Random dynamical systems in deep neural networks

Maximilian Engel
KdVI (UvA) and FU Berlin

FoMI: A Seminar on the Foundations of Machine Intelligence
April 29, 2026

(Random) Dynamical Systems and Neural Networks

Artificial Neural Network as a function

$$\Phi : \mathbb{R}^D \times \mathbb{R}^d \to \mathbb{R}^q, \quad (x, y) \mapsto \Phi(x, y).$$

(Random) Dynamical Systems and Neural Networks

Artificial Neural Network as a function

$$\Phi : \mathbb{R}^D \times \mathbb{R}^d \to \mathbb{R}^q, \quad (x, y) \mapsto \Phi(x, y).$$

- **Training data**: $(y_1, z_1), \ldots, (y_N, z_N) \in \mathbb{R}^d \times \mathbb{R}^q$

(Random) Dynamical Systems and Neural Networks

Artificial Neural Network as a function

$$\Phi : \mathbb{R}^D \times \mathbb{R}^d \to \mathbb{R}^q, \quad (x, y) \mapsto \Phi(x, y).$$

- **Training data**: $(y_1, z_1), \ldots, (y_N, z_N) \in \mathbb{R}^d \times \mathbb{R}^q$
- **Training risk** (loss): $\mathcal{L}(x) := N^{-1} \sum_{i=1}^{N} |\Phi(x, y_i) - z_i|^2$

(Random) Dynamical Systems and Neural Networks

Artificial Neural Network as a function

$$\Phi : \mathbb{R}^D \times \mathbb{R}^d \to \mathbb{R}^q, \quad (x, y) \mapsto \Phi(x, y).$$

- **Training data**: $(y_1, z_1), \dots, (y_N, z_N) \in \mathbb{R}^d \times \mathbb{R}^q$
- **Training risk** (loss): $\mathcal{L}(x) := N^{-1} \sum_{i=1}^{N} |\Phi(x, y_i) - z_i|^2$

Two dynamical problem classes [Chemnitz/E./Kuehn/Kuntz 2025+]:

(Random) Dynamical Systems and Neural Networks

Artificial Neural Network as a function

$$\Phi : \mathbb{R}^D \times \mathbb{R}^d \to \mathbb{R}^q, \quad (x, y) \mapsto \Phi(x, y).$$

- **Training data**: $(y_1, z_1), \dots, (y_N, z_N) \in \mathbb{R}^d \times \mathbb{R}^q$
- **Training risk** (loss): $\mathcal{L}(x) := N^{-1} \sum_{i=1}^{N} |\Phi(x, y_i) - z_i|^2$

Two dynamical problem classes [Chemnitz/E./Kuehn/Kuntz 2025+]:

- A) **Dynamics** of **the Network** (Learning Process)

$$\varphi : \mathbb{R}^D \to \mathbb{R}^D, \quad x \mapsto \varphi(x) = x - \eta \nabla \mathcal{L}(x) \qquad \text{(GD)}$$

$$\hat{\varphi} : \Omega \times \mathbb{R}^D \to \mathbb{R}^D, \quad x \mapsto \hat{\varphi}_\omega(x) = x - \eta \nabla \hat{\mathcal{L}}_\omega(x) \qquad \text{(SGD)}$$

(Random) Dynamical Systems and Neural Networks

Artificial Neural Network as a function

$$\Phi : \mathbb{R}^D \times \mathbb{R}^d \to \mathbb{R}^q, \quad (x, y) \mapsto \Phi(x, y).$$

- **Training data**: $(y_1, z_1), \dots, (y_N, z_N) \in \mathbb{R}^d \times \mathbb{R}^q$
- **Training risk** (loss): $\mathcal{L}(x) := N^{-1} \sum_{i=1}^{N} |\Phi(x, y_i) - z_i|^2$

Two dynamical problem classes [Chemnitz/E./Kuehn/Kuntz 2025+]:

- A) **Dynamics of the Network** (Learning Process)

$$\varphi : \mathbb{R}^D \to \mathbb{R}^D, \quad x \mapsto \varphi(x) = x - \eta \nabla \mathcal{L}(x) \qquad \text{(GD)}$$
$$\hat{\varphi} : \Omega \times \mathbb{R}^D \to \mathbb{R}^D, \quad x \mapsto \hat{\varphi}_\omega(x) = x - \eta \nabla \hat{\mathcal{L}}_\omega(x) \qquad \text{(SGD)}$$

- B) **Dynamics on the Network** (Information Propagation)

