A dynamical systems view of learners, samplers and forecasters

Nisha Chandramoorthy

Collaborators: **Andreas Loukas** (Genentech), **Khashayar Gatmiry** (MIT), **Stefanie Jegelka** (MIT), **Youssef Marzouk** (MIT), **Jeongjin Park** (Georgia Tech), **Thien Le** (MIT), **Evan Montoya** (Georgia Tech)

December 15, 2023

What is the dynamical systems view?

- Unified view of deterministic and stochastic algorithms as discrete-time Markov processes

What is the dynamical systems view?

- Unified view of deterministic and stochastic algorithms as discrete-time Markov processes
- Asymptotic behavior in time is of interest: convergence to fixed points is only one possibility

What is the dynamical systems view?

- Unified view of deterministic and stochastic algorithms as discrete-time Markov processes
- Asymptotic behavior in time is of interest: convergence to fixed points is only one possibility
- Stability of solution manifold/basin of attraction – local stability described with stable/unstable manifolds

What is the dynamical systems view?

- Unified view of deterministic and stochastic algorithms as discrete-time Markov processes
- Asymptotic behavior in time is of interest: convergence to fixed points is only one possibility
- Stability of solution manifold/basin of attraction – local stability described with stable/unstable manifolds
- Potential consequences:
  - reduce model size by exploiting exploration?

What is the dynamical systems view?

- Unified view of deterministic and stochastic algorithms as discrete-time Markov processes
- Asymptotic behavior in time is of interest: convergence to fixed points is only one possibility
- Stability of solution manifold/basin of attraction – local stability described with stable/unstable manifolds
- Potential consequences:
  - reduce model size by exploiting exploration?
  - dynamics-aware generalization

Intersections with areas of statistics

Generalization: The performance of a learning algorithm on unseen data (typically from the same distribution)

C, Loukas, Gatmiry and Jegelka, NeurIPS 2022

Intersections with areas of statistics

Generalization: The performance of a learning algorithm on unseen data (typically from the same distribution)

C, Loukas, Gatmiry and Jegelka, NeurIPS 2022

Sampling: an algorithm to sample from a target distribution (e.g., a Bayesian posterior) when it is partially specified

C, Schäfer and Marzouk, 2023

Intersections with areas of statistics

Generalization: The performance of a learning algorithm on unseen data (typically from the same distribution)

C, Loukas, Gatmiry and Jegelka, NeurIPS 2022

Sampling: an algorithm to sample from a target distribution (e.g., a Bayesian posterior) when it is partially specified

C, Schäfer and Marzouk, 2023

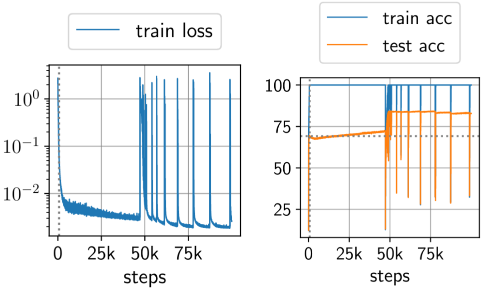
Forecasting: predicting chaotic timeseries from data

Park and C, 2023

More **long-term** stability in the training algorithm leads to better generalization

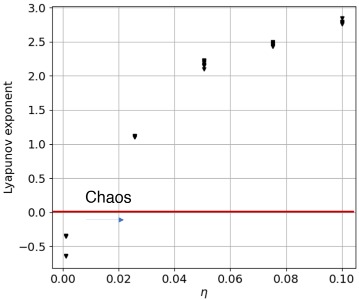
Non-converging optimization

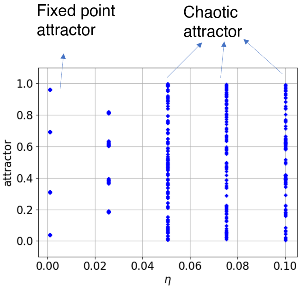
What happens in training beyond the stopping point?



Courtesy: [Lyu Li Arora 2022]. Recent interest [Kong and Tao 2021, Cohen et al 2021, Lobacheva et al 2021, Zhang Li Sra Jadbabaie 2022] in non-converging training algorithms

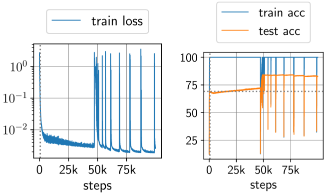
Training algorithms as nonlinear dynamical systems





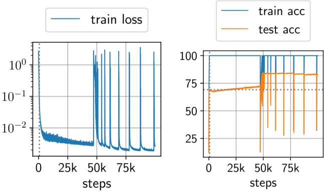
- heavy-tailed fluctuations in SGD leads to better generalization [**Martin and Mahoney 2017, 2019, 2020**]
- generalization linked to fractal dimension of SGD attractor [**Şimşekli et al 2020**], data-dependent generalization [**Xu and Raginsky 2017**] based on Fernique-Talagrand functional [**Hodgkinson et al 2022**]

Non-converging optimization



Courtesy: [Lyu Li Arora 2022]. Recent interest [Kong and Tao 2021, Cohen et al 2021, Lobacheva et al 2021, Zhang Li Sra Jadbabaie 2022] in non-converging training algorithms

Non-converging optimization



Courtesy: [Lyu Li Arora 2022]. Recent interest [Kong and Tao 2021, Cohen et al 2021, Lobacheva et al 2021, Zhang Li Sra Jadbabaie 2022] in non-converging training algorithms

(Q1) How can we *define* and *study* the generalization properties of a non-converging learning algorithm?

(Q2) Can the statistical/ergodic properties of the algorithm *predict* its generalization performance?

SGD/GD dynamics on weight space:

$$w_{t+1} = w_t - \eta \hat{\nabla} L_S(w_t),$$

where

- $w_t \in M$ are the weights at time $t \in \mathbb{Z}^+$

SGD/GD dynamics on weight space:

$$w_{t+1} = w_t - \eta \hat{\nabla} L_S(w_t),$$

where

- $w_t \in M$ are the weights at time $t \in \mathbb{Z}^+$
- S is a set of n training samples $z_1, \cdots, z_n$ iid according to $\mathcal{D}$

SGD/GD dynamics on weight space:

$$w_{t+1} = w_t - \eta \hat{\nabla} L_S(w_t),$$

where

- $w_t \in M$ are the weights at time $t \in \mathbb{Z}^+$
- S is a set of n training samples $z_1, \cdots, z_n$ iid according to $\mathcal{D}$
- $L_S(w) = (1/n) \sum_{z \in S} \ell(z, w)$ is the sample average of the loss $\ell(\cdot, \cdot)$.

SGD/GD dynamics on weight space:

$$w_{t+1} = w_t - \eta \hat{\nabla} L_S(w_t),$$

where

- $w_t \in M$ are the weights at time $t \in \mathbb{Z}^+$
- S is a set of n training samples $z_1, \cdots, z_n$ iid according to $\mathcal{D}$
- $L_S(w) = (1/n) \sum_{z \in S} \ell(z, w)$ is the sample average of the loss $\ell(\cdot, \cdot)$.
- $\hat{\nabla} L_S(w_t)$ is the estimate of the weight space gradient of L_S.

