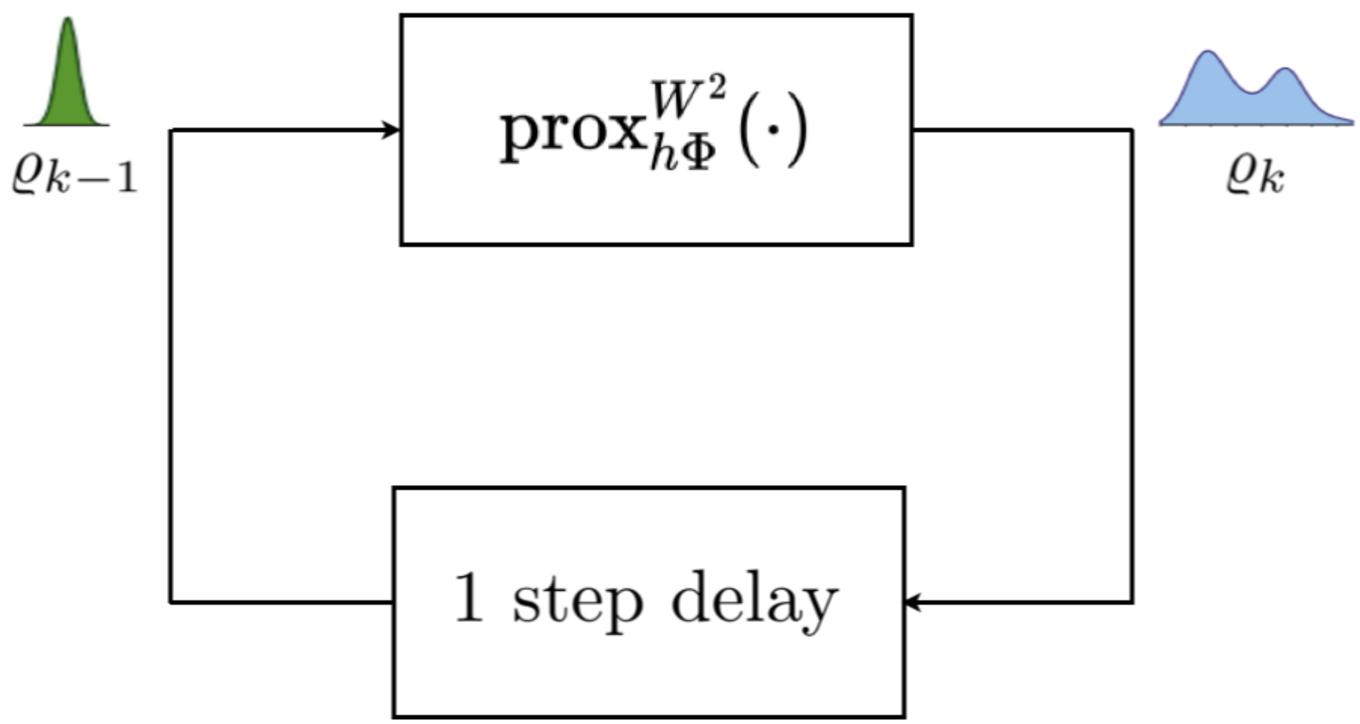
Learn surface feature from data

Solving prediction problem as Wasserstein gradient flow

What's New?

Main idea: Solve $\dfrac{\partial \rho}{\partial t} = \mathcal{L}_{\mathrm{FP}}\rho, \;\; \rho(x, t=0) = \rho_0$ as gradient flow in $\mathcal{P}_2(\mathcal{X})$

Infinite dimensional variational recursion:



Proximal operator: $$\varrho_k = \mathrm{prox}^{W^2}_{h\Phi}(\varrho_{k-1}) := \underset{\varrho \in \mathcal{P}_2(\mathcal{X})}{\arg\inf} \left\{ \frac{1}{2} W^2(\varrho, \varrho_{k-1}) + h\Phi(\varrho) \right\}$$

Optimal transport cost: $$W^2(\varrho, \varrho_{k-1}) := \inf_{\pi \in \Pi(\varrho, \varrho_{k-1})} \int_{\mathcal{X} \times \mathcal{X}} c(x, y) \, \mathrm{d}\pi(x, y)$$

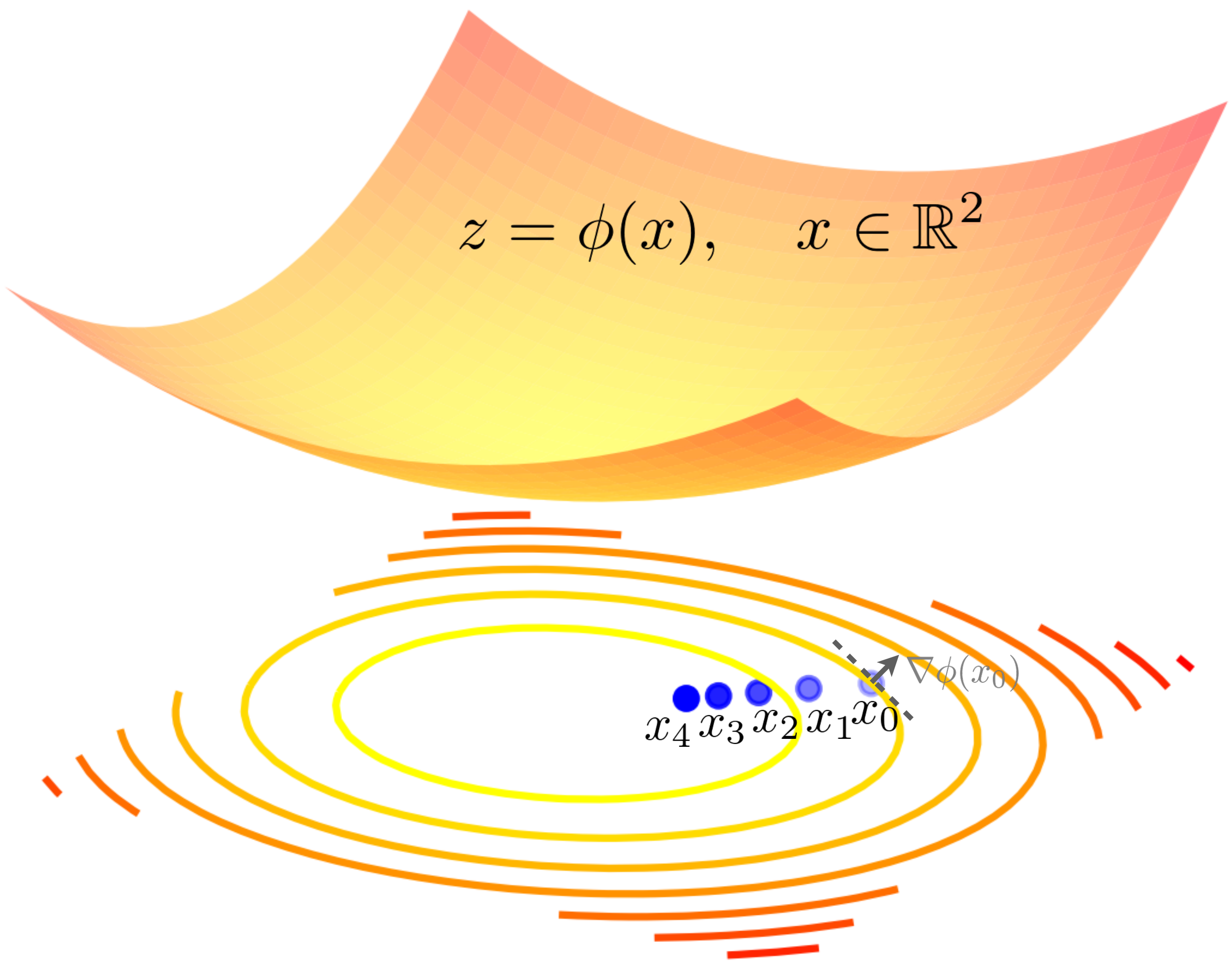
Free energy functional: $$\Phi(\varrho) := \int_{\mathcal{X}} \psi \varrho \, \mathrm{d}x + \beta^{-1} \int_{\mathcal{X}} \varrho \log \varrho \, \mathrm{d}x$$

Geometric Meaning of Gradient Flow

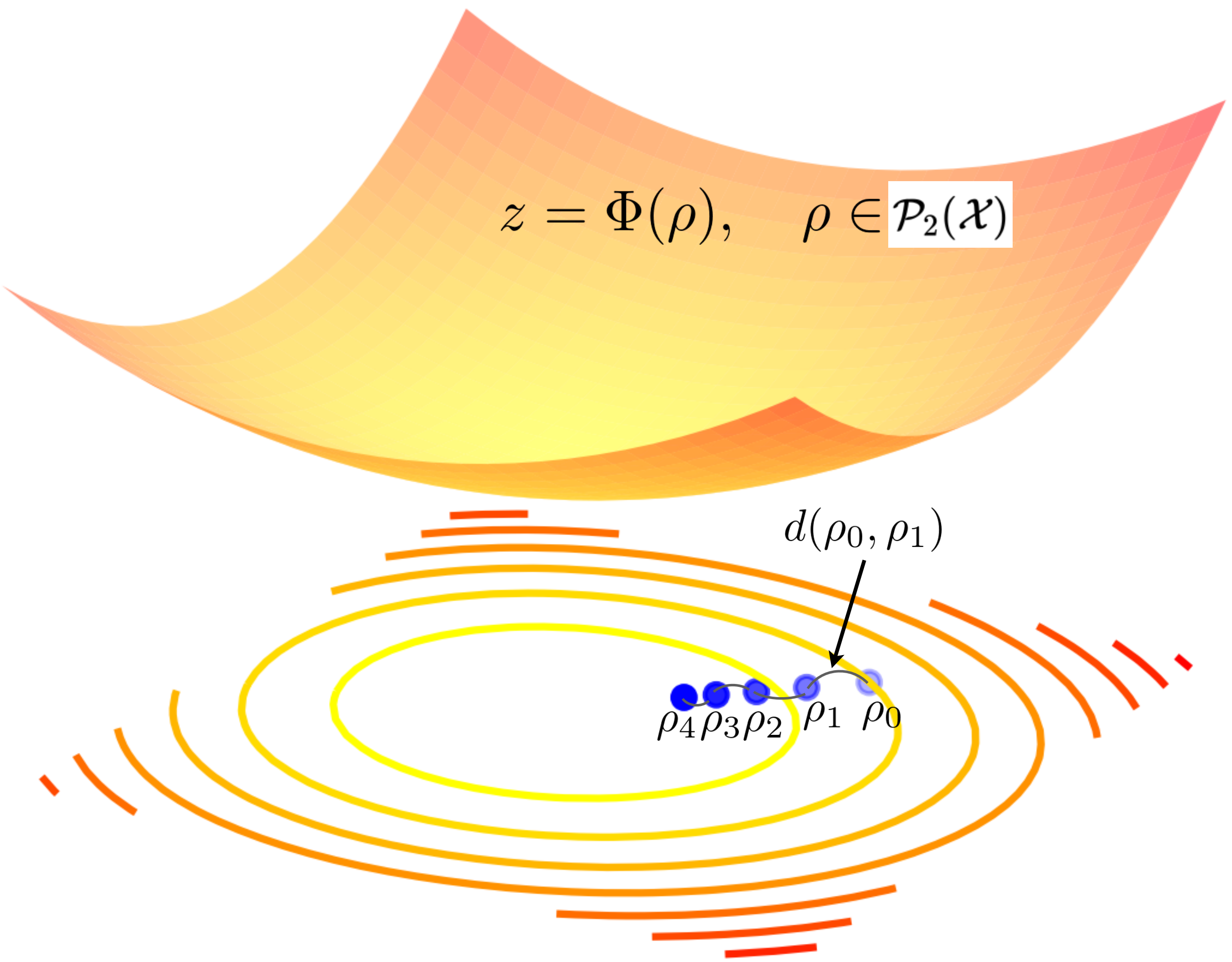
Gradient Flow in $\mathcal{X}$	Gradient Flow in $\mathcal{P}_2(\mathcal{X})$
$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = -\nabla\varphi(\boldsymbol{x}), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0$	$\frac{\partial\rho}{\partial t} = -\nabla^{W}\Phi(\rho), \quad \rho(\boldsymbol{x},0) = \rho_0$
Recursion:	**Recursion:**
$\begin{aligned}\boldsymbol{x}_k &= \boldsymbol{x}_{k-1} - h\nabla\varphi(\boldsymbol{x}_k) \\ &= \underset{\boldsymbol{x}\in\mathcal{X}}{\arg\min}\left\{\frac{1}{2}\|\boldsymbol{x} - \boldsymbol{x}_{k-1}\|_2^2 + h\varphi(\boldsymbol{x})\right\} \\ &=: \mathrm{prox}_{h\varphi}^{\|\cdot\|_2}(\boldsymbol{x}_{k-1})\end{aligned}$	$\begin{aligned}\rho_k &= \rho(\cdot, t = kh) \\ &= \underset{\rho\in\mathcal{P}_2(\mathcal{X})}{\arg\min}\left\{\frac{1}{2}W^2(\rho,\rho_{k-1}) + h\Phi(\rho)\right\} \\ &=: \mathrm{prox}_{h\Phi}^{W^2}(\rho_{k-1})\end{aligned}$
Convergence:	**Convergence:**
$\boldsymbol{x}_k \to \boldsymbol{x}(t = kh) \quad \text{as} \quad h\downarrow 0$	$\rho_k \to \rho(\cdot, t = kh) \quad \text{as} \quad h\downarrow 0$
φ as Lyapunov function:	**Φ as Lyapunov functional:**
$\frac{\mathrm{d}}{\mathrm{d}t}\varphi = -\|\nabla\varphi\|_2^2 \leq 0$	$\frac{\mathrm{d}}{\mathrm{d}t}\Phi = -\mathbb{E}_\rho\left[\left\|\nabla\frac{\delta\Phi}{\delta\rho}\right\|_2^2\right] \leq 0$