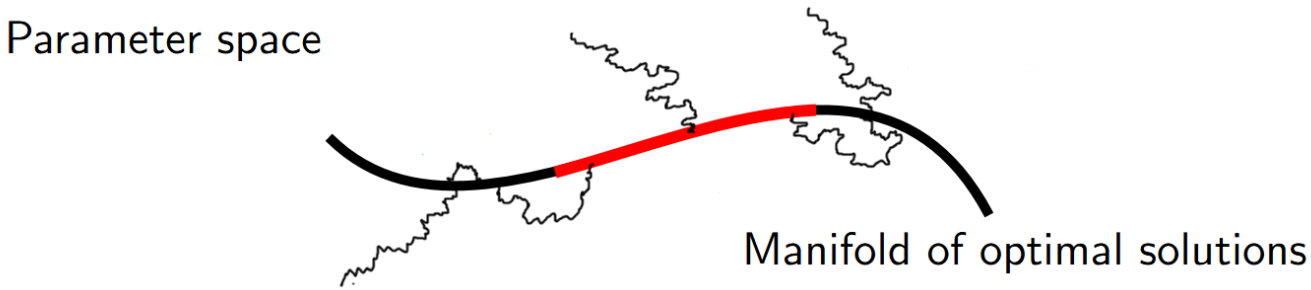
$$\Phi(x, \cdot) : \mathbb{R}^d \to \mathbb{R}^q \quad \text{or} \quad \hat{\Phi}_\omega(x, \cdot) : \Omega \times \mathbb{R}^d \to \mathbb{R}^q. \quad \text{(Neural map)}$$

Key questions for this talk

Key questions for this talk

A) **Dynamics of the Network**

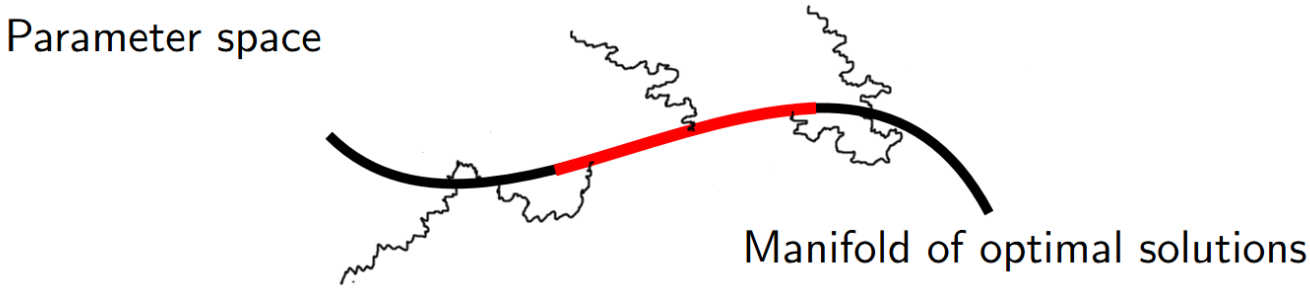
- Characterization of parameter solutions learned via (S)GD?
- Related to **generalization** from training to new data sets?



Key questions for this talk

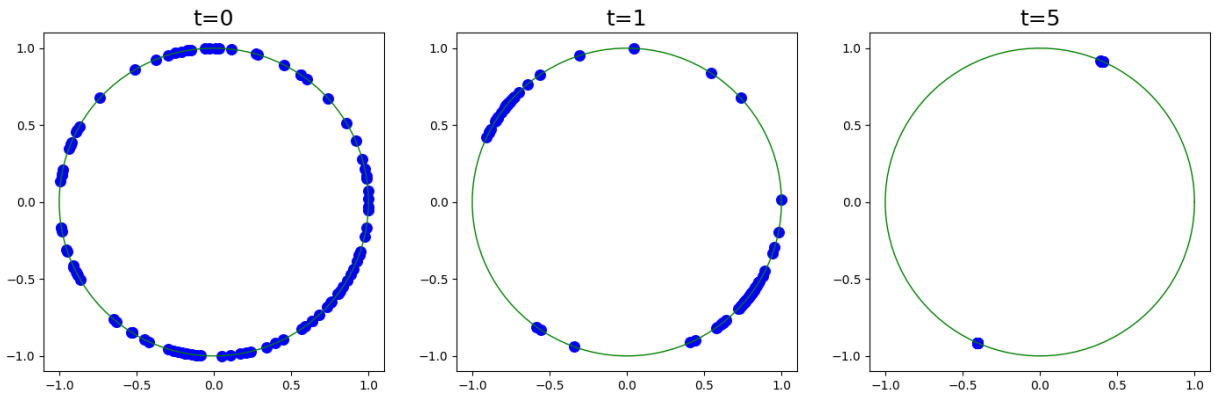
A) **Dynamics** of **the Network**

- Characterization of parameter solutions learned via (S)GD?
- Related to **generalization** from training to new data sets?