SGD/GD dynamics on weight space:

$$w_{t+1} = w_t - \eta \hat{\nabla} L_S(w_t),$$

where

- $w_t \in M$ are the weights at time $t \in \mathbb{Z}^+$
- S is a set of n training samples $z_1, \cdots, z_n$ iid according to $\mathcal{D}$
- $L_S(w) = (1/n) \sum_{z \in S} \ell(z, w)$ is the sample average of the loss $\ell(\cdot, \cdot)$.
- $\hat{\nabla} L_S(w_t)$ is the estimate of the weight space gradient of L_S.

In general, deterministic/stochastic nonlinear dynamics on compact set. No guarantee of convergence to fixed points.

SGD/GD dynamics on weight space:

$$w_{t+1} = w_t - \eta \hat{\nabla} L_S(w_t),$$

where

- $w_t \in M$ are the weights at time $t \in \mathbb{Z}^+$
- S is a set of n training samples $z_1, \cdots, z_n$ iid according to $\mathcal{D}$
- $L_S(w) = (1/n) \sum_{z \in S} \ell(z, w)$ is the sample average of the loss $\ell(\cdot, \cdot)$.
- $\hat{\nabla} L_S(w_t)$ is the estimate of the weight space gradient of L_S.

In general, deterministic/stochastic nonlinear dynamics on compact set. No guarantee of convergence to fixed points. There exist multiple invariant, ergodic distributions on weight space, M.

Ergodic properties of training

A probability measure μ on M is ergodic for the training dynamics if for all continuous functions $f : M \to \mathbb{R}$, and μ-a.e. w_0,

$$(1/T) \sum_{t=0}^{T-1} f(w_t) \to \mathbb{E}_{w\sim\mu}[f(w)].$$

Ergodic properties of training

A probability measure μ on M is ergodic for the training dynamics if for all continuous functions $f : M \to \mathbb{R}$, and μ-a.e. w_0,

$$(1/T) \sum_{t=0}^{T-1} f(w_t) \to \mathbb{E}_{w \sim \mu}[f(w)].$$

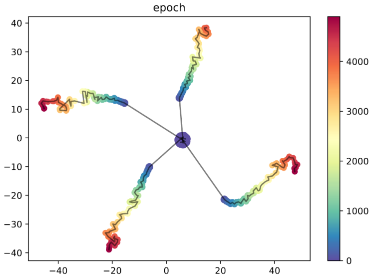
In general, there are many invariant, ergodic measures.

Ergodic properties of training

A probability measure μ on M is ergodic for the training dynamics if for all continuous functions $f : M \to \mathbb{R}$, and μ-a.e. w_0,

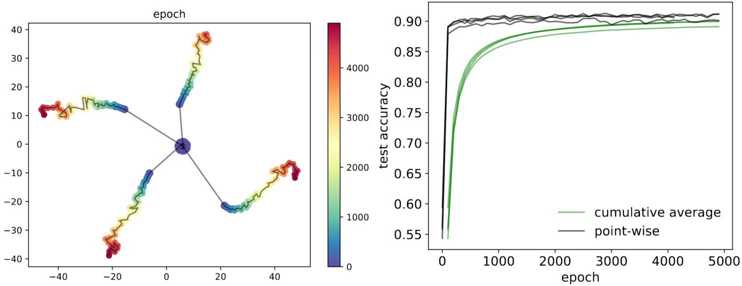
$$(1/T) \sum_{t=0}^{T-1} f(w_t) \to \mathbb{E}_{w \sim \mu}[f(w)].$$

In general, there are many invariant, ergodic measures.



Convergence of loss time-averages

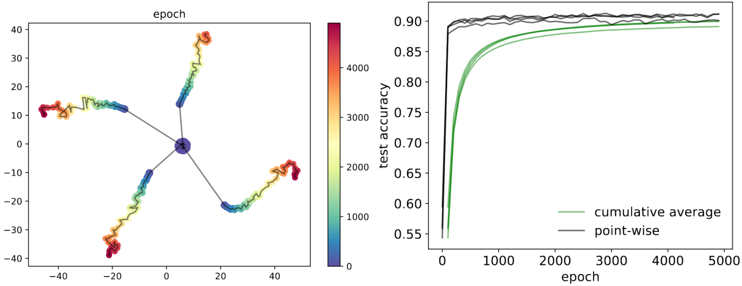
Assumption 1: For almost every w_0 and every z, time-average of $\ell(z, \cdot)$ converges to a constant $\langle \ell_z \rangle_S$, independent of w_0.



Orbits of four different initializations of a VGG16 training with SGD.

Convergence of loss time-averages

Assumption 1: For almost every w_0 and every z, time-average of $\ell(z, \cdot)$ converges to a constant $\langle \ell_z \rangle_S$, independent of w_0.



Orbits of four different initializations of a VGG16 training with SGD.

Assumption allows us to extend algorithmic stability to *statistical algorithmic stability* (SAS).

Statistical Algorithmic Stability

Classical algorithmic stability [Bousquet and Elisseeff 2002]:

$$\beta := \sup_{z} \sup_{S,S'} |\ell(z, w_S^*) - \ell(z, w_{S'}^*)|.$$

Statistical Algorithmic Stability

Classical algorithmic stability [Bousquet and Elisseeff 2002]:

$$\beta := \sup_{z} \sup_{S,S'} |\ell(z, w_S^*) - \ell(z, w_{S'}^*)|.$$

Statistical Algorithmic Stability (SAS): We say an algorithm is SAS with coefficient β if

$$\beta := \sup_{z} \sup_{S,S'} |\langle \ell_z \rangle_S - \langle \ell_z \rangle_{S'}|.$$

Statistical Algorithmic Stability

Classical algorithmic stability [Bousquet and Elisseeff 2002]:

$$\beta := \sup_{z} \sup_{S,S'} |\ell(z, w_S^*) - \ell(z, w_{S'}^*)|.$$

Statistical Algorithmic Stability (SAS): We say an algorithm is SAS with coefficient β if

$$\beta := \sup_{z} \sup_{S,S'} |\langle \ell_z \rangle_S - \langle \ell_z \rangle_{S'}|.$$

The higher the value of β, the lower the statistical stability.