Geometric Meaning of Gradient Flow

Gradient Flow in $\mathcal{X}$

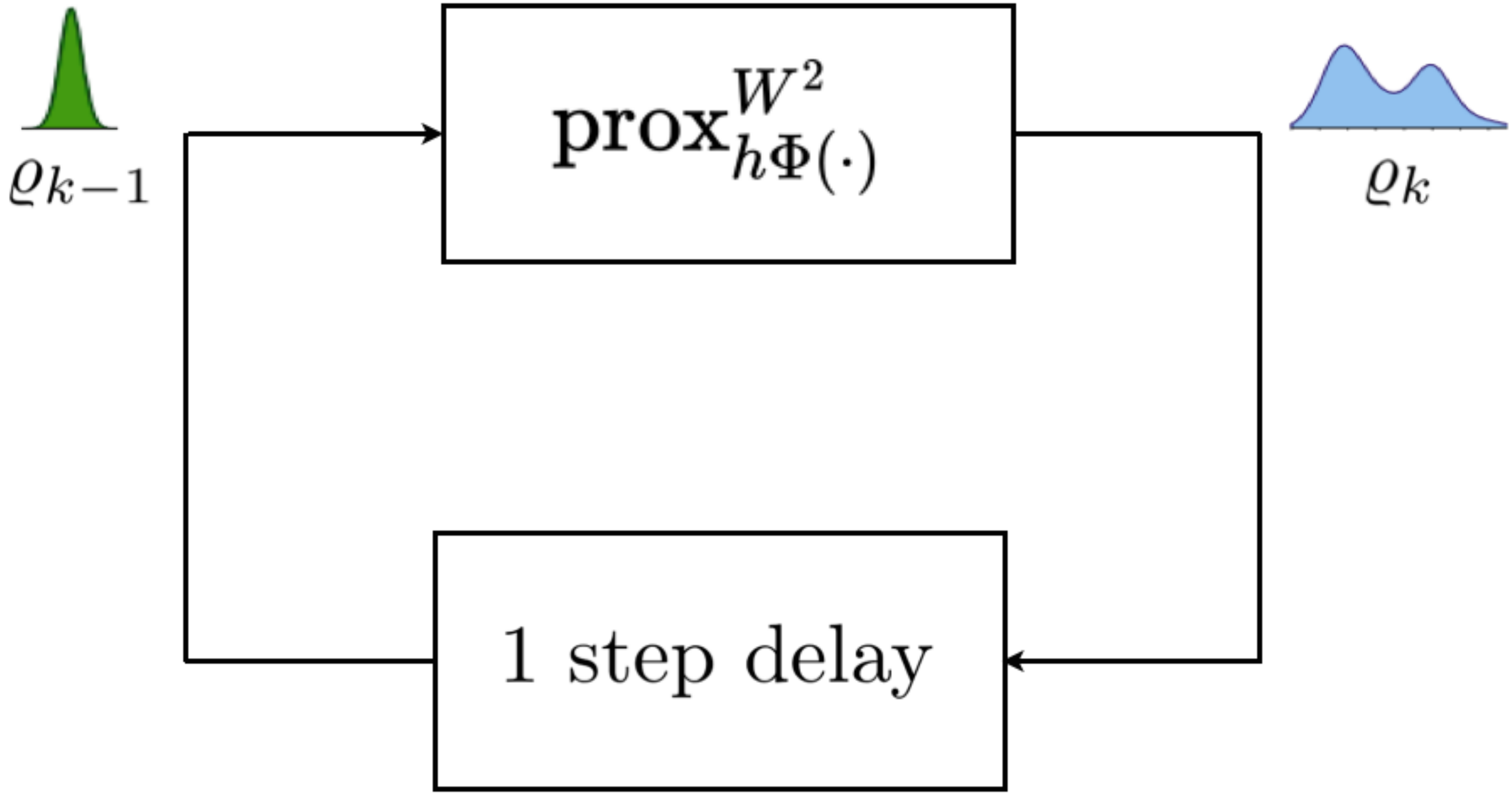


Gradient Flow in $\mathcal{P}_2(\mathcal{X})$



Algorithm: Gradient Ascent on the Dual Space

Uncertainty propagation via point clouds



No spatial discretization or function approximation

Algorithm: Gradient Ascent on the Dual Space

$$\frac{\partial \rho}{\partial t} = \nabla \cdot (\nabla \psi \rho) + \beta^{-1} \Delta \rho$$

$\Updownarrow$ **Proximal Recursion**

$$\rho_k = \rho(\boldsymbol{x}, t = kh) = \underset{\rho \in \mathcal{P}_2(\mathbb{R}^n)}{\arg\inf} \left\{ \frac{1}{2} W^2(\rho, \rho_{k-1}) + h\, \Phi(\rho) \right\}$$

$\Downarrow$ **Discrete Primal Formulation**

$$\boldsymbol{\varrho}_k = \arg\min_{\boldsymbol{\varrho}} \left\{ \min_{\boldsymbol{M} \in \Pi(\boldsymbol{\varrho}_{k-1}, \boldsymbol{\varrho})} \frac{1}{2} \langle \boldsymbol{C}_k, \boldsymbol{M} \rangle + h \, \langle \boldsymbol{\psi}_{k-1} + \beta^{-1} \log \boldsymbol{\varrho}, \boldsymbol{\varrho} \rangle \right\}$$

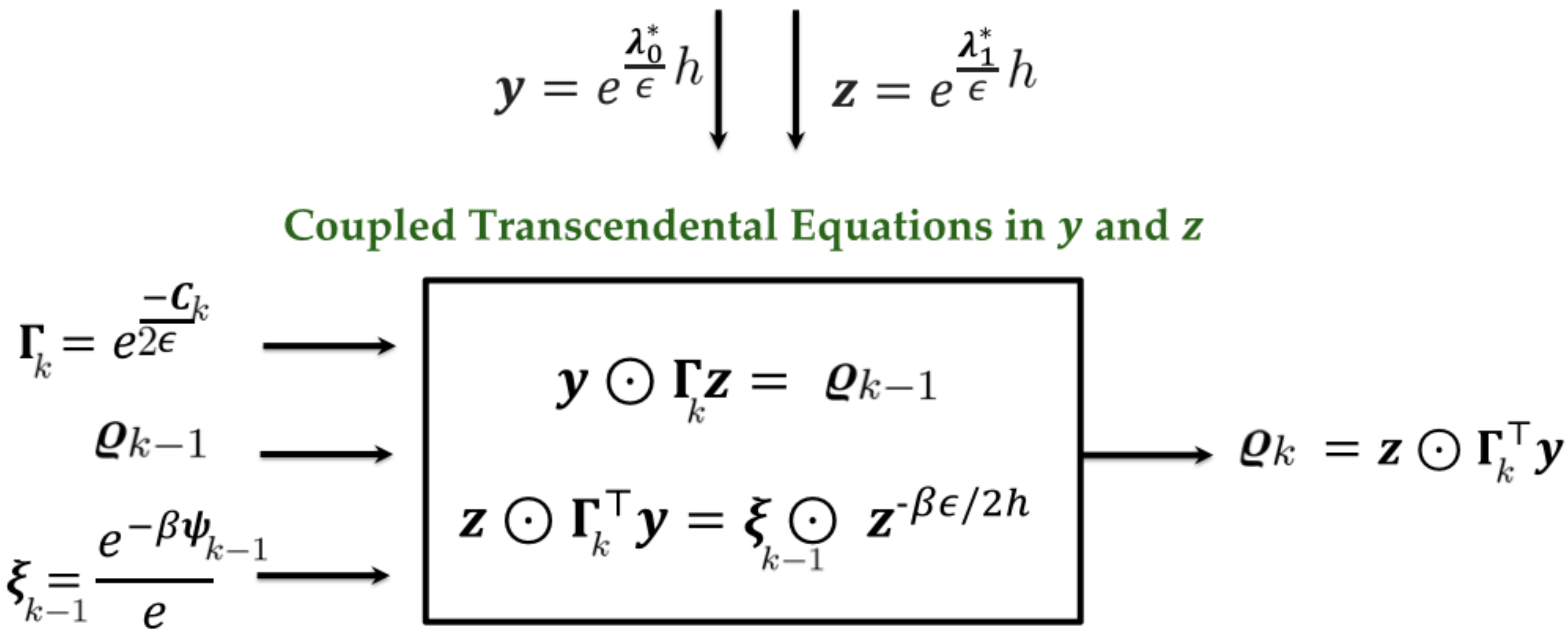
$\Downarrow$ **Entropic Regularization**

$$\boldsymbol{\varrho}_k = \arg\min_{\boldsymbol{\varrho}} \left\{ \min_{\boldsymbol{M} \in \Pi(\boldsymbol{\varrho}_{k-1}, \boldsymbol{\varrho})} \frac{1}{2} \langle \boldsymbol{C}_k, \boldsymbol{M} \rangle + \epsilon H(\boldsymbol{M}) + h \, \langle \boldsymbol{\psi}_{k-1} + \beta^{-1} \log \boldsymbol{\varrho}, \boldsymbol{\varrho} \rangle \right\}$$

$\Updownarrow$ **Dualization**

$$\boldsymbol{\lambda}_0^{\mathrm{opt}}, \boldsymbol{\lambda}_1^{\mathrm{opt}} = \underset{\boldsymbol{\lambda}_0, \boldsymbol{\lambda}_1 \geq 0}{\arg\max} \Big\{ \langle \boldsymbol{\lambda}_0, \boldsymbol{\varrho}_{k-1} \rangle - F^{\star}(-\boldsymbol{\lambda}_1) - \frac{\epsilon}{h} \left(\exp(\boldsymbol{\lambda}_0^{\top} h/\epsilon) \exp(-\boldsymbol{C}_k / 2\epsilon) \exp(\boldsymbol{\lambda}_1 h/\epsilon) \right) \Big\}$$

Recursion on the Cone

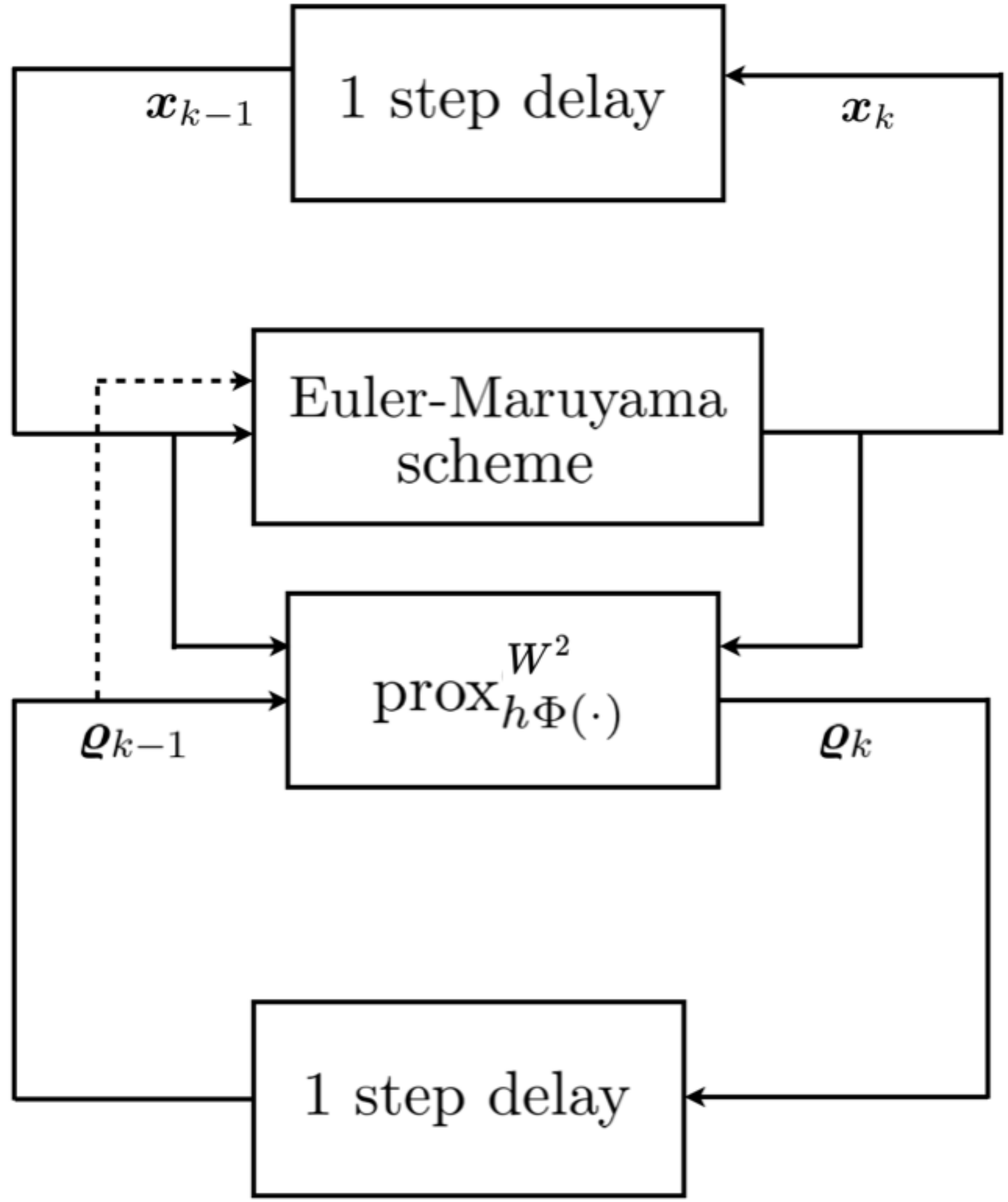


Theorem: Consider the recursion on the cone $\mathbb{R}^n_{\geq 0} \times \mathbb{R}^n_{\geq 0}$

$$\boldsymbol{y} \odot (\boldsymbol{\Gamma}_k \boldsymbol{z}) = \varrho_{k-1}, \quad \boldsymbol{z} \odot \left(\boldsymbol{\Gamma}_k{}^\top \boldsymbol{y}\right) = \boldsymbol{\xi}_{k-1} \odot \boldsymbol{z}^{-\frac{\beta\epsilon}{h}},$$

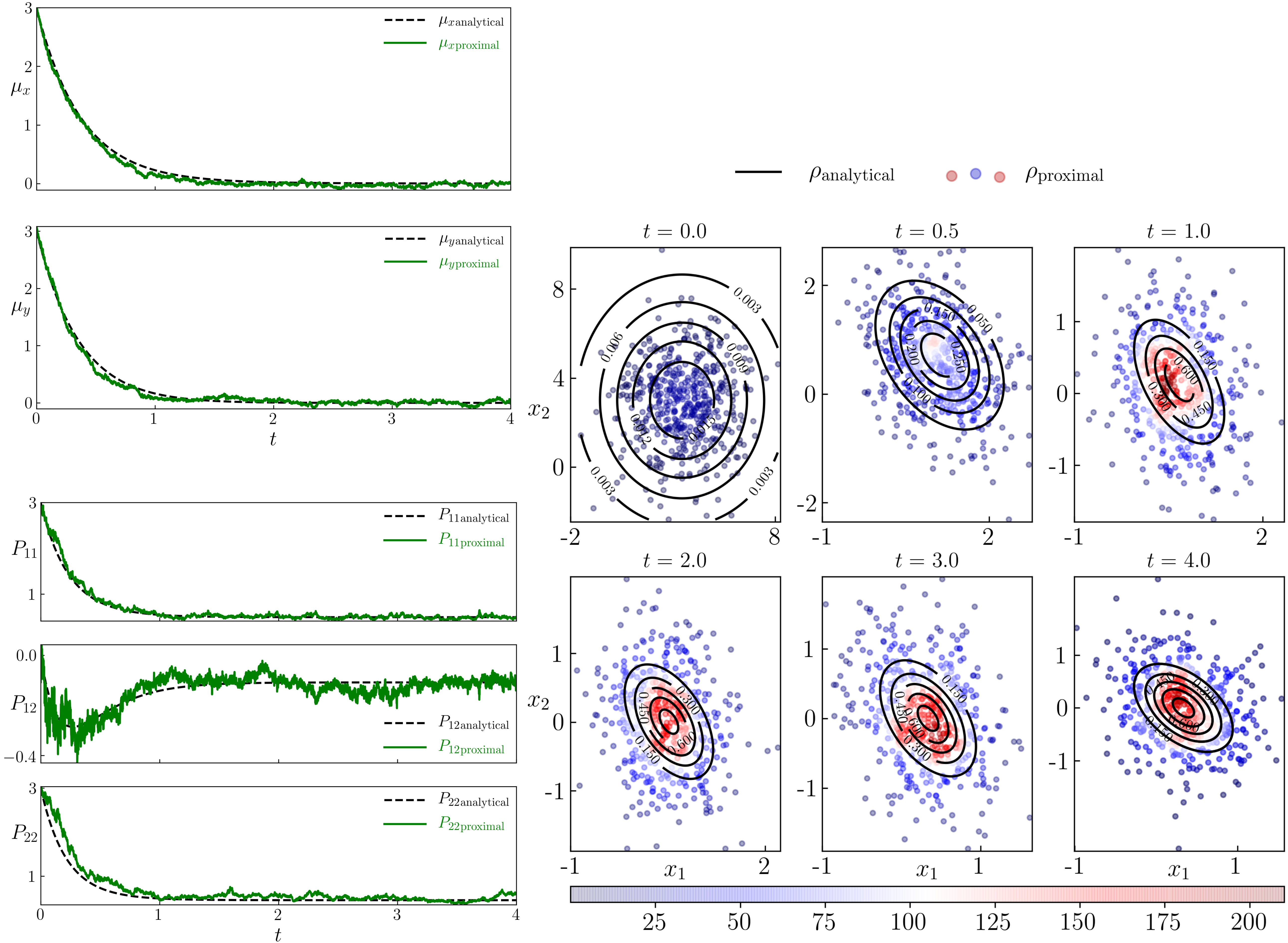
Then the solution $(\boldsymbol{y}^*, \boldsymbol{z}^*)$ gives the proximal update $\varrho_k = \boldsymbol{z}^* \odot (\boldsymbol{\Gamma}_k{}^\top \boldsymbol{y}^*)$