Statistical Algorithmic Stability

Classical algorithmic stability [Bousquet and Elisseeff 2002]:

$$\beta := \sup_{z} \sup_{S,S'} |\ell(z, w_S^*) - \ell(z, w_{S'}^*)|.$$

Statistical Algorithmic Stability (SAS): We say an algorithm is SAS with coefficient β if

$$\beta := \sup_{z} \sup_{S,S'} |\langle \ell_z \rangle_S - \langle \ell_z \rangle_{S'}|.$$

The higher the value of β, the lower the statistical stability. Unlike classical algorithmic stability, SAS

Statistical Algorithmic Stability

Classical algorithmic stability [Bousquet and Elisseeff 2002]:

$$\beta := \sup_{z} \sup_{S,S'} |\ell(z, w_S^*) - \ell(z, w_{S'}^*)|.$$

Statistical Algorithmic Stability (SAS): We say an algorithm is SAS with coefficient β if

$$\beta := \sup_{z} \sup_{S,S'} |\langle \ell_z \rangle_S - \langle \ell_z \rangle_{S'}|.$$

The higher the value of β, the lower the statistical stability. Unlike classical algorithmic stability, SAS

- applies to non-converging learning algorithms

Statistical Algorithmic Stability

Classical algorithmic stability [Bousquet and Elisseeff 2002]:

$$\beta := \sup_{z} \sup_{S,S'} |\ell(z, w_S^*) - \ell(z, w_{S'}^*)|.$$

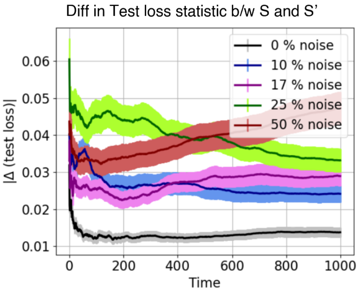
Statistical Algorithmic Stability (SAS): We say an algorithm is SAS with coefficient β if

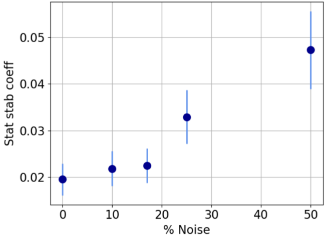
$$\beta := \sup_{z} \sup_{S,S'} |\langle \ell_z \rangle_S - \langle \ell_z \rangle_{S'}|.$$

The higher the value of β, the lower the statistical stability. Unlike classical algorithmic stability, SAS

- applies to non-converging learning algorithms
- is constant on network function/parameter space

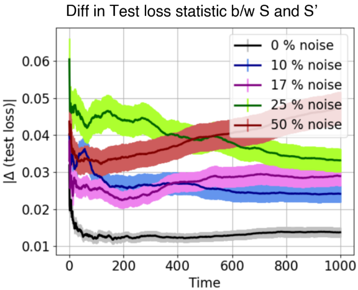
Numerical approximation of β for SGD on VGG16 model trained on CIFAR10

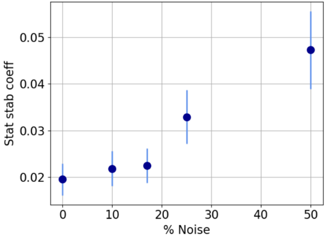




Noisy CIFAR10 labels.
Anticlockwise: Sample mean over 45 (S, S') pairs, with error bars, of time-averaged test loss difference. Lower bound on β with error bars computed as sample mean. Test loss vs. time (epoch).

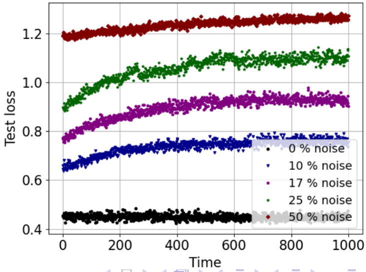
Numerical approximation of β for SGD on VGG16 model trained on CIFAR10





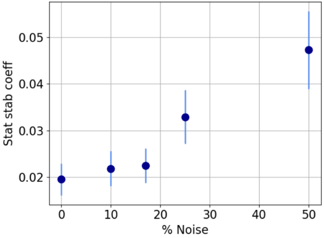
Noisy CIFAR10 labels.

Anticlockwise: Sample mean over 45 (S, S') pairs, with error bars, of time-averaged test loss difference. Lower bound on β with error bars computed as sample mean. Test loss vs. time (epoch).



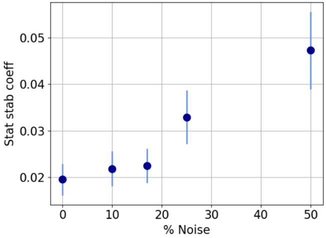
Observation: more the SAS of training, better generalization

Lower bound on β

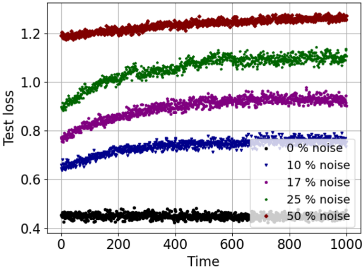


Observation: more the SAS of training, better generalization

Lower bound on β



Generalization error



Generalization of a non-converging algorithm

- Training error, $\hat{R}_S := (1/n) \sum_{z \in S} \langle \ell_z \rangle_S$

Generalization of a non-converging algorithm

- Training error, $\hat{R}_S := (1/n) \sum_{z \in S} \langle \ell_z \rangle_S$
- Test/generalization error, $R_S := \mathbb{E}_{z \sim \mathcal{D}} \langle \ell_z \rangle_S$.

Generalization of a non-converging algorithm

- Training error, $\hat{R}_S := (1/n)\sum_{z\in S}\langle \ell_z \rangle_S$
- Test/generalization error, $R_S := \mathbb{E}_{z\sim\mathcal{D}}\langle \ell_z \rangle_S$.

Theorem 1 **(SAS implies generalization)** For an algorithm with SAS coefficient β and large # of samples n, the *generalization gap* = $R_S - \hat{R}_S = \mathcal{O}(\beta\sqrt{n})$ with high probability.

Generalization of a non-converging algorithm

- Training error, $\hat{R}_S := (1/n) \sum_{z \in S} \langle \ell_z \rangle_S$
- Test/generalization error, $R_S := \mathbb{E}_{z \sim \mathcal{D}} \langle \ell_z \rangle_S$.

Theorem 1 **(SAS implies generalization)** For an algorithm with SAS coefficient β and large # of samples n, the *generalization gap* = $R_S - \hat{R}_S = \mathcal{O}(\beta\sqrt{n})$ with high probability.

Smaller $\beta \equiv$ more SAS $\implies$ better generalization

Generalization of a non-converging algorithm

- Training error, $\hat{R}_S := (1/n) \sum_{z \in S} \langle \ell_z \rangle_S$
- Test/generalization error, $R_S := \mathbb{E}_{z \sim \mathcal{D}} \langle \ell_z \rangle_S$.

Theorem 1 **(SAS implies generalization)** For an algorithm with SAS coefficient β and large # of samples n, the *generalization gap* = $R_S - \hat{R}_S = \mathcal{O}(\beta\sqrt{n})$ with high probability.

Smaller $\beta \equiv$ more SAS $\implies$ better generalization

What makes an algorithm statistically stable?

Pointwise approach [**Hardt et al 2016**] toward algorithmic stability

$$w_{t+1}^{S} - w_{t+1}^{S'} = w_t^{S} - w_t^{S'} - \eta(\hat{\nabla} L_S(w_t^S) - \hat{\nabla} L_{S'}(w_t^{S'}))$$

- Uninformative for SAS, which is a time-independent notion

What makes an algorithm statistically stable?