Algorithmic Setup

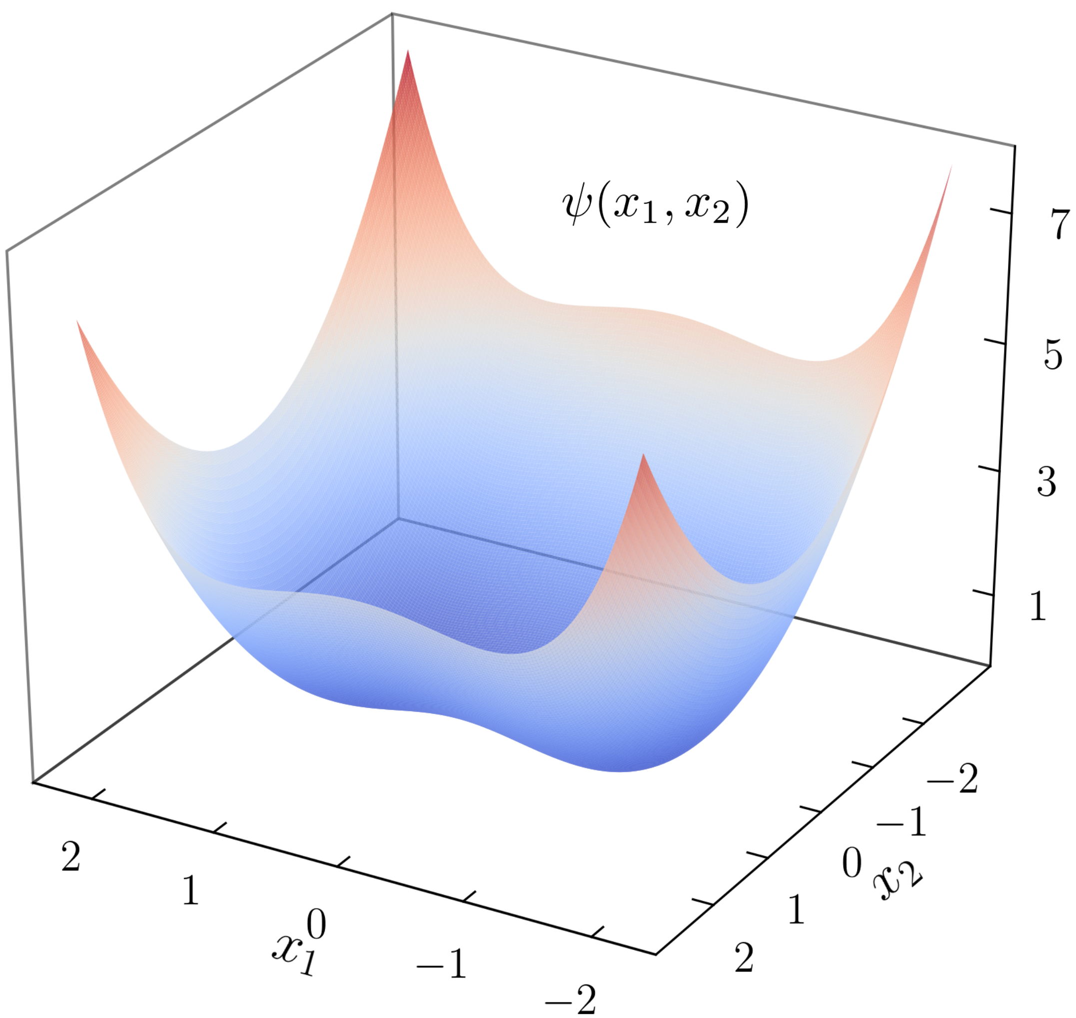


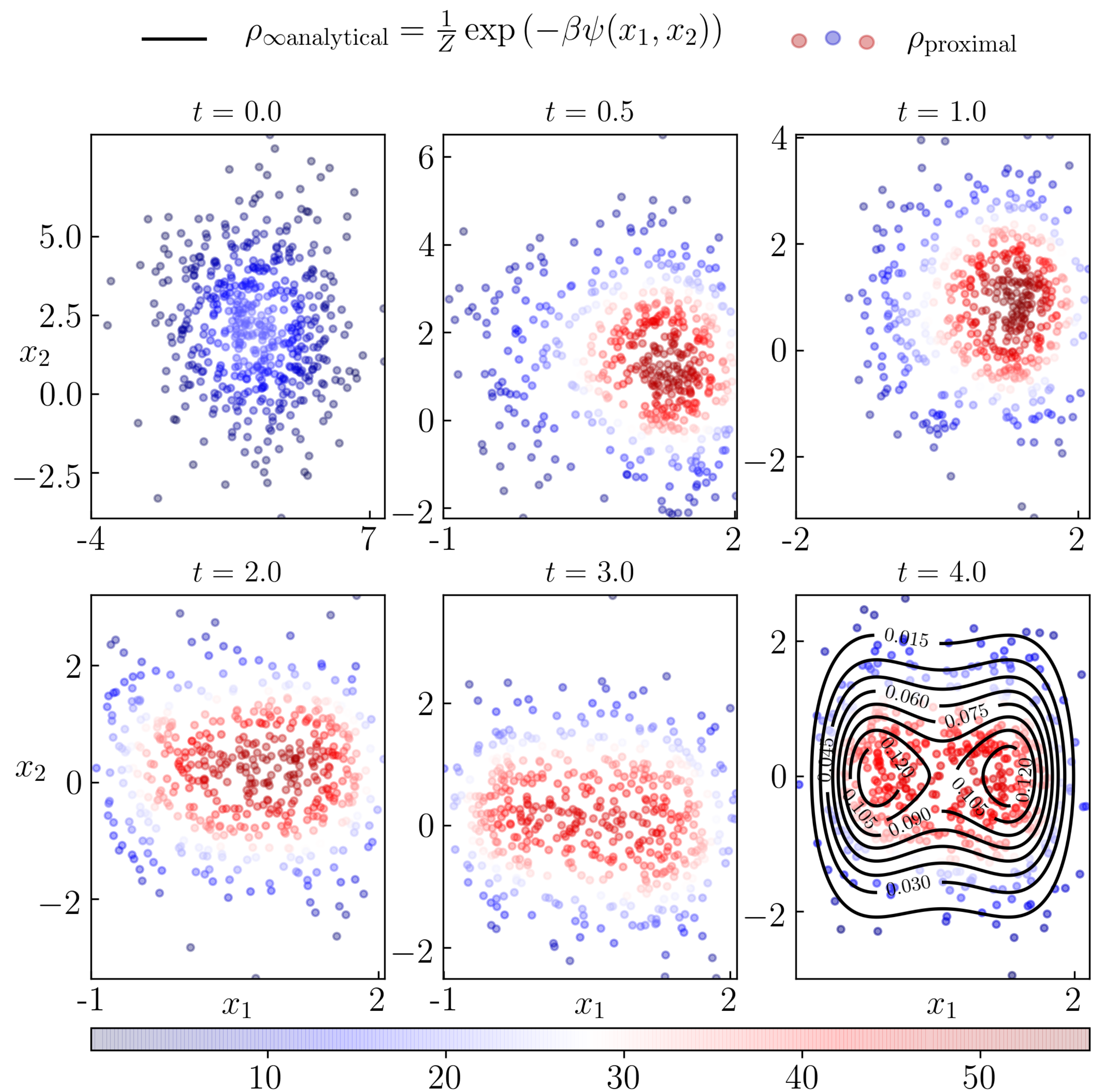
Theorem: Block co-ordinate iteration of $(\mathbf{y}, \mathbf{z})$ recursion is contractive on $\mathbb{R}^n_{>0} \times \mathbb{R}^n_{>0}$.

Proximal Prediction: 2D Linear Gaussian

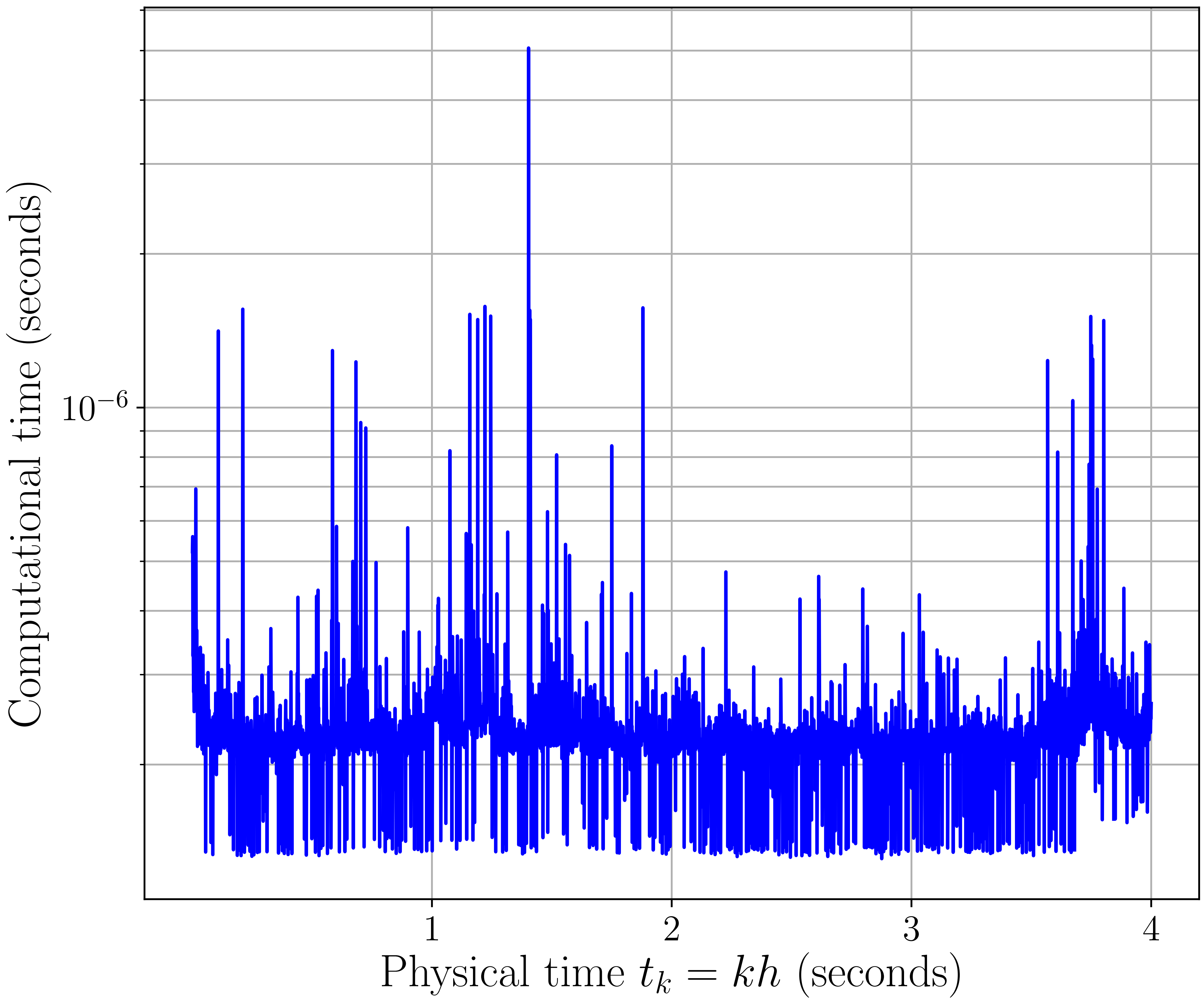


Proximal Prediction: Nonlinear Non-Gaussian





Computational Time: Nonlinear Non-Gaussian



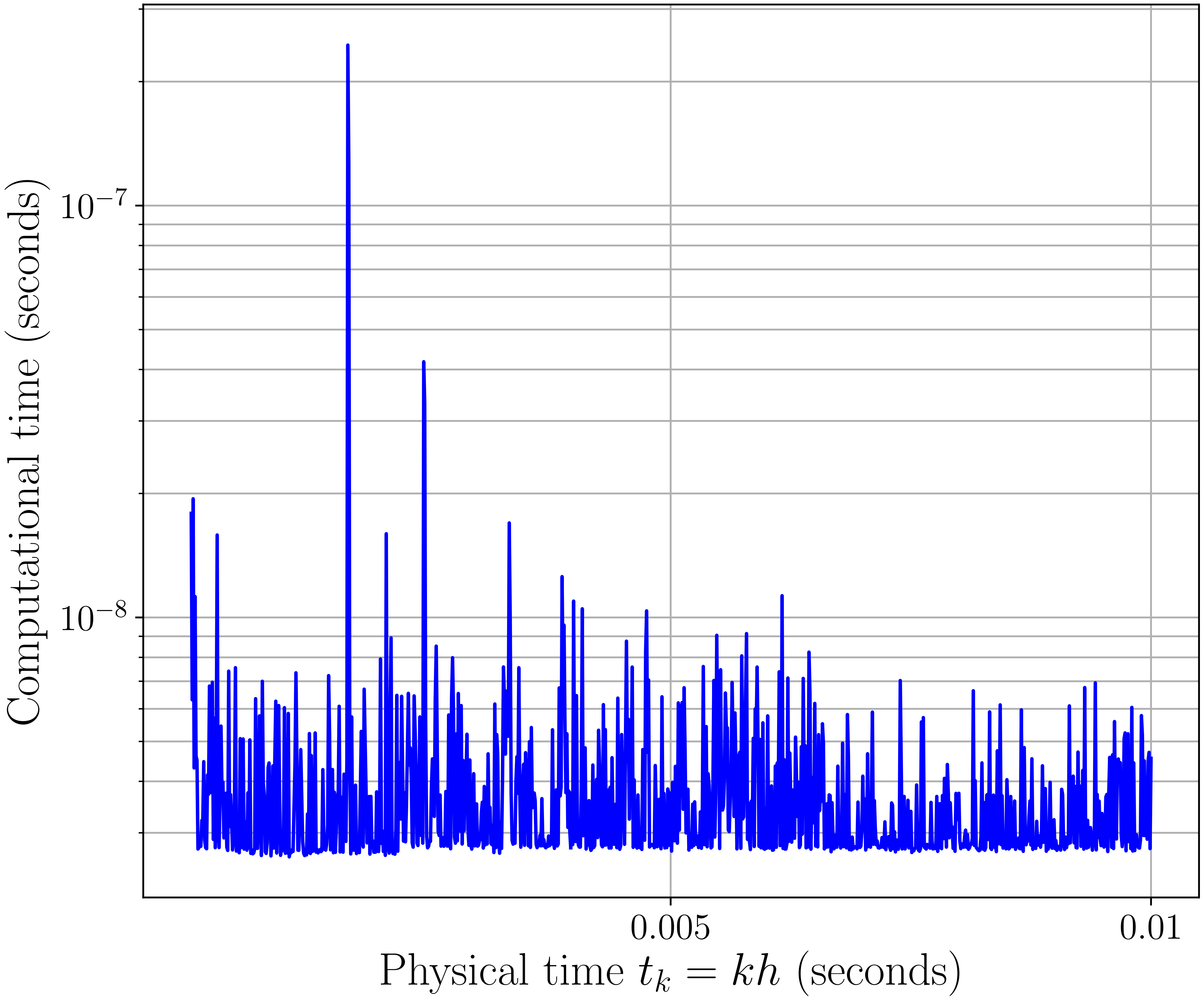
Proximal Prediction: Satellite in Geocentric Orbit