Pointwise approach [**Hardt et al 2016**] toward algorithmic stability

$$w_{t+1}^{S} - w_{t+1}^{S'} = w_t^{S} - w_t^{S'} - \eta(\hat{\nabla} L_S(w_t^{S}) - \hat{\nabla} L_{S'}(w_t^{S'}))$$

- Uninformative for SAS, which is a time-independent notion
- Early stopping based on the upper bound does not apply to non-converging algorithms

What makes an algorithm statistically stable?

Pointwise approach [**Hardt et al 2016**] toward algorithmic stability

$$w_{t+1}^{S} - w_{t+1}^{S'} = w_t^{S} - w_t^{S'} - \eta(\hat{\nabla} L_S(w_t^S) - \hat{\nabla} L_{S'}(w_t^{S'}))$$

- Uninformative for SAS, which is a time-independent notion
- Early stopping based on the upper bound does not apply to non-converging algorithms
- Must take global **(operator-theoretic)** approach to SAS-based generalization

Predicting generalization gap from timeseries data

Theorem 2 (**Slower convergence of loss statistics implies larger** β) Let λ be the slowest mixing rate of the transition operators on loss space. Then, the corresponding training algorithm with n samples has SAS coefficient

$$\beta = \mathcal{O}\left(\frac{1}{n}\frac{L_D}{1-\lambda}\right),$$

where $L_D = \sup_w \mathrm{Lip}(\nabla \ell(\cdot, w))$

Predicting generalization gap from timeseries data

Theorem 2 (**Slower convergence of loss statistics implies larger** β) Let λ be the slowest mixing rate of the transition operators on loss space. Then, the corresponding training algorithm with n samples has SAS coefficient

$$\beta = \mathcal{O}\left(\frac{1}{n}\frac{L_D}{1-\lambda}\right),$$

where $L_D = \sup_w \mathrm{Lip}(\nabla \ell(\cdot, w))$

- exploit perturbation theory of uniformly ergodic Markov chains (see e.g. [Mitraphanov 2005])

Predicting generalization gap from timeseries data

Theorem 2 (**Slower convergence of loss statistics implies larger** β) Let λ be the slowest mixing rate of the transition operators on loss space. Then, the corresponding training algorithm with n samples has SAS coefficient

$$\beta = \mathcal{O}\left(\frac{1}{n}\frac{L_D}{1-\lambda}\right),$$

where $L_D = \sup_w \mathrm{Lip}(\nabla \ell(\cdot, w))$

- exploit perturbation theory of uniformly ergodic Markov chains (see e.g. [Mitraphanov 2005])
- Under conditions of the above result, $\beta \sim \mathcal{O}(1/n)$.

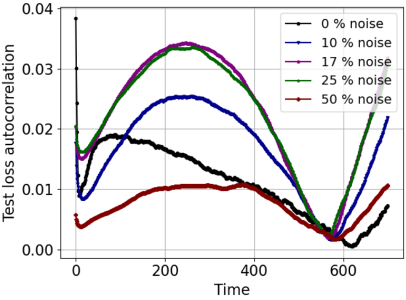
Numerical verification of the connection between speed of convergence of statistics and SAS, and hence generalization

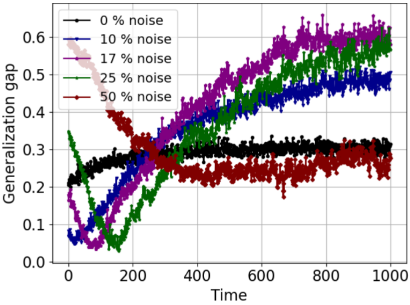
Learning algorithms in which loss time-averages converge slower, e.g., correlations in the loss persist, are less SAS.

Numerical verification of the connection between speed of convergence of statistics and SAS, and hence generalization

Learning algorithms in which loss time-averages converge slower, e.g., correlations in the loss persist, are less SAS.

SGD with constant step size of 0.01 on ResNet18 model trained on corrupted CIFAR10 dataset.





More long-term stability in the training algorithm leads to better generalization

Goal: Transport-based Bayesian Inference and Sampling

Goal: Transport-based Bayesian Inference and Sampling

Today: New **transport** construction

.

Goal: Transport-based Bayesian Inference and Sampling

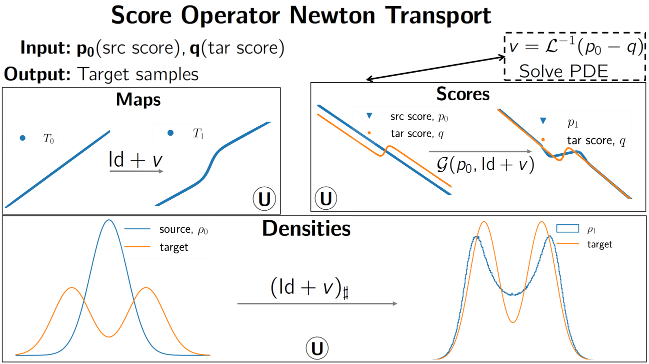
Today: New **transport** construction

.

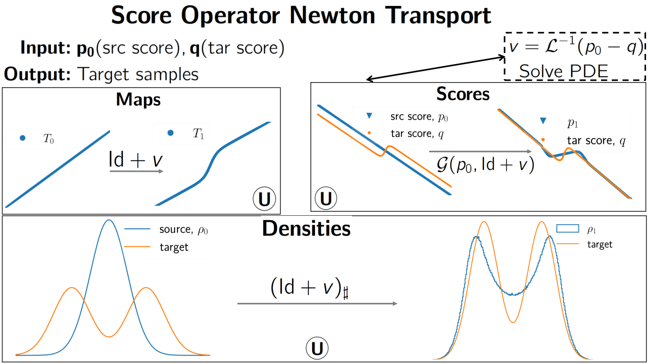
score of a probability distribution with density $\rho := \nabla \log \rho$

Sampling: an algorithm to sample from a target distribution (e.g., a Bayesian posterior) when it is partially specified through samples or a statistical model

Sampling: an algorithm to sample from a target distribution (e.g., a Bayesian posterior) when it is partially specified through samples or a statistical model



Sampling: an algorithm to sample from a target distribution (e.g., a Bayesian posterior) when it is partially specified through samples or a statistical model



Introduce a new **transport map** construction

Sampling via measure transport

- Target measure: ν with density ρ^{ν}.
- Tractable source measure μ with density ρ^{μ}.
- $\mathrm{supp}(\mu) = \mathbb{X}$ and $\mathrm{supp}(\nu) = \mathbb{Y}$.

Sampling via measure transport

- Target measure: ν with density ρ^{ν}.
- Tractable source measure μ with density ρ^{μ}.
- $\mathrm{supp}(\mu) = \mathbb{X}$ and $\mathrm{supp}(\nu) = \mathbb{Y}$.

A transport map $T : \mathbb{X} \to \mathbb{Y}$ is an invertible transformation such that $T_{\sharp}\mu = \nu$.