Here, $\mathcal{X} \equiv \mathbb{R}^6$

$$\begin{pmatrix} \mathrm{d}x \\ \mathrm{d}y \\ \mathrm{d}z \\ \mathrm{d}v_x \\ \mathrm{d}v_y \\ \mathrm{d}v_z \end{pmatrix} = \begin{pmatrix} v_x \\ v_y \\ v_z \\ -\dfrac{\mu x}{r^3} + (f_x)_{\mathrm{pert}} - \gamma v_x \\ -\dfrac{\mu y}{r^3} + (f_y)_{\mathrm{pert}} - \gamma v_y \\ -\dfrac{\mu z}{r^3} + (f_z)_{\mathrm{pert}} - \gamma v_z \end{pmatrix} \mathrm{d}t + \sqrt{2\beta^{-1}\gamma} \begin{pmatrix} 0 \\ 0 \\ 0 \\ \mathrm{d}w_1 \\ \mathrm{d}w_2 \\ \mathrm{d}w_3 \end{pmatrix},$$

$$\begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix}_{\mathrm{pert}} = \begin{pmatrix} s\theta\, c\phi & c\theta\, c\phi & -s\phi \\ s\theta\, s\phi & c\theta\, s\phi & c\phi \\ c\theta & -s\theta & 0 \end{pmatrix} \begin{pmatrix} \dfrac{k}{2r^4}\left(3(s\theta)^2 - 1\right) \\ -\dfrac{k}{r^5} s\theta\, c\theta \\ 0 \end{pmatrix}, k := 3J_2 R_{\mathrm{E}}^2, \mu = \mathrm{constant}$$

Computational Time: Satellite in Geocentric Orbit



Extensions: Nonlocal Interactions

PDF dependent sample path dynamics:

$$\mathrm{d}\mathbf{x} = -\left(\nabla U\left(\mathbf{x}\right) + \nabla\rho * V\right)\mathrm{d}t + \sqrt{2\beta^{-1}}\,\mathrm{d}\mathbf{w}$$

Mckean-Vlasov-Fokker-Planck-Kolmogorov integro PDE:

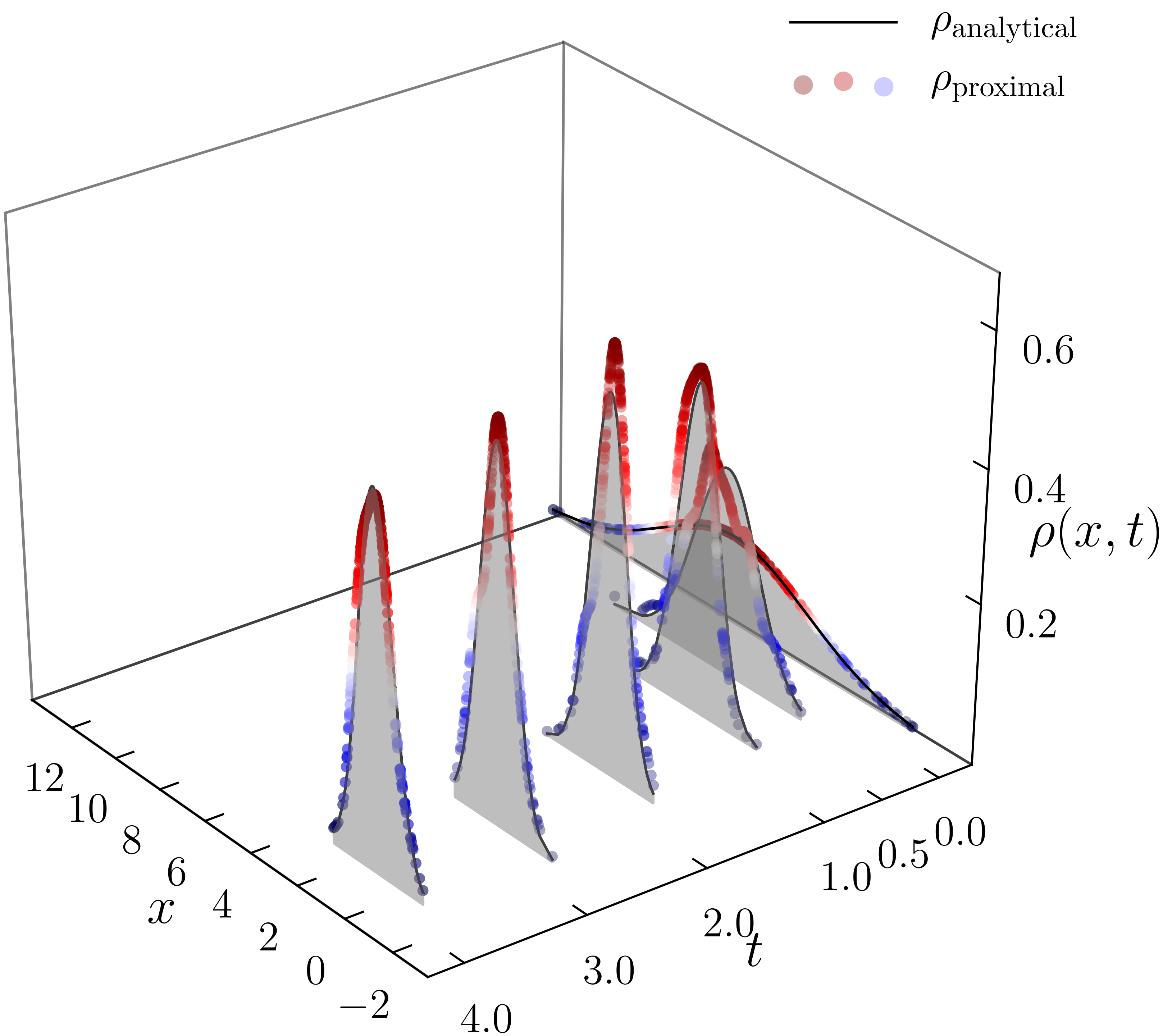
$$\frac{\partial\rho}{\partial t} = \nabla\cdot\left(\rho\nabla\left(U + \rho * V\right)\right) + \beta^{-1}\Delta\rho$$

Free energy:

$$F(\rho) := \mathbb{E}_{\rho}\left[U + \beta^{-1}\rho\log\rho + \rho * V\right]$$

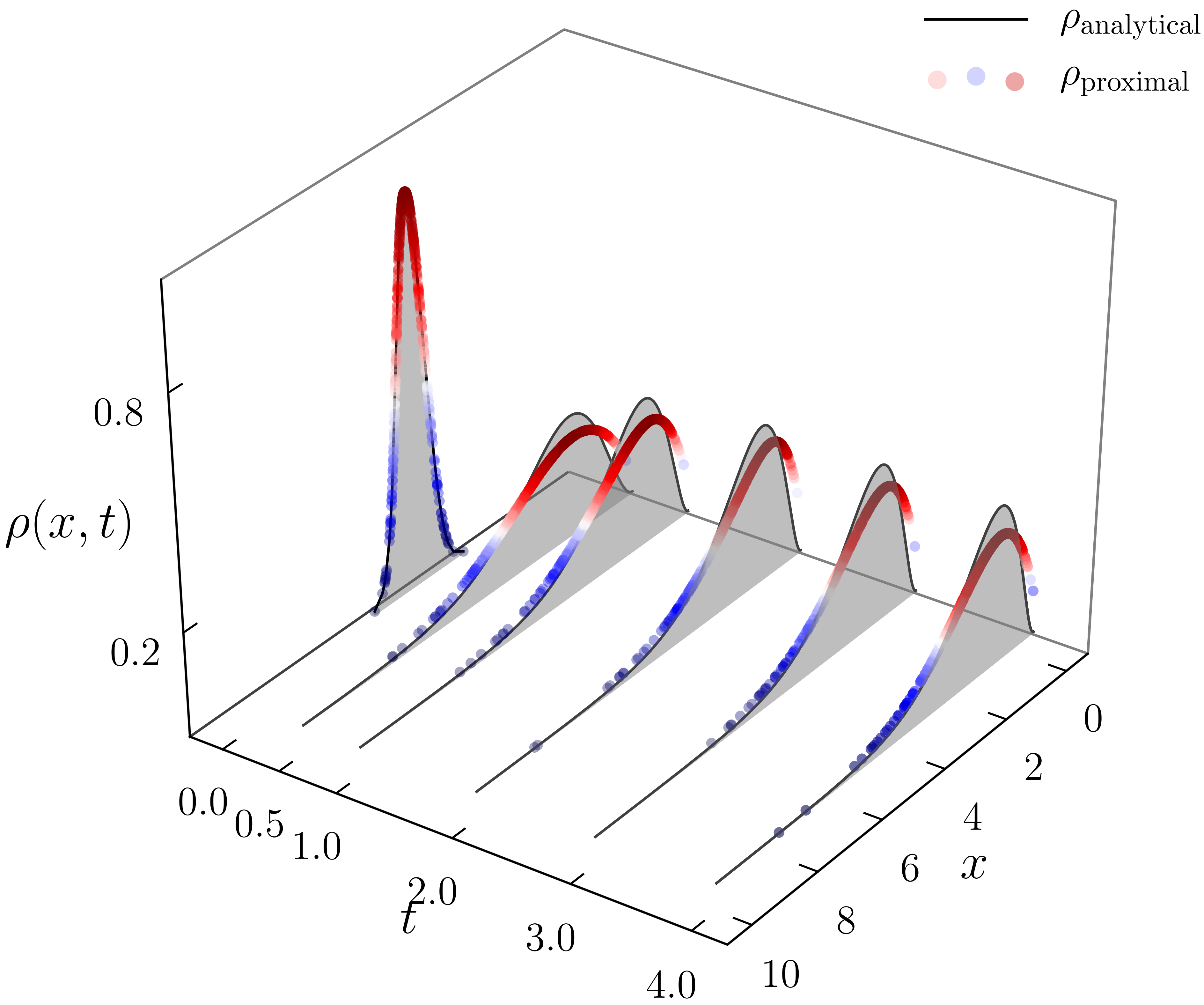
Extensions: Nonlocal Interactions

$$U(\cdot) = V(\cdot) = \| \cdot \|_2^2$$



Extensions: Multiplicative Noise

Cox-Ingersoll-Ross: $\mathrm{d}x = a\,(\theta - x)\;\mathrm{d}t + b\sqrt{x}\,\mathrm{d}w$, $2a > b^2, \theta > 0$



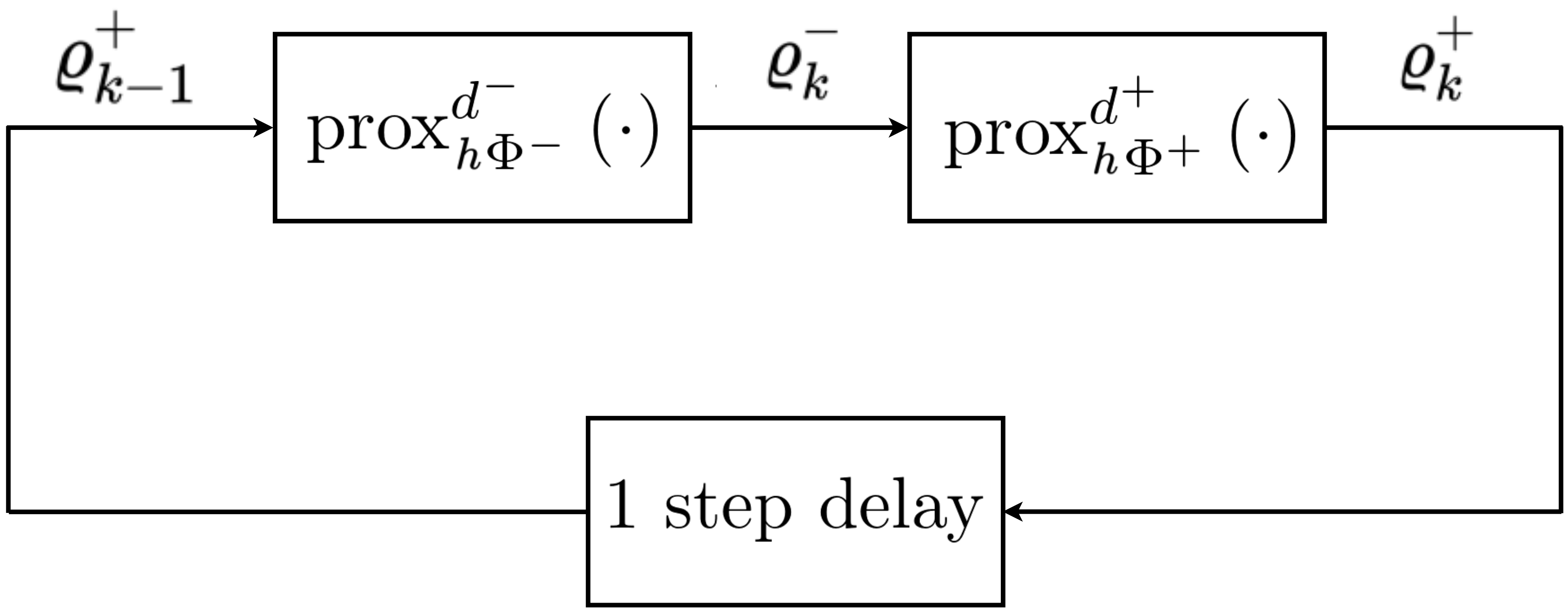
Solving filtering as Wasserstein gradient flow

What's New?

Main idea: Solve the Kushner-Stratonovich SPDE

$$\mathrm{d}\rho^{+} = \left[\mathcal{L}_{\mathrm{FP}}\mathrm{d}t + \mathcal{L}\left(\mathrm{d}z, \mathrm{d}t, \rho^{+}\right)\right]\rho^{+}, \;\; \rho(x, t=0) = \rho_0 \text{ as gradient flow in } \mathcal{P}_2(\mathcal{X})$$

Recursion of {deterministic $\circ$ stochastic} proximal operators:



Convergence: $\varrho_k^{+}(h) \to \rho^{+}(x, t = kh)$ as $h \downarrow 0$

For prior, as before: $d^{-} \equiv W^2, \quad \Phi^{-} \equiv \mathbb{E}_{\varrho}\left[\psi + \beta^{-1}\log\varrho\right]$

For posterior: $d^{+} \equiv d_{\mathrm{FR}}^{2}$ or $D_{\mathrm{KL}}, \quad \Phi^{+} \equiv \dfrac{1}{2}\mathbb{E}_{\varrho^{+}}\left[(y_k - h(x))^{\top}R^{-1}(y_k - h(x))\right]$

Explicit Recovery of the Kalman-Bucy Filter

Model:

$$\mathrm{d}\mathbf{x}(t) = \mathbf{A}\mathbf{x}(t)\mathrm{d}t + \mathbf{B}\mathrm{d}\mathbf{w}(t), \quad \mathrm{d}\mathbf{w}(t) \sim \mathcal{N}(0, \mathbf{Q}\mathrm{d}t)$$

$$\mathrm{d}\mathbf{z}(t) = \mathbf{C}\mathbf{x}(t)\mathrm{d}t + \mathrm{d}\mathbf{v}(t), \quad \mathrm{d}\mathbf{v}(t) \sim \mathcal{N}(0, \mathbf{R}\mathrm{d}t)$$

Given $\mathbf{x}(0) \sim \mathcal{N}(\mu_0, \mathbf{P}_0)$, want to recover:

$$\mathrm{d}\mu^+(t) = \mathbf{A}\mu^+(t)\mathrm{d}t + \underbrace{\mathbf{K}(t)}_{\mathbf{P}^+\mathbf{C}\mathbf{R}^{-1}} (\mathrm{d}\mathbf{z}(t) - \mathbf{C}\mu^+(t)\mathrm{d}t),$$

$$\dot{\mathbf{P}}^+(t) = \mathbf{A}\mathbf{P}^+(t) + \mathbf{P}^+(t)\mathbf{A}^\top + \mathbf{B}\mathbf{Q}\mathbf{B}^\top - \mathbf{K}(t)\mathbf{R}\mathbf{K}(t)^\top.$$

— A.H. and T.T. Georgiou, Gradient Flows in Uncertainty Propagation and Filtering of Linear Gaussian Systems, *CDC 2017*.