The score operator

Goal: new transport map that exploits availability of **scores**.

The score operator

Goal: new transport map that exploits availability of **scores**.

Idea: Define an infinite-dimensional score matching problem

The score operator

Goal: new transport map that exploits availability of **scores**.

Idea: Define an infinite-dimensional score matching problem

Change of variables/pushforward operation:

$$\rho^{\nu} = \frac{\rho^{\mu} \circ T^{-1}}{|\det \nabla T| \circ T^{-1}}$$

Pushforward operation on **scores**:

$$\begin{aligned}\mathcal{G}(s, U) &= \left(s(\nabla U)^{-1} - \nabla \log |\det \nabla U| (\nabla U)^{-1}\right) \circ U^{-1} \\ &= \left(s\,(\nabla U)^{-1} - \mathrm{tr}\left((\nabla U)^{-1} \nabla^2 U\right)(\nabla U)^{-1}\right) \circ U^{-1},\end{aligned}$$

The score operator

Goal: new transport map that exploits availability of **scores**.

Idea: Define an infinite-dimensional score matching problem

Change of variables/pushforward operation:

$$\rho^{\nu} = \frac{\rho^{\mu} \circ T^{-1}}{|\det \nabla T| \circ T^{-1}}$$

Pushforward operation on **scores**:

$$\begin{aligned}
\mathcal{G}(s, U) &= \left(s (\nabla U)^{-1} - \nabla \log |\det \nabla U| (\nabla U)^{-1} \right) \circ U^{-1} \\
&= \left(s \, (\nabla U)^{-1} - \mathrm{tr} \left((\nabla U)^{-1} \nabla^2 U \right) (\nabla U)^{-1} \right) \circ U^{-1},
\end{aligned}$$

The score operator

Some properties of $\mathcal{G}$

- $\mathcal{G}(s, \mathrm{Id}) = s$
- $\mathcal{G}(s, U_2 \circ U_1) = \mathcal{G}(\mathcal{G}(s, U_1), U_2)$

The score operator

Some properties of $\mathcal{G}$

- $\mathcal{G}(s, \mathrm{Id}) = s$
- $\mathcal{G}(s, U_2 \circ U_1) = \mathcal{G}(\mathcal{G}(s, U_1), U_2)$

Infinite-dimensional score matching problem: Find **T** such that

$$\mathcal{G}(p, T) = q,$$

where,
p: Source score = $\nabla \log \rho^{\mu}$
q: Target score = $\nabla \log \rho^{\nu}$.

The score operator

Some properties of $\mathcal{G}$

- $\mathcal{G}(s, \mathrm{Id}) = s$
- $\mathcal{G}(s, U_2 \circ U_1) = \mathcal{G}(\mathcal{G}(s, U_1), U_2)$

Infinite-dimensional score matching problem: Find **T** such that

$$\mathcal{G}(p, T) = q,$$

where,
p: Source score = $\nabla \log \rho^{\mu}$
q: Target score = $\nabla \log \rho^{\nu}$.

- Want to avoid parameterization
- Use Newton-type method

A zero of the score residual

Infinite-dimensional score matching problem: Find a zero $\in \mathcal{C}^2(\mathbb{X}, \mathbb{Y})$ of the **score residual** operator

$$\mathcal{R}(T) := \mathcal{G}(p, T) - q$$

A zero of the score residual

Infinite-dimensional score matching problem: Find a zero $\in \mathcal{C}^2(\mathbb{X}, \mathbb{Y})$ of the **score residual** operator

$$\mathcal{R}(T) := \mathcal{G}(p, T) - q$$

Expand $\mathcal{G}$ about (q, Id)

$$\begin{aligned}\mathcal{G}(p, T) = \mathcal{G}(q, \mathrm{Id}) + D_1\mathcal{G}(q, \mathrm{Id})(p - q) + D_2\mathcal{G}(q, \mathrm{Id})(T - \mathrm{Id}) \\ + \Delta(p, T)\end{aligned}$$

A zero of the score residual

Infinite-dimensional score matching problem: Find a zero $\in \mathcal{C}^2(\mathbb{X}, \mathbb{Y})$ of the **score residual** operator

$$\mathcal{R}(T) := \mathcal{G}(p, T) - q$$

Expand $\mathcal{G}$ about (q, Id)

$$\begin{aligned}\mathcal{G}(p, T) = {} & \mathcal{G}(q, \mathrm{Id}) + D_1\mathcal{G}(q, \mathrm{Id})(p - q) + D_2\mathcal{G}(q, \mathrm{Id})(T - \mathrm{Id}) \\ & + \Delta(p, T)\end{aligned}$$

$$\mathcal{L}(q)\, v := -D_2\mathcal{G}(q, \mathrm{Id})\, v = \nabla q\, v + q\, \nabla v + \mathrm{tr}(\nabla^2 v).$$

The zero of the score residual operator

The **linearized** score-matching problem: **Elliptic** PDE system

$$(p - q) = \mathcal{L}\, v \tag{1}$$

Newton-type step:

$$-(q - p_n) = \mathcal{L}(q)v_n = (\nabla q)v_n + q(\nabla v_n) + \mathrm{tr}(\nabla^2 v_n).$$

The zero of the score residual operator

The **linearized** score-matching problem: **Elliptic** PDE system

$$(p - q) = \mathcal{L}\, v \tag{1}$$

Newton-type step:

$$-(q - p_n) = \mathcal{L}(q) v_n = (\nabla q) v_n + q(\nabla v_n) + \mathrm{tr}(\nabla^2 v_n).$$

Newton-type update:

$$T_{n+1} \leftarrow (\mathrm{Id} + v_n) \circ T_n$$
$$p_{n+1} \leftarrow \mathcal{G}(p_n, \mathrm{Id} + v_n).$$

Score Operator Newton (SCONE) iteration for transport maps

Newton-type step:

$$-(q - p_n) = \mathcal{L}(q)v_n = (\nabla q)v_n + q(\nabla v_n) + \mathrm{tr}(\nabla^2 v_n).$$

- Inspired by Kolmogorov-Arnold-Moser iteration in dynamical systems theory, and Nash-Moser iteration in PDEs.
- Conceptually different from empirical, parametric score-matching [**Koehler et al 2022**]

Score Operator Newton (SCONE) iteration for transport maps

Newton-type step:

$$-(q - p_n) = \mathcal{L}(q) v_n = (\nabla q) v_n + q(\nabla v_n) + \mathrm{tr}(\nabla^2 v_n).$$

Newton-type update:

$$\begin{aligned} T_{n+1} &\leftarrow (\mathrm{Id} + v_n) \circ T_n \\ p_{n+1} &\leftarrow \mathcal{G}(p_n, \mathrm{Id} + v_n). \end{aligned}$$

- Inspired by Kolmogorov-Arnold-Moser iteration in dynamical systems theory, and Nash-Moser iteration in PDEs.
- Conceptually different from empirical, parametric score-matching [**Koehler et al 2022**]

Score Operator Newton (SCONE) iteration for transport maps

Newton-type step:

$$-(q - p_n) = \mathcal{L}(q)v_n = (\nabla q)v_n + q(\nabla v_n) + \mathrm{tr}(\nabla^2 v_n).$$

Newton-type update:

$$\begin{aligned} T_{n+1} &\leftarrow (\mathrm{Id} + v_n) \circ T_n \\ p_{n+1} &\leftarrow \mathcal{G}(p_n, \mathrm{Id} + v_n). \end{aligned}$$

- Inspired by Kolmogorov-Arnold-Moser iteration in dynamical systems theory, and Nash-Moser iteration in PDEs.

Score Operator Newton (SCONE) iteration for transport maps

Newton-type step:

$$-(q - p_n) = \mathcal{L}(q) v_n = (\nabla q) v_n + q(\nabla v_n) + \mathrm{tr}(\nabla^2 v_n).$$

Newton-type update:

$$\begin{aligned} T_{n+1} &\leftarrow (\mathrm{Id} + v_n) \circ T_n \\ p_{n+1} &\leftarrow \mathcal{G}(p_n, \mathrm{Id} + v_n). \end{aligned}$$

- Inspired by Kolmogorov-Arnold-Moser iteration in dynamical systems theory, and Nash-Moser iteration in PDEs.
- Conceptually different from empirical, parametric score-matching [**Koehler et al 2022**]

The SCONE construction of transport

- Optimize distance functional on probability measure space

The SCONE construction of transport

- Optimize distance functional on probability measure space
- Find an optimal map in an ansatz space

The SCONE construction of transport

- Optimize distance functional on probability measure space
- Find an optimal map in an ansatz space
- Triangular transport [**Moselhy and Marzouk 2012**], normalizing flows [**Papamakarios et al 2021**], neural ODEs [**Grathwohl et al 2018**]

The SCONE construction of transport

- Optimize distance functional on probability measure space
- Find an optimal map in an ansatz space
- Triangular transport [**Moselhy and Marzouk 2012**], normalizing flows [**Papamakarios et al 2021**], neural ODEs [**Grathwohl et al 2018**]

- Gradient flow of appropriate distance functional

The SCONE construction of transport

- Optimize distance functional on probability measure space
- Find an optimal map in an ansatz space
- Triangular transport [**Moselhy and Marzouk 2012**], normalizing flows [**Papamakarios et al 2021**], neural ODEs [**Grathwohl et al 2018**]

- Gradient flow of appropriate distance functional
- Transport map implicitly obtained via particle paths

The SCONE construction of transport

- Optimize distance functional on probability measure space
- Find an optimal map in an ansatz space
- Triangular transport [**Moselhy and Marzouk 2012**], normalizing flows [**Papamakarios et al 2021**], neural ODEs [**Grathwohl et al 2018**]

- Gradient flow of appropriate distance functional
- Transport map implicitly obtained via particle paths
- [**Jordan et al 1998**], [**Wibisono 2018**],

The SCONE construction of transport

- Optimize distance functional on probability measure space
- Find an optimal map in an ansatz space
- Triangular transport [**Moselhy and Marzouk 2012**], normalizing flows [**Papamakarios et al 2021**], neural ODEs [**Grathwohl et al 2018**]

- Gradient flow of appropriate distance functional
- Transport map implicitly obtained via particle paths
- [**Jordan et al 1998**], [**Wibisono 2018**], Stein Variational Gradient descent [**Liu and Wang 2016**] and many variants [**Chewi et al 2020, Duncan et al 2019**]

The SCONE construction of transport

- Optimize distance functional on probability measure space
- Find an optimal map in an ansatz space
- Triangular transport [**Moselhy and Marzouk 2012**], normalizing flows [**Papamakarios et al 2021**], neural ODEs [**Grathwohl et al 2018**]

- Gradient flow of appropriate distance functional
- Transport map implicitly obtained via particle paths
- [**Jordan et al 1998**], [**Wibisono 2018**], Stein Variational Gradient descent [**Liu and Wang 2016**] and many variants [**Chewi et al 2020, Duncan et al 2019**]

SCONE gives an iterative construction as the limit of a sequence of compositions. Based on finding zero of a score-residual operator.

Existence of a $\mathcal{C}^r$ transport and convergence of SCONE iteration

Theorem [**SCONE**(informal)] For every $\epsilon > 0, s \in \mathbb{N}$, there exists a $\delta > 0$ such that $\|p - q\|_s \leqslant \epsilon$ implies $\exists\ T \in \mathcal{C}^{s+2,\cdot}(M)$ such that (i) $\mathcal{G}(p, T) = q$ and (ii) $\| T - \mathrm{Id}\|_{s+2} \leqslant \delta$. Moreover, $T = \lim_{n\to\infty} T_n$ and $q = \lim_{n\to\infty} p_n$ in $\mathcal{C}^{s+2,\cdot}(\bar{\Omega})$, where $(T_n)_{n\geqslant 0}$ and $(p_n)_{n\geqslant 0}$ are the sequences generated by the Score Operator Newton iteration.

Existence of a $\mathcal{C}^r$ transport and convergence of SCONE iteration

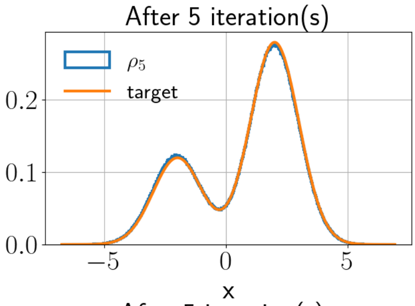
Theorem [**SCONE**(informal)] For every $\epsilon > 0$, $s \in \mathbb{N}$, there exists a $\delta > 0$ such that $\|p - q\|_s \leqslant \epsilon$ implies $\exists\ T \in \mathcal{C}^{s+2,\cdot}(M)$ such that (i) $\mathcal{G}(p, T) = q$ and (ii) $\|T - \mathrm{Id}\|_{s+2} \leqslant \delta$. Moreover, $T = \lim_{n\to\infty} T_n$ and $q = \lim_{n\to\infty} p_n$ in $\mathcal{C}^{s+2,\cdot}(\bar{\Omega})$, where $(T_n)_{n\geqslant 0}$ and $(p_n)_{n\geqslant 0}$ are the sequences generated by the Score Operator Newton iteration.