— A.H. and T.T. Georgiou, Gradient Flows in Filtering and Fisher-Rao Geometry, *ACC 2018*.

Explicit Recovery of the Wonham Filter

Model:

$$x(t) \sim \mathrm{Markov}(Q),$$
$$\mathrm{d}z(t) = h(x(t))\,\mathrm{d}t \;+\; \sigma_v(t)\mathrm{d}v(t)$$

J.SIAM Control
Ser. A, Vol. 2, No. 3
Printed in U.S.A., 1965

SOME APPLICATIONS OF STOCHASTIC DIFFERENTIAL EQUATIONS TO OPTIMAL NONLINEAR FILTERING*

W. M. WONHAM†

State space: $\Omega := \{a_1, \ldots, a_m\}$

Posterior $\pi^+(t) := \{\pi_1^+(t), \ldots, \pi_m^+(t)\}$ **solves the nonlinear SDE:**

$$\mathrm{d}\pi^+(t) = \pi^+(t)Q\,\mathrm{d}t \;+\; \frac{1}{(\sigma_v(t))^2}\pi^+(t)\Big(H - \widehat{h}(t)I\Big)\Big(\mathrm{d}z(t) - \widehat{h}(t)\mathrm{d}t\Big),$$

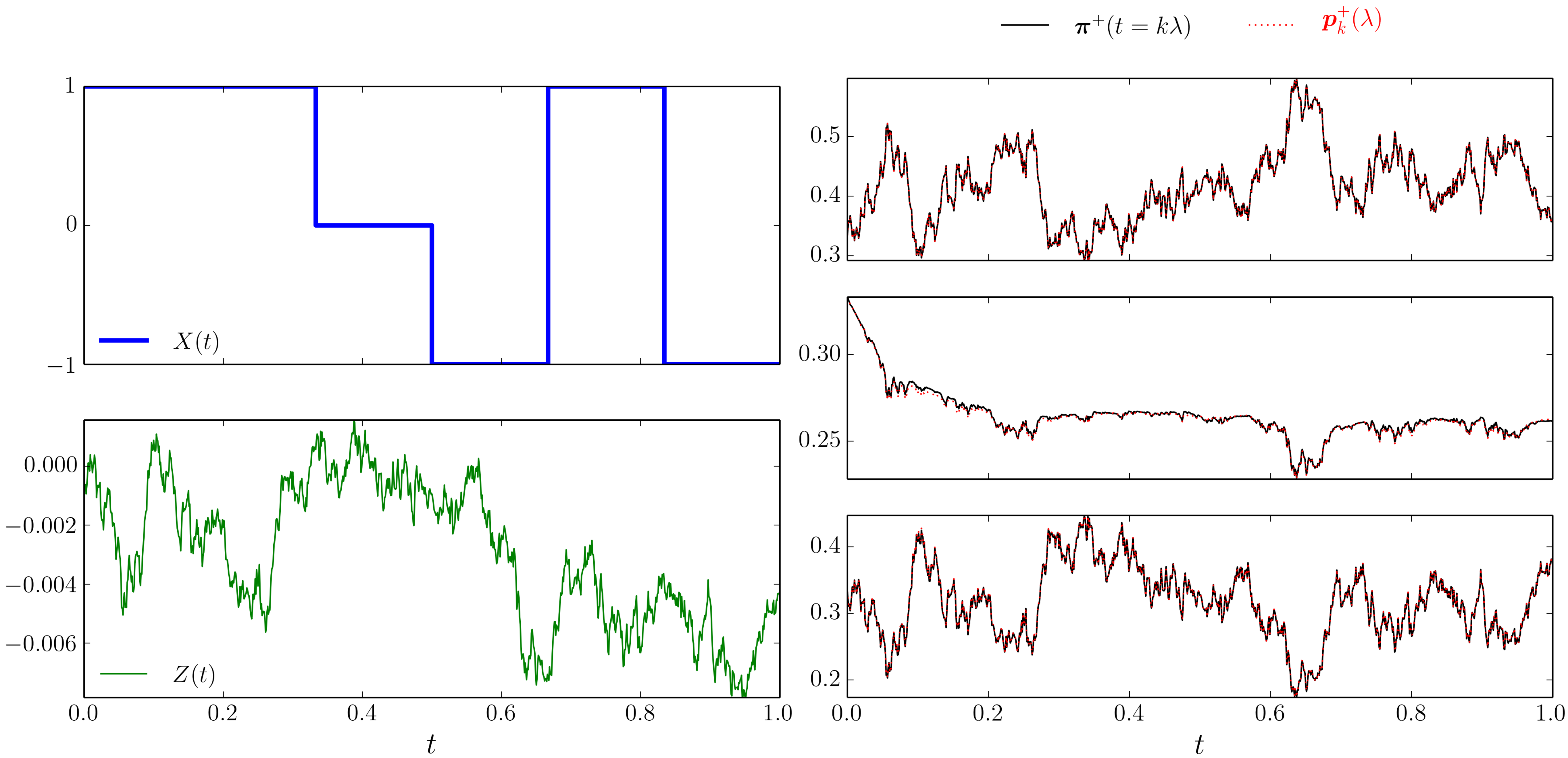
where $H := \mathrm{diag}(h(a_1), \ldots, h(a_m)), \quad \widehat{h}(t) := \sum_{i=1}^{m} h(a_i)\pi_i^+(t),$

Initial condition: $\pi^+(t=0) = \pi_0,$

By defn. $\pi^+(t) = \mathbb{P}(x(t) = a_i \mid z(s), 0 \leq s \leq t)$

— A.H. and T.T. Georgiou, Proximal Recursion for the Wonham Filter, *CDC 2019*.

Numerical Results for the Wonham Filter



— A.H. and T.T. Georgiou, Proximal Recursion for the Wonham Filter, *CDC 2019*.

Solving density control as Wasserstein gradient flow

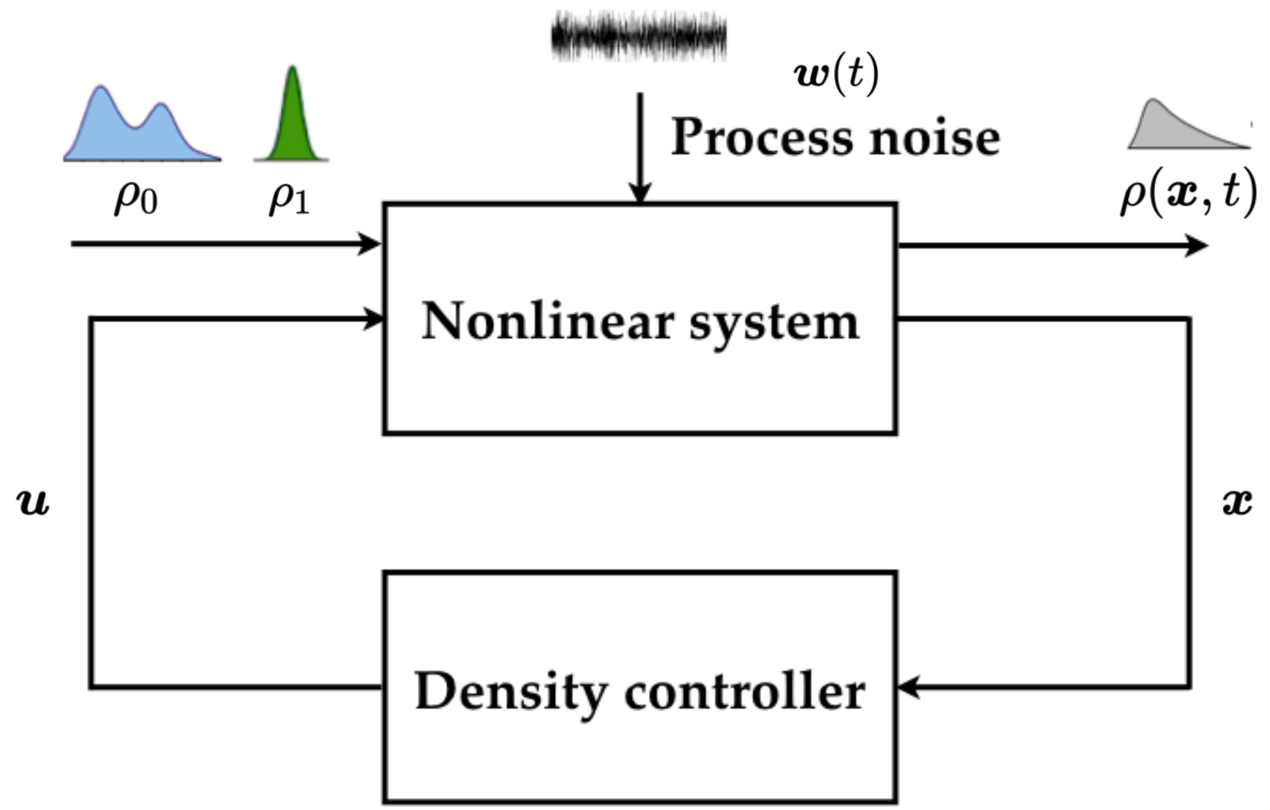
Finite Horizon Feedback Density Control

$$\underset{u \in \mathcal{U}}{\text{minimize}} \quad \mathbb{E}\left[\int_0^1 \|\boldsymbol{u}(\boldsymbol{x},t)\|_2^2 \, \mathrm{d}t\right]$$

subject to

$$\mathrm{d}\boldsymbol{x} = \left\{\boldsymbol{f}(\boldsymbol{x},t) + \boldsymbol{B}(t)\boldsymbol{u}(\boldsymbol{x},t)\right\}\mathrm{d}t + \sqrt{2\epsilon}\boldsymbol{B}(t)\,\mathrm{d}\boldsymbol{w},$$

$$\boldsymbol{x}(t=0) \sim \rho_0, \quad \boldsymbol{x}(t=1) \sim \rho_1$$



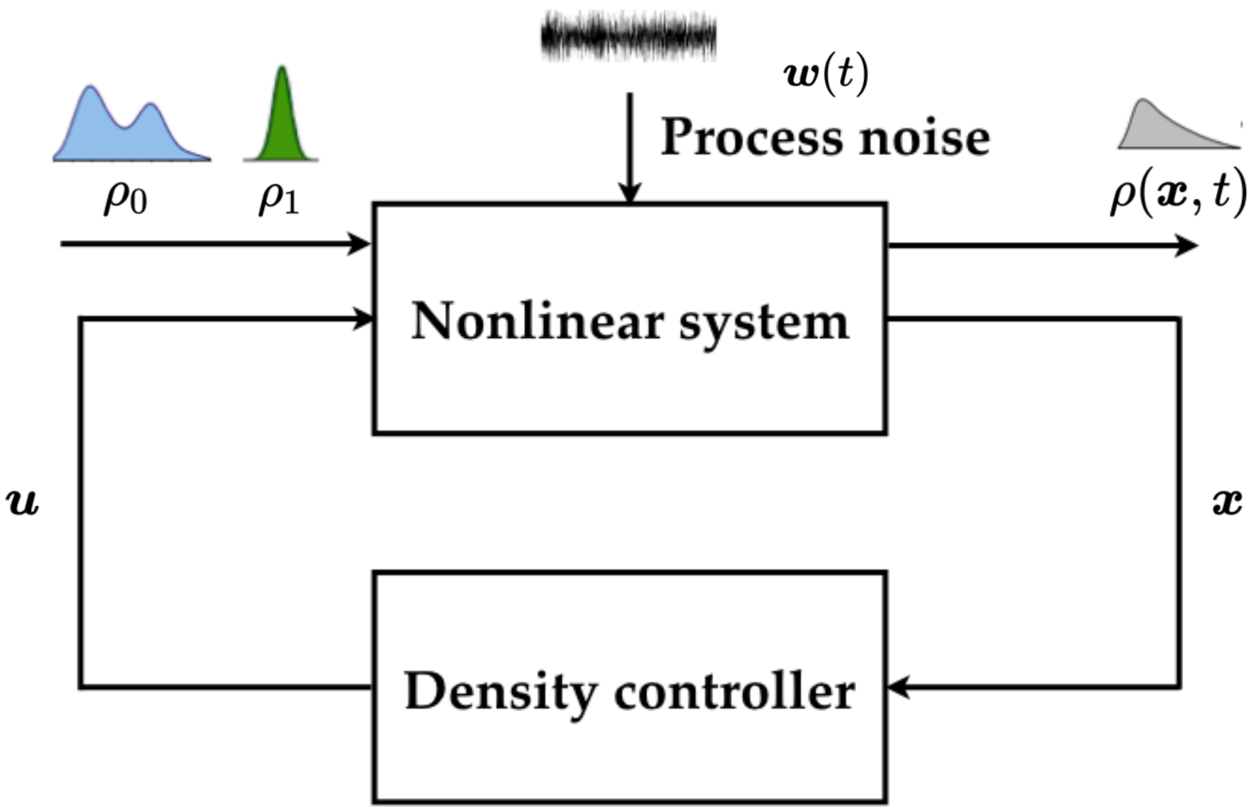
Finite Horizon Feedback Density Control

$$\underset{u \in \mathcal{U}}{\text{minimize}} \quad \mathbb{E}\left[\int_0^1 \|\boldsymbol{u}(\boldsymbol{x},t)\|_2^2 \, \mathrm{d}t\right]$$

subject to

$$\mathrm{d}\boldsymbol{x} = \left\{\boldsymbol{f}(\boldsymbol{x},t) + \boldsymbol{B}(t)\boldsymbol{u}(\boldsymbol{x},t)\right\}\mathrm{d}t + \sqrt{2\epsilon}\boldsymbol{B}(t)\,\mathrm{d}\boldsymbol{w},$$

$$\boldsymbol{x}(t=0) \sim \rho_0, \quad \boldsymbol{x}(t=1) \sim \rho_1$$



Necessary conditions for optimality: coupled nonlinear PDEs (FPK + HJB)

$$\frac{\partial \rho^{\text{opt}}}{\partial t} + \nabla \cdot \left(\rho^{\text{opt}}\left(\boldsymbol{f} + \boldsymbol{B}(t)^\top \nabla \psi\right)\right) = \epsilon \mathbf{1}^\top \left(\boldsymbol{D}(t) \odot \text{Hess}\left(\rho^{\text{opt}}\right)\right)\mathbf{1},$$

$$\frac{\partial \psi}{\partial t} + \frac{1}{2}\|\boldsymbol{B}(t)^\top \nabla \psi\|_2^2 + \langle \nabla \psi, \boldsymbol{f}\rangle = -\epsilon \langle \boldsymbol{D}(t), \text{Hess}(\psi)\rangle$$

Boundary conditions:

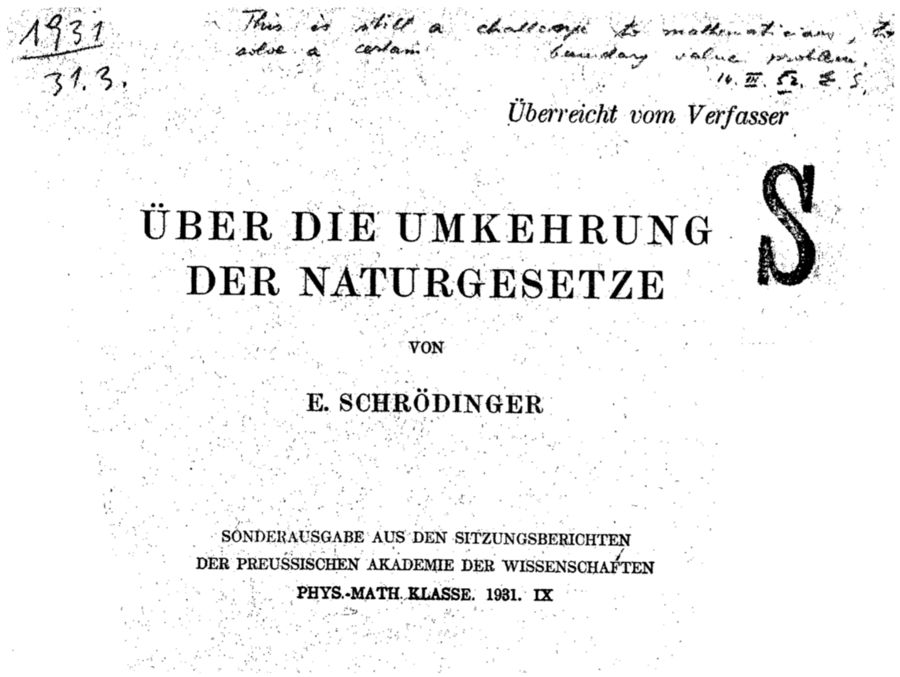
$$\rho^{\text{opt}}(\boldsymbol{x},0) = \rho_0(\boldsymbol{x}), \quad \rho^{\text{opt}}(\boldsymbol{x},1) = \rho_1(\boldsymbol{x})$$

Optimal control:

$$\boldsymbol{u}^{\text{opt}}(\boldsymbol{x},t) = \boldsymbol{B}(t)^\top \nabla \psi$$

Feedback Synthesis via the Schrödinger System

Schrödinger's (until recently) forgotten papers:



1931
31.3.

This is still a challenge to mathematicians, to solve a certain boundary value problem.

Überreicht vom Verfasser

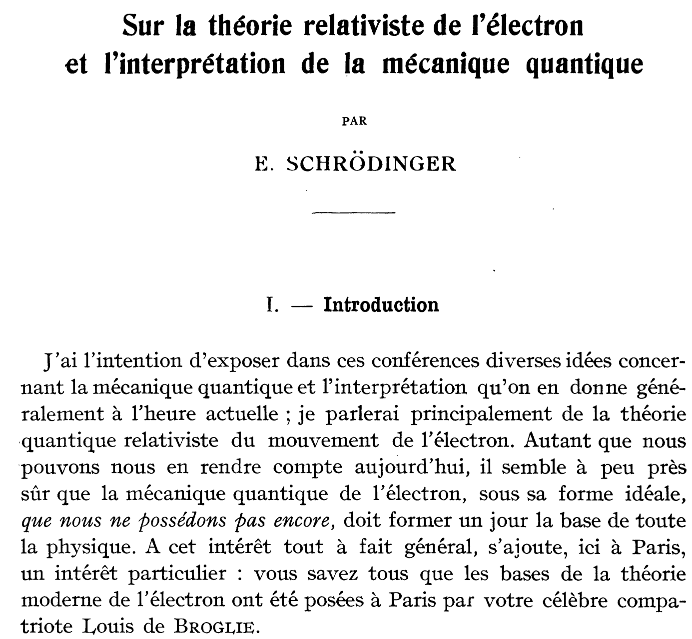
S

ÜBER DIE UMKEHRUNG
DER NATURGESETZE

VON

E. SCHRÖDINGER

SONDERAUSGABE AUS DEN SITZUNGSBERICHTEN
DER PREUSSISCHEN AKADEMIE DER WISSENSCHAFTEN
PHYS.-MATH. KLASSE. 1931. IX



Sur la théorie relativiste de l'électron
et l'interprétation de la mécanique quantique

PAR

E. SCHRÖDINGER

I. — **Introduction**

J'ai l'intention d'exposer dans ces conférences diverses idées concernant la mécanique quantique et l'interprétation qu'on en donne généralement à l'heure actuelle ; je parlerai principalement de la théorie quantique relativiste du mouvement de l'électron. Autant que nous pouvons nous en rendre compte aujourd'hui, il semble à peu près sûr que la mécanique quantique de l'électron, sous sa forme idéale, *que nous ne possédons pas encore*, doit former un jour la base de toute la physique. A cet intérêt tout à fait général, s'ajoute, ici à Paris, un intérêt particulier : vous savez tous que les bases de la théorie moderne de l'électron ont été posées à Paris par votre célèbre compatriote Louis de BROGLIE.

Hopf-Cole transform: $\left(\rho^{\text{opt}}, \psi\right) \mapsto \left(\varphi, \hat{\varphi}\right)$

$$\varphi\left(x,t\right) = \exp\left(\frac{\psi\left(x,t\right)}{2\epsilon}\right),$$

$$\hat{\varphi}\left(x,t\right) = \rho^{\text{opt}}\left(x,t\right)\exp\left(-\frac{\psi\left(x,t\right)}{2\epsilon}\right),$$

Optimal controlled joint state PDF: $\rho^{\text{opt}}\left(x,t\right) = \hat{\varphi}(x,t)\varphi(x,t)$

Optimal control: $u^{\text{opt}}(x,t) = 2\epsilon B(t)^{\top}\nabla\log\varphi\left(x,t\right)$

Feedback Synthesis via the Schrödinger System

2 coupled nonlinear PDEs $\longrightarrow$ boundary-coupled linear PDEs!!

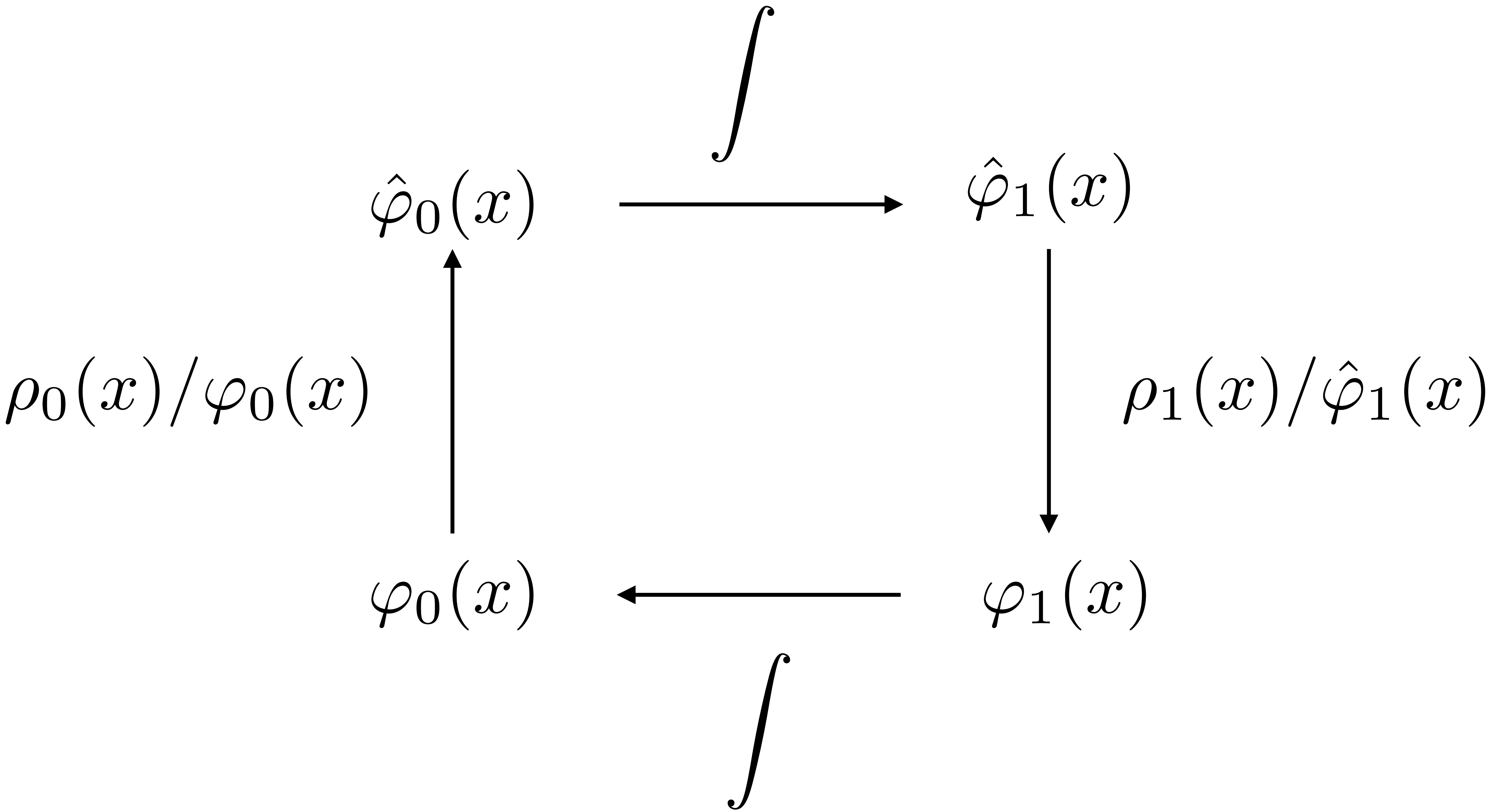
$$\underbrace{\frac{\partial \hat{\varphi}}{\partial t} = -\nabla \cdot (\hat{\varphi} \boldsymbol{f}) + \epsilon \mathbf{1}^{\top} (\boldsymbol{D}(t) \odot \operatorname{Hess}(\hat{\varphi}))\mathbf{1}}_{\text{forward Kolmogorov PDE}}, \quad \varphi_0 \hat{\varphi}_0 = \rho_0,$$

$$\underbrace{\frac{\partial \varphi}{\partial t} = -\langle \nabla \varphi, \boldsymbol{f} \rangle - \epsilon \langle \boldsymbol{D}(t), \operatorname{Hess}(\varphi) \rangle}_{\text{backward Kolmogorov PDE}}, \qquad \varphi_1 \hat{\varphi}_1 = \rho_1.$$

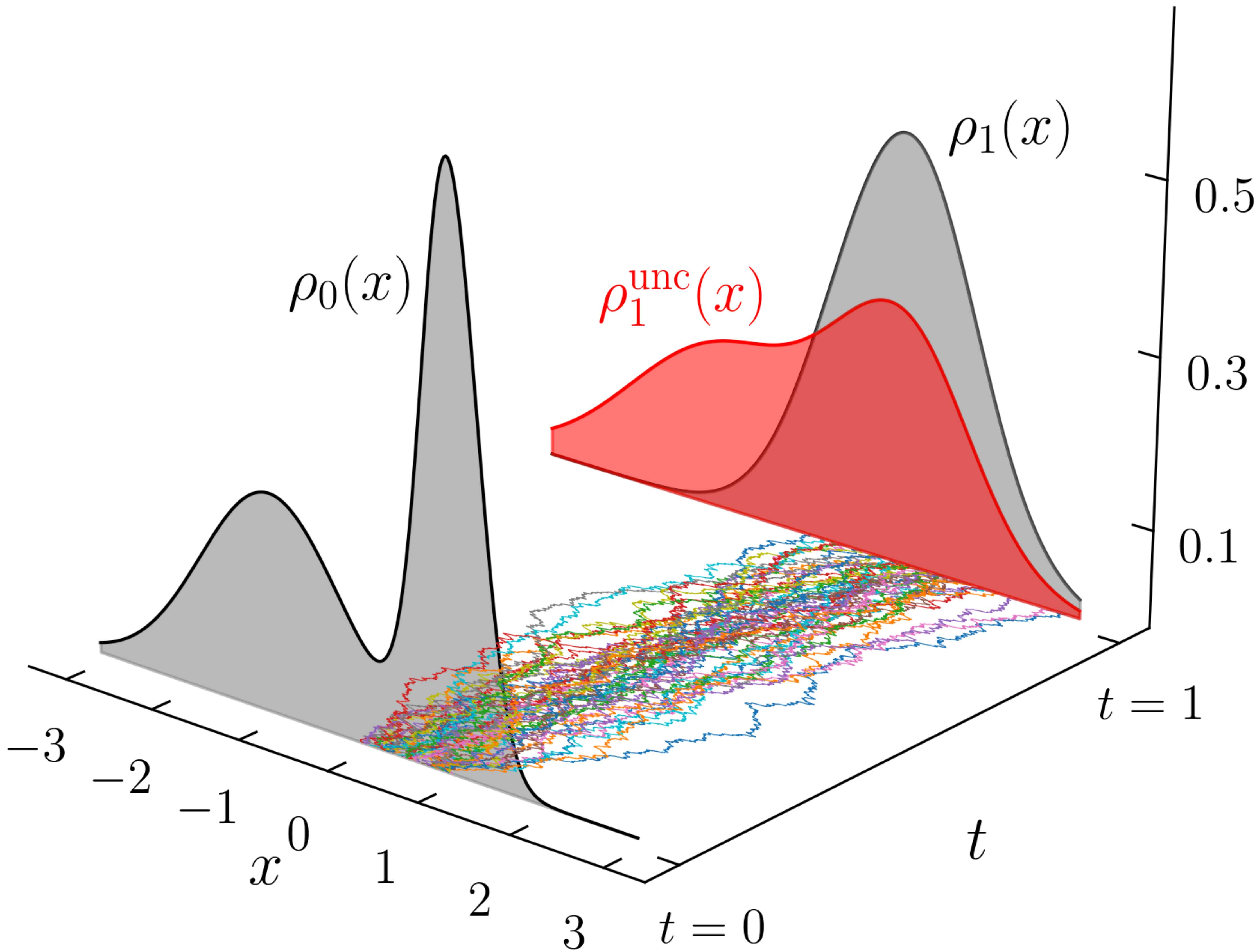
Wasserstein proximal algorithm $\longrightarrow$ fixed point recursion over $(\hat{\varphi}_0, \varphi_1)$

(Contractive in Hilbert metric)