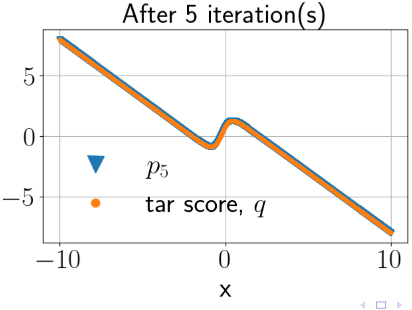
- Contraction mapping principle (Banach fixed point theorem) applied to

$$\mathcal{J}(v) = \mathcal{L}^{-1}(\mathcal{G}(q + \mathcal{L}v, \mathrm{Id} + v) - q),$$

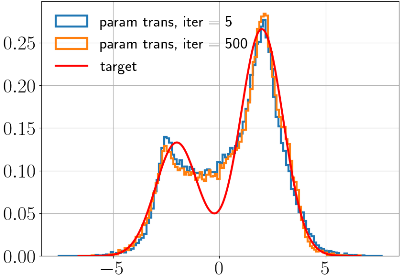
- Use elliptic regularity for proving continuity of derivative

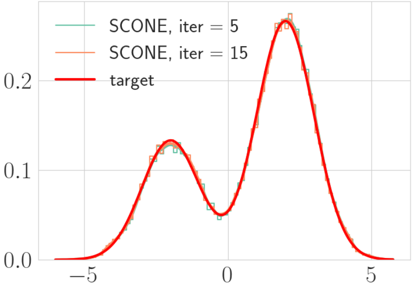
Numerical validation





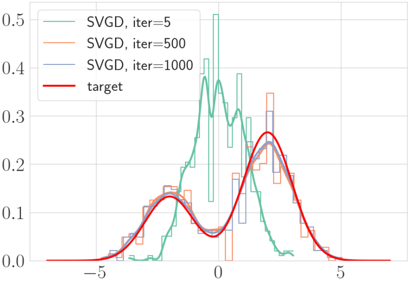
Comparison against parametric transport at the same computational cost

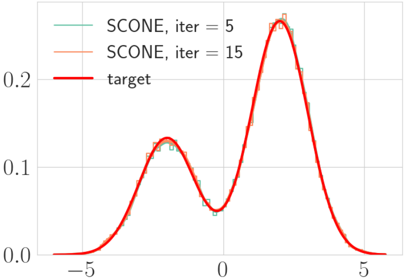




Left: Monotone transport [**Parno et al 2022**], up to 10th order Hermite polynomials, Number of parameters x Number of samples to approximate KL divergence = 11x(512*512/11). Right: SCONE with 512 grid points

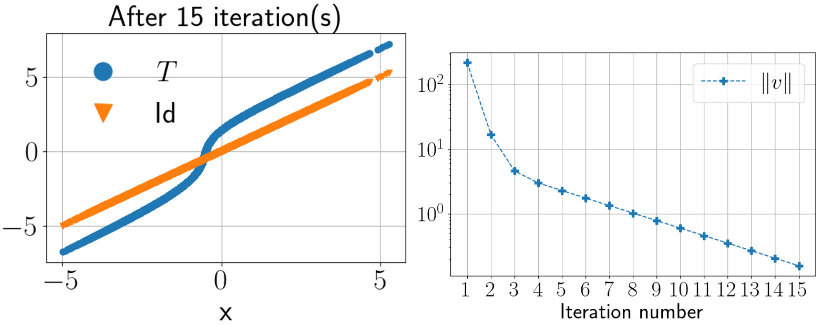
1D comparisons





Left: SVGD [**Liu and Wang 2016**] with 512 particles, RBF kernel, gradient descent with step size 0.01

Convergence of SCONE construction to the increasing rearrangement in 1D

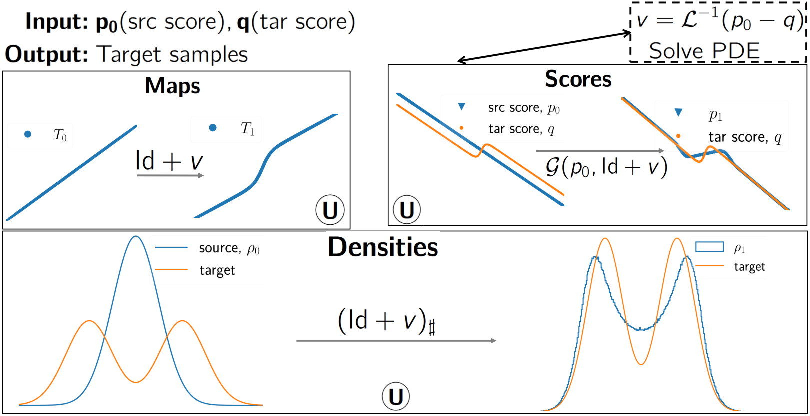


- Global dependence of v helps avoid mode collapse
- Tail behavior captured due to score matching

Score Operator Newton Transport

Input: $\mathbf{p_0}$(src score), $\mathbf{q}$(tar score)

Output: Target samples



Score Operator Newton construction: can be used for sampling, generative modeling, Bayesian inference and filtering in chaotic systems

- A deterministic nonparametric transport method derived with an operator root-finding principle.

Score Operator Newton construction: can be used for sampling, generative modeling, Bayesian inference and filtering in chaotic systems

- A deterministic nonparametric transport method derived with an operator root-finding principle.
- Convergence in Hölder norms using elliptic regularity

Score Operator Newton construction: can be used for sampling, generative modeling, Bayesian inference and filtering in chaotic systems

- A deterministic nonparametric transport method derived with an operator root-finding principle.
- Convergence in Hölder norms using elliptic regularity
- Newton-like features: unstable, converges fast

Score Operator Newton construction: can be used for sampling, generative modeling, Bayesian inference and filtering in chaotic systems

- A deterministic nonparametric transport method derived with an operator root-finding principle.
- Convergence in Hölder norms using elliptic regularity
- Newton-like features: unstable, converges fast
- Global nature of elliptic PDE helps i) avoid mode collapse and ii) capture tails

Score Operator Newton construction: can be used for sampling, generative modeling, Bayesian inference and filtering in chaotic systems

- A deterministic nonparametric transport method derived with an operator root-finding principle.
- Convergence in Hölder norms using elliptic regularity
- Newton-like features: unstable, converges fast
- Global nature of elliptic PDE helps i) avoid mode collapse and ii) capture tails
- Next steps: nonparametric PDE solves e.g. particle vortex methods, smooth particle hydrodynamics, PINNs etc.

Score Operator Newton construction: can be used for sampling, generative modeling, Bayesian inference and filtering in chaotic systems

- A deterministic nonparametric transport method derived with an operator root-finding principle.
- Convergence in Hölder norms using elliptic regularity
- Newton-like features: unstable, converges fast
- Global nature of elliptic PDE helps i) avoid mode collapse and ii) capture tails
- Next steps: nonparametric PDE solves e.g. particle vortex methods, smooth particle hydrodynamics, PINNs etc.
- Low-rank approximations of elliptic PDE solution?

C, Schäfer, Marzouk 2023 https://arxiv.org/abs/2305.09792

Learning chaotic dynamics from data

- Neural ODE [Chen et al 2018]:

$$\frac{d}{dt}\varphi_h^t(x) = h(\varphi_h^t(x)), \quad x \in \mathbb{R}^d. \tag{2}$$

Learning chaotic dynamics from data

- Neural ODE [Chen et al 2018]:

$$\frac{d}{dt}\varphi_h^t(x) = h(\varphi_h^t(x)), \quad x \in \mathbb{R}^d. \tag{2}$$

- ERM problem to minimize loss of the form $\ell(x_0, h) = \|\varphi_h^{\delta t}(x_0) - x_{\delta t}\|^2$

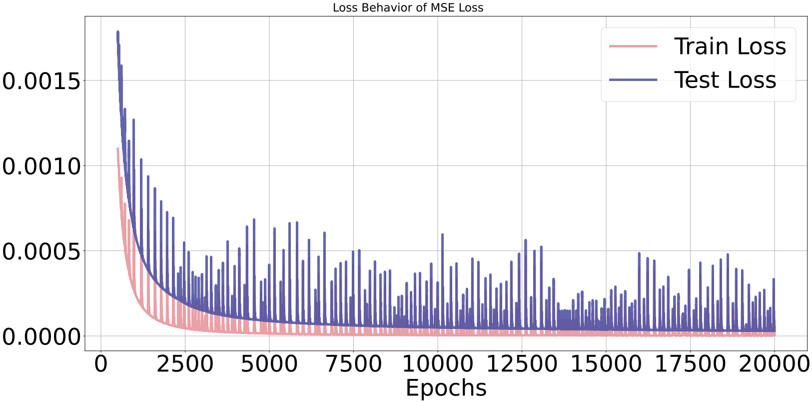
Learning chaotic dynamics from data

- Neural ODE [Chen et al 2018]:

$$\frac{d}{dt}\varphi_h^t(x) = h(\varphi_h^t(x)), \quad x \in \mathbb{R}^d. \tag{2}$$

- ERM problem to minimize loss of the form $\ell(x_0, h) = \|\varphi_h^{\delta t}(x_0) - x_{\delta t}\|^2$
- Training and test errors (for one-step predictions) small, but does not generalize

Learning the Lorenz ’63 system



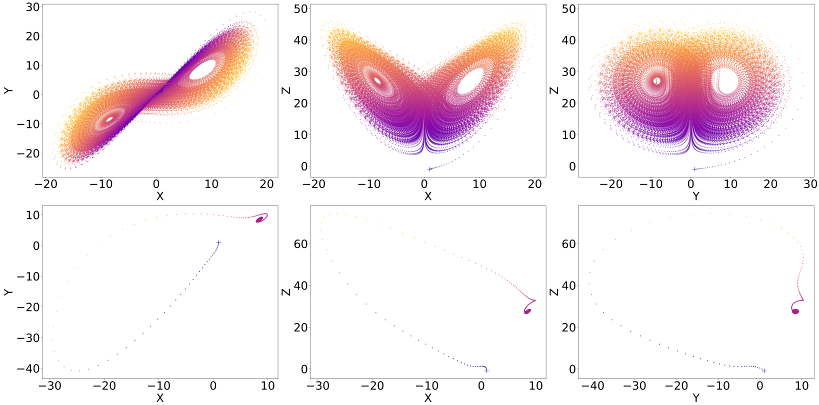
Good “generalization” performance. Three layer feed forward network trained with AdamW

Generalization => learning dynamics?

	Lyapunov Exponent
True LE	$\approx$ **[0.9, 0, -14.5]**
Neural ODE	$[0.8926, -0.0336, -6.0616]$

Generalization => learning dynamics?

	Lyapunov Exponent
True LE	$\approx$ **[0.9, 0, -14.5]**
Neural ODE	$[0.8926, -0.0336, -6.0616]$



Statistically accurate Neural ODE models

	Lyapunov Exponent
True LE	$\approx$ **[0.9, 0, -14.5]**
Neural ODE	$[0.8926, -0.0336, -6.0616]$
+Jacobian info	$[0.9022, -0.0024, -14.4803]$

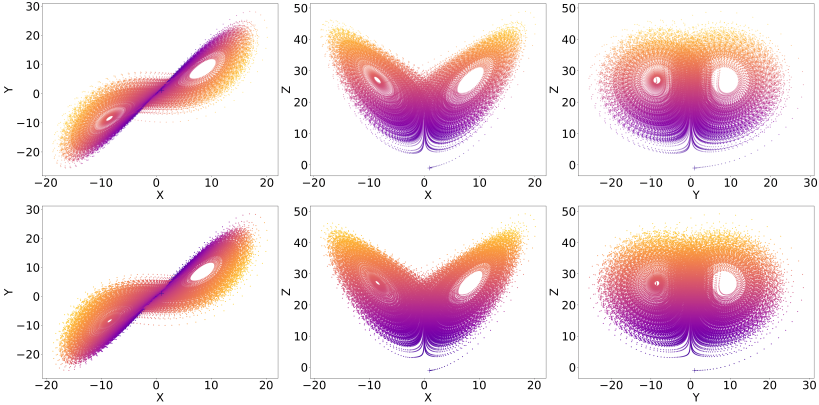
- Modified loss:
  $\ell(x_0, h) = \|\varphi_h^{\delta t}(x) - x_{\delta t}\|^2 + \lambda \|d\varphi_h^{\delta t}(x) - dx_{\delta t}(x)\|^2$

Statistically accurate Neural ODE models

	Lyapunov Exponent
True LE	$\approx$ **[0.9, 0, -14.5]**
Neural ODE	$[0.8926, -0.0336, -6.0616]$
+Jacobian info	$[0.9022, -0.0024, -14.4803]$

- Modified loss:
  $\ell(x_0, h) = \|\varphi_h^{\delta t}(x) - x_{\delta t}\|^2 + \lambda \|d\varphi_h^{\delta t}(x) - dx_{\delta t}(x)\|^2$
- With modified loss, statistical moments (correlations, LEs) are accurate

Learning out-of-attractor dynamics



Learning ergodic dynamics from data using Neural ODEs

- Can Neural ODEs learn statistics from timeseries data alone?

Learning ergodic dynamics from data using Neural ODEs

- Can Neural ODEs learn statistics from timeseries data alone?
- How should the loss modification depend on the dynamics?

Learning ergodic dynamics from data using Neural ODEs

- Can Neural ODEs learn statistics from timeseries data alone?
- How should the loss modification depend on the dynamics?
- How do we predict bifurcations? [Liu-Schiaffini 2023]

Learning ergodic dynamics from data using Neural ODEs

- Can Neural ODEs learn statistics from timeseries data alone?
- How should the loss modification depend on the dynamics?
- How do we predict bifurcations? [Liu-Schiaffini 2023]
- How should generalization error be defined?

Learning ergodic dynamics from data using Neural ODEs

- Can Neural ODEs learn statistics from timeseries data alone?
- How should the loss modification depend on the dynamics?
- How do we predict bifurcations? [Liu-Schiaffini 2023]
- How should generalization error be defined?
- Many more problems at the intersection of dynamics and learning theory!

Learning: Statistical stability implies generalization

C, Loukas, Gatmiry and Jegelka, NeurIPS 2022

Sampling: Score Operator Newton transport – root-finding principle for sampling

C, Schäfer and Marzouk, arxiv:2305.09792, 2023

Forecasting: context-dependent generalization analyses

Park and C, 2023