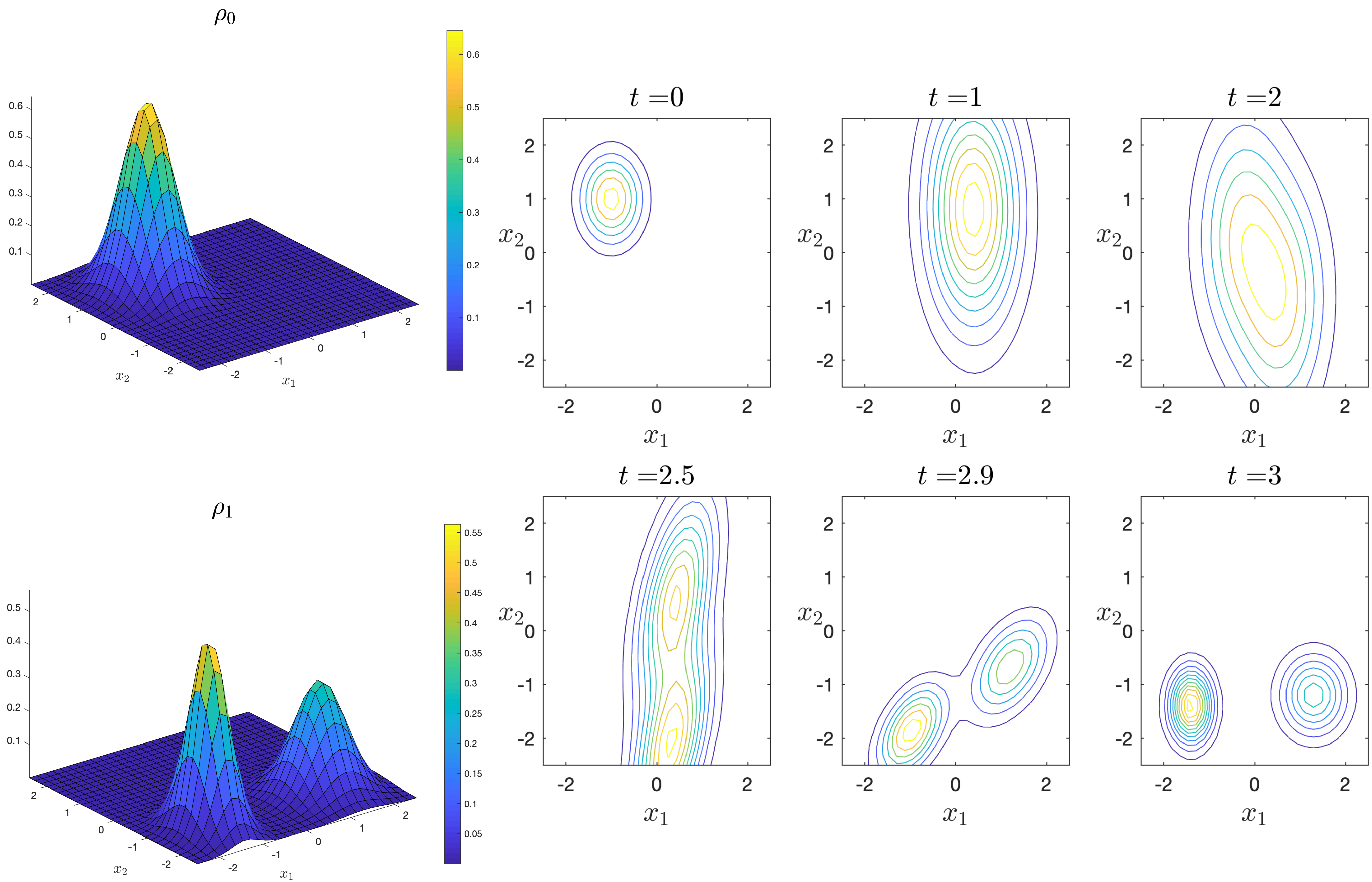
Fixed Point Recursion over $(\hat{\varphi}_0, \varphi_1)$



Feedback Density Control: Zero Prior Dynamics

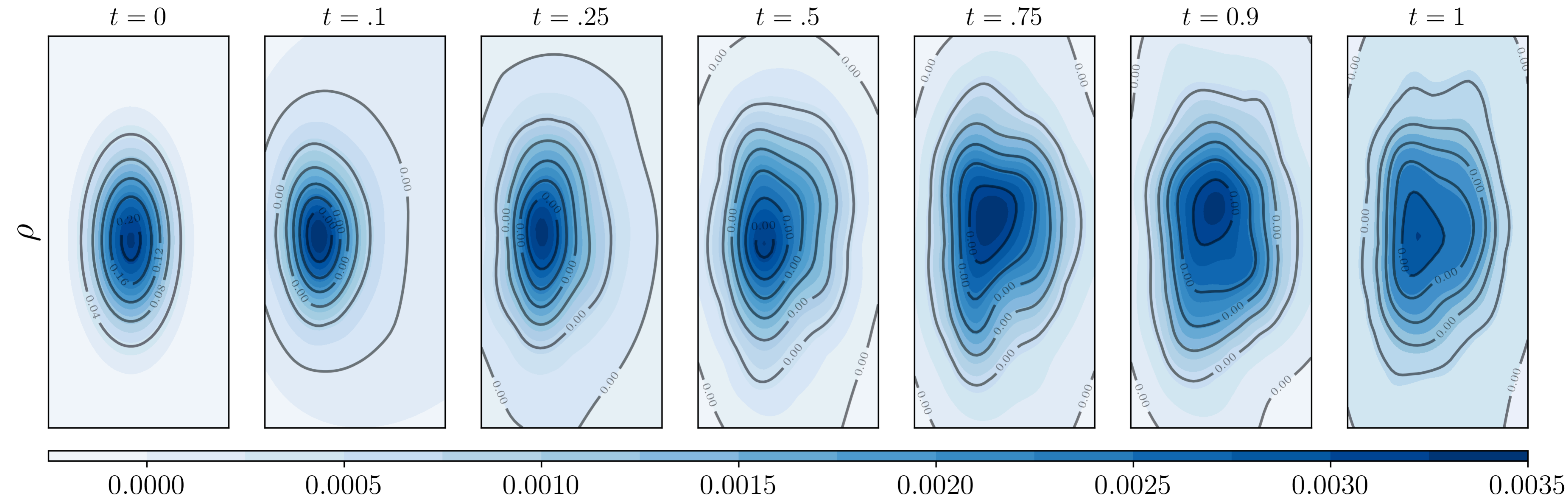


Feedback Density Control: LTI Prior Dynamics

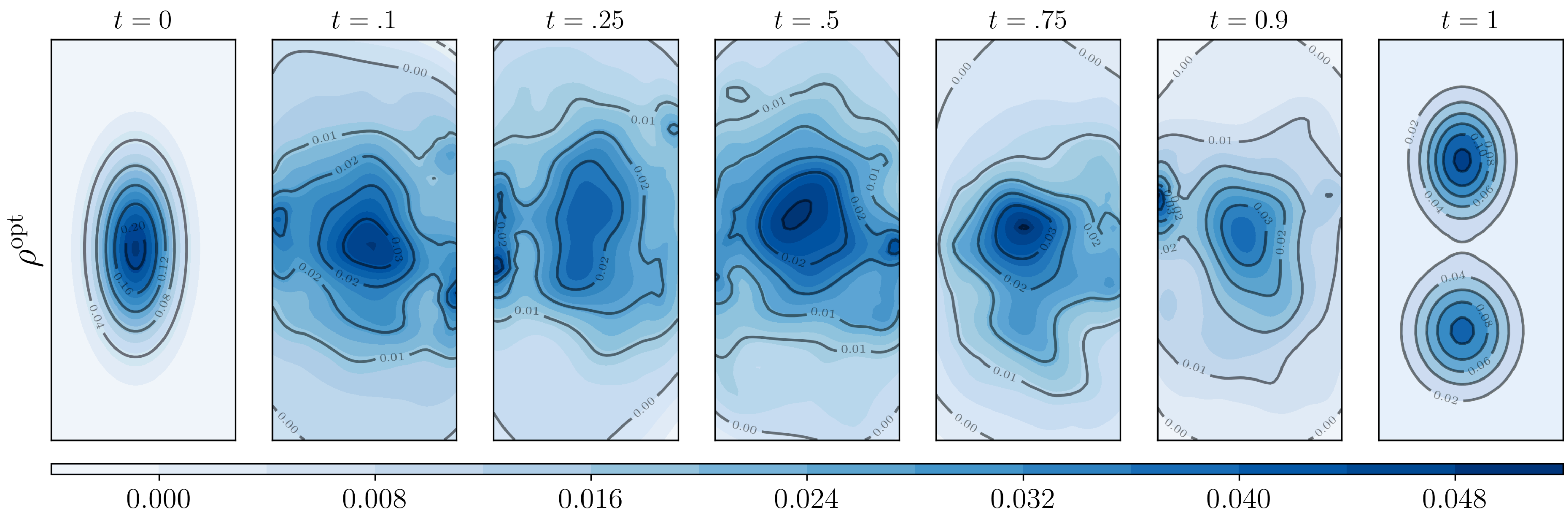


Feedback Density Control: Nonlinear Grad. Drift

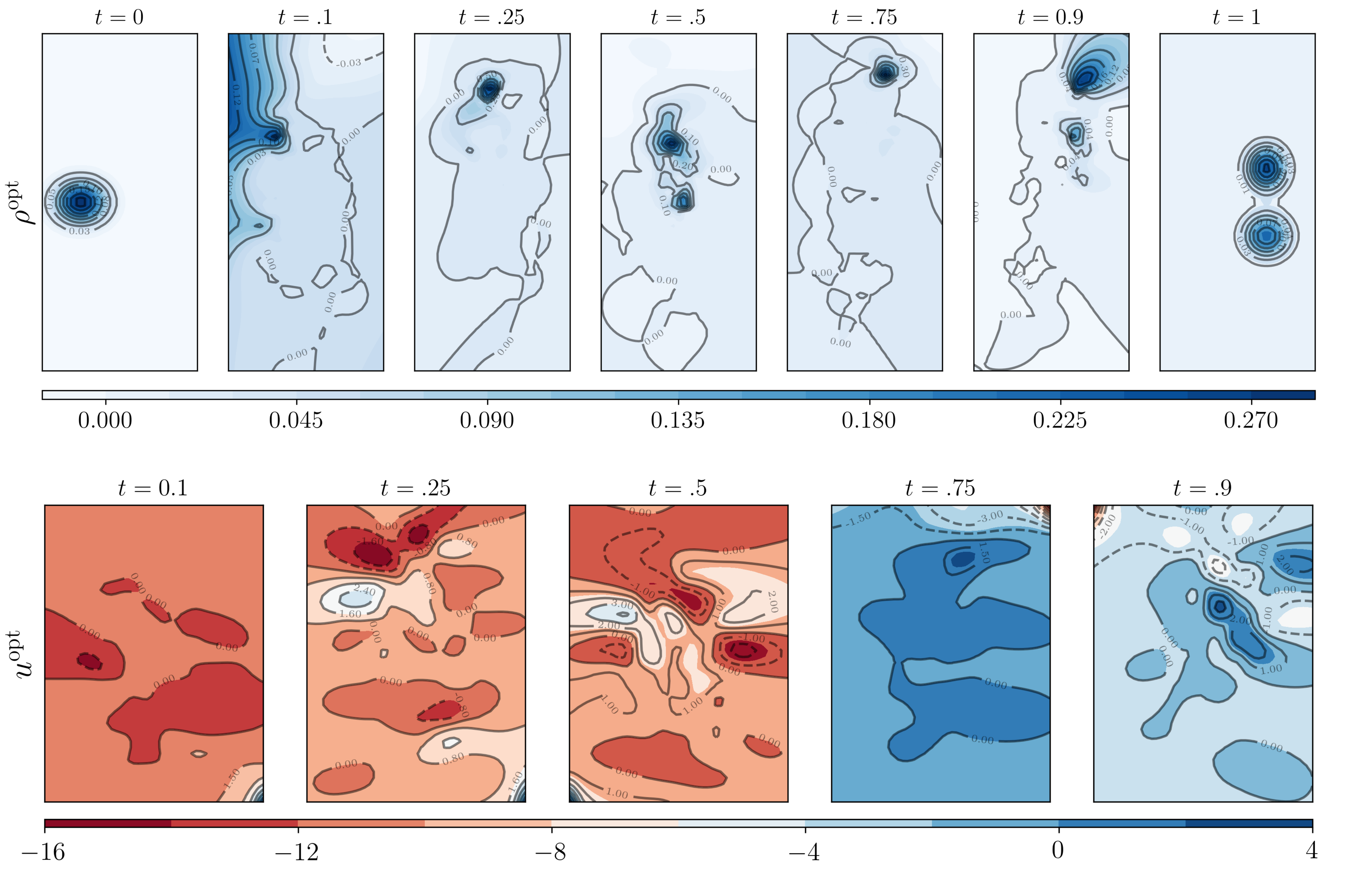
Uncontrolled joint PDF evolution:



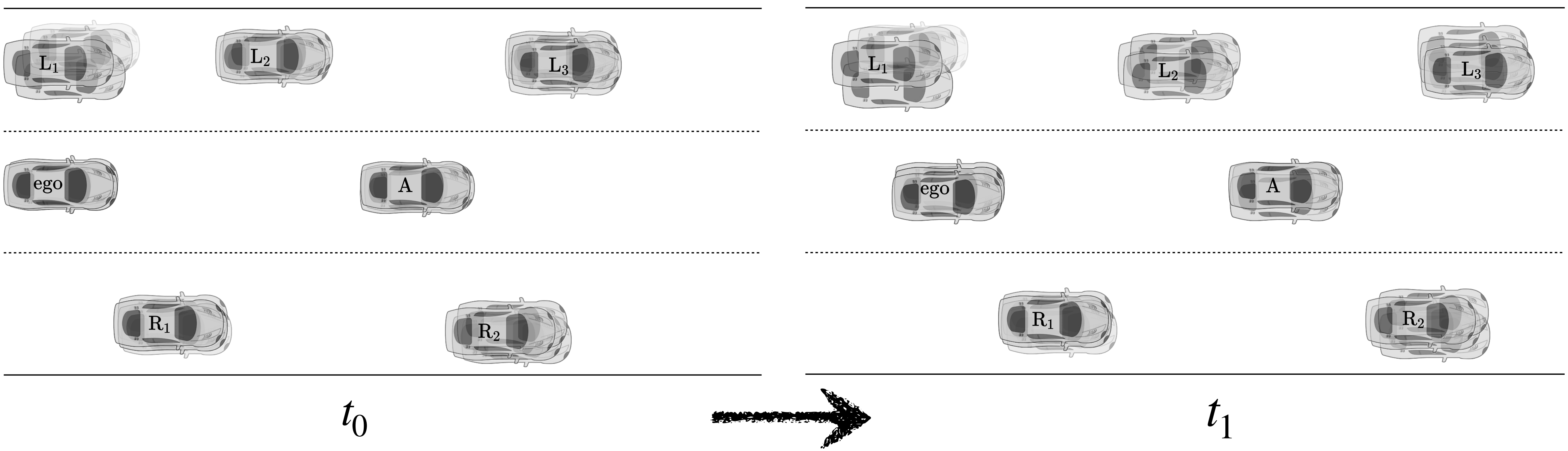
Optimal controlled joint PDF evolution:

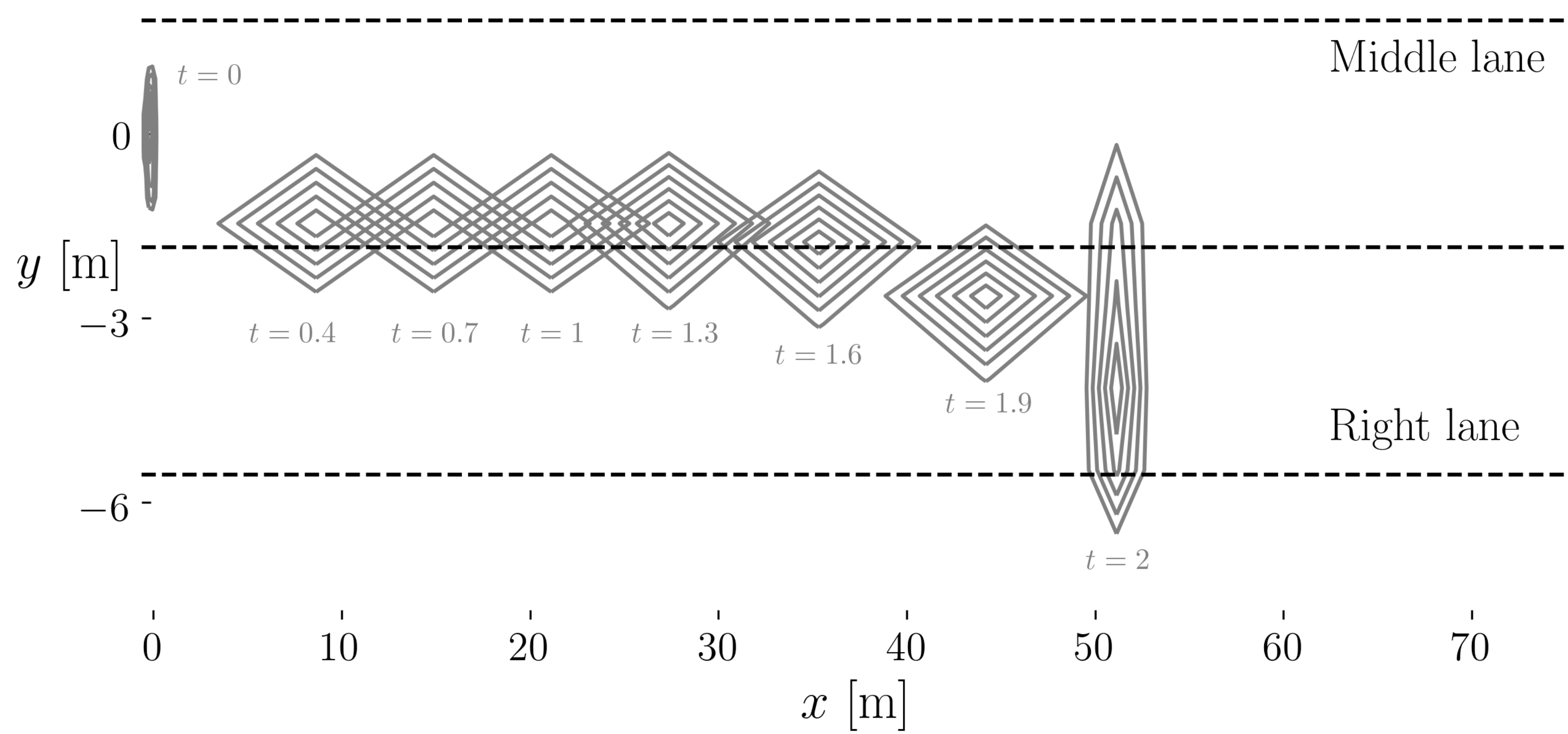


Feedback Density Control: Mixed Conservative-Dissipative Drift



Density Control for Safe Automated Driving





Learning a neural network as Wasserstein gradient flow

Learning Neural Network from Data

(feature vector, label) $= (\boldsymbol{x}_i, y_i) \in \mathbb{R}^d \times \mathbb{R}, \quad i = 1, \ldots, n$

Consider shallow NN: 1 hidden layer with n_{H} neurons

NN parameter vector $\boldsymbol{\theta} := (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_{n_{\mathrm{H}}})^\top \in \mathbb{R}^{p n_{\mathrm{H}}}$

Approximating function:

$$\hat{f}(\boldsymbol{x}, \boldsymbol{\theta}) = \frac{1}{n_{\mathrm{H}}} \sum_{i=1}^{n_{\mathrm{H}}} \Phi(\boldsymbol{x}, \boldsymbol{\theta}_i), \quad \text{example: } \Phi(\boldsymbol{x}, \boldsymbol{\theta}_i) = a_i \sigma\left(\boldsymbol{w}_i^\top \boldsymbol{x} + b_i\right)$$

Population risk functional:

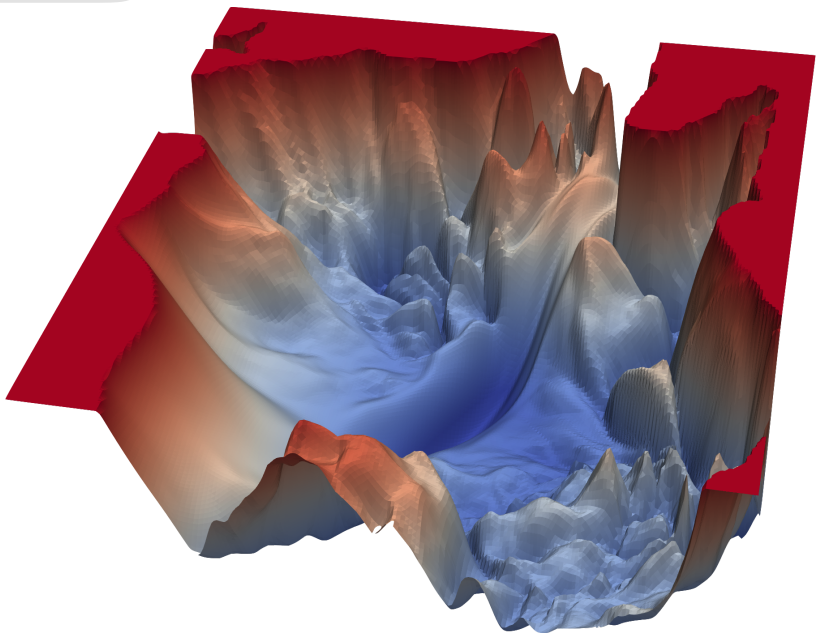
$$R(\hat{f}) = \mathbb{E}_{(\boldsymbol{x},y)}\left[\left(y - \hat{f}(\boldsymbol{x}, \boldsymbol{\theta})\right)^2\right] \approx \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \hat{f}(\boldsymbol{x}_i, \boldsymbol{\theta})\right)^2$$

Learning problem: $\underset{\boldsymbol{\theta} \in \mathbb{R}^{p n_{\mathrm{H}}}}{\text{minimize}}\, R(\hat{f})$

Learning Neural Network from Data

Learning problem: $\underset{\boldsymbol{\theta} \in \mathbb{R}^{p n_{\mathrm{H}}}}{\text{minimize}}\; R(\hat{f}\,)$

Challenge: highly non-convex (many local minima)

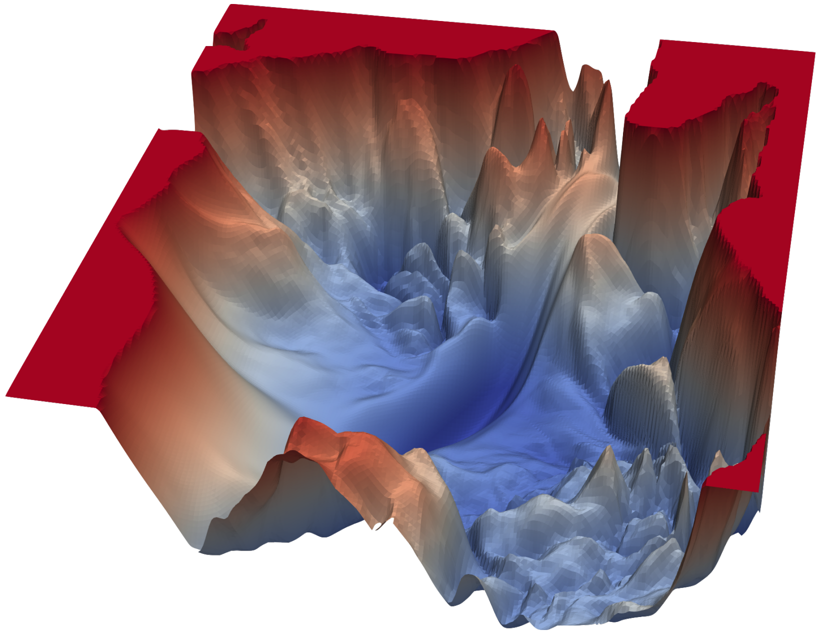


Surprise: SGD and its variants work in practice!!

Learning Neural Network from Data

Learning problem: $\underset{\boldsymbol{\theta}\in\mathbb{R}^{pn_{\mathrm{H}}}}{\text{minimize}}\, R(\hat{f}\,)$

Challenge: highly non-convex (many local minima)



Surprise: SGD and its variants work in practice!!

Good news: emerging theory (starting in 2018!!)

Chizat and Bach (NIPS 2018), Mei, Montanari and Nguyen (PNAS 2018), Rotskoff and Vanden-Eijnden (arXiv:1805.00915, 2018), Williams et al (arXiv:1906.07842, 2019)

Idea: Think of the mean field, i.e., infinite width $(n_{\mathrm{H}} \to \infty)$ limit

$$\hat{f} \equiv \hat{f}\,(\boldsymbol{x}, \rho) = \int_{\mathbb{R}^p} \Phi(\boldsymbol{x}, \boldsymbol{\theta})\rho(\boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta}$$

Then, learning problem: $\underset{\rho\in\mathcal{P}_2(\mathbb{R}^p)}{\text{minimize}}\, R(\hat{f}\,)$

Mean Field Density Dynamics of SGD

Free energy functional: $F(\rho) := R(\hat{f}(\boldsymbol{x}, \rho))$

For quadratic loss:

$$F(\rho) = \underbrace{F_0}_{\text{independent of } \rho} + \underbrace{\int_{\mathbb{R}^p} V(\boldsymbol{\theta})\, \rho(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta}}_{\text{advection potential energy, linear in } \rho} + \underbrace{\int_{\mathbb{R}^p}\int_{\mathbb{R}^p} U(\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}})\, \rho(\boldsymbol{\theta})\rho(\tilde{\boldsymbol{\theta}}) \mathrm{d}\boldsymbol{\theta} \mathrm{d}\tilde{\boldsymbol{\theta}}}_{\text{interaction potential energy, nonlinear in } \rho},$$

where

$$F_0 := \mathbb{E}_{(\boldsymbol{x},y)}\left[y^2\right], \qquad V(\boldsymbol{\theta}) := \mathbb{E}_{(\boldsymbol{x},y)}\left[-2y\Phi(\boldsymbol{x}, \boldsymbol{\theta})\right], \qquad U(\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}}) := \mathbb{E}_{(\boldsymbol{x},y)}\left[\Phi(\boldsymbol{x}, \boldsymbol{\theta})\Phi(\boldsymbol{x}, \tilde{\boldsymbol{\theta}})\right]$$

PDF dynamics for SGD:

$$\frac{\partial \rho}{\partial t} = \nabla \cdot (\rho \nabla (\underbrace{V + U \circledast \rho}_{\frac{\delta F}{\delta \rho}})), \text{ where } (U \circledast \rho)(\boldsymbol{\theta}) := \int_{\mathbb{R}^p} U(\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}}) \rho(\tilde{\boldsymbol{\theta}}) \mathrm{d}\tilde{\boldsymbol{\theta}}$$

This PDE is the gradient flow of functional F w.r.t. the Wasserstein metric W

Wasserstein Proximal Recursion for Training NN

$$\varrho_k(\tau, \boldsymbol{\theta}) = \underset{\varrho \in \mathcal{P}(\mathbb{R}^p)}{\arg\min} \frac{1}{2} \left(W\left(\varrho(\boldsymbol{\theta}), \varrho_{k-1}(\tau, \boldsymbol{\theta})\right)\right)^2 + \tau\, F(\varrho(\boldsymbol{\theta}))$$

$$= \text{prox}_{\tau F}^{W}\left(\varrho_{k-1}\right)$$

Classifying two Gaussians:

$d = 40, n = 100,$

$a = 1, b = 0, \sigma(\cdot) = \tanh(\cdot),$

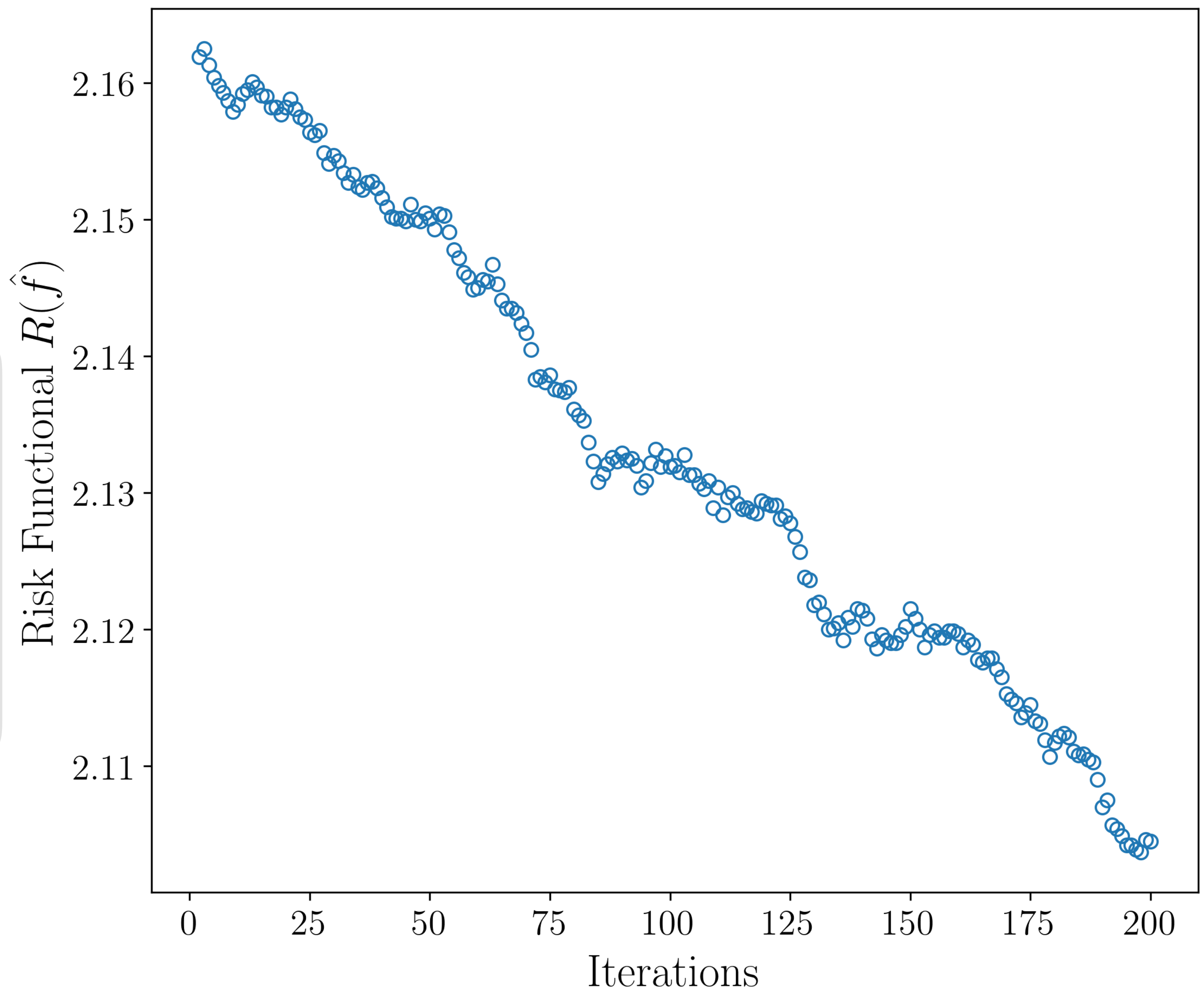
Joint law of $(\boldsymbol{x}, y) \in \mathbb{R}^d \times \mathbb{R}$:

$$\text{Prob}\left(y = +1, \boldsymbol{x} \sim \mathcal{N}\left(\mathbf{0}, (1+\Delta)^2 \boldsymbol{I}_d\right)\right) = \frac{1}{2},$$

$$\text{Prob}\left(y = -1, \boldsymbol{x} \sim \mathcal{N}\left(\mathbf{0}, (1-\Delta)^2 \boldsymbol{I}_d\right)\right) = \frac{1}{2},$$

$\tau = 10^{-3}, n_{\text{sample}} = 100, \Delta = 0.2,$

Noisy SGD with $\beta = \frac{1}{3}$



Take Home Message

Thank You

Support: NSF CITRIS PEOPLE AND ROBOTS Ford