B) **Dynamics** on **the Network**

- Mechanisms for clustering in **transformers** of LLMs?



Random Dynamical Systems (RDS)

Random dynamical system (RDS) (θ, φ) [Arnold 1998] consists of

Random Dynamical Systems (RDS)

Random dynamical system (RDS) (θ, φ) [Arnold 1998] consists of

- dynamical system $(\theta_t)_{t\in\mathbb{T}}$ on probability space $(\Omega, \mathcal{F}, \mathbb{P})$

Random Dynamical Systems (RDS)

Random dynamical system (RDS) (θ, φ) [Arnold 1998] consists of

- dynamical system $(\theta_t)_{t\in\mathbb{T}}$ on probability space $(\Omega, \mathcal{F}, \mathbb{P})$
- and *cocycle* $(\varphi_\omega^t : X \to X)_{t\in\mathbb{T}}$ such that for all $\omega \in \Omega$ and $x \in X$

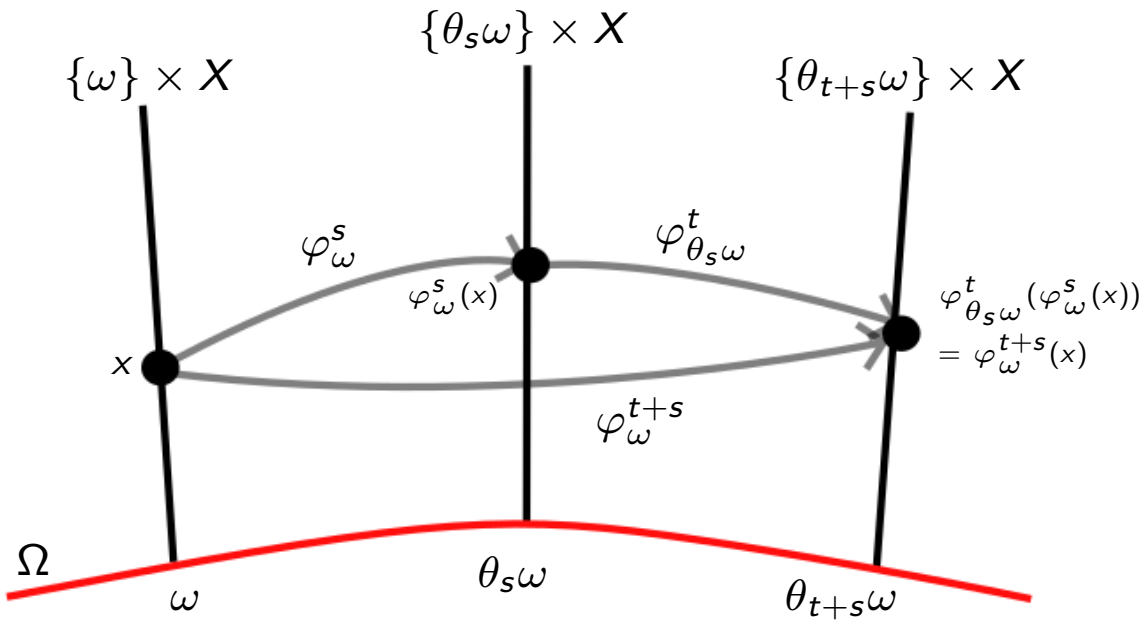
$$\varphi_\omega^{t+s}(x) = \varphi_{\theta_s\omega}^t(\varphi_\omega^s(x))\,.$$

Random Dynamical Systems (RDS)

Random dynamical system (RDS) (θ, φ) [Arnold 1998] consists of

- dynamical system $(\theta_t)_{t\in\mathbb{T}}$ on probability space $(\Omega, \mathcal{F}, \mathbb{P})$
- and *cocycle* $(\varphi_\omega^t : X \to X)_{t\in\mathbb{T}}$ such that for all $\omega \in \Omega$ and $x \in X$

$$\varphi_\omega^{t+s}(x) = \varphi_{\theta_s\omega}^t(\varphi_\omega^s(x)) \,.$$



RDS in neural networks

A) **Stochastic Gradient Descent (SGD)**:

RDS in neural networks

A) **Stochastic Gradient Descent (SGD)**:

- Write $\mathcal{L}(x) := N^{-1} \sum_{i=1}^{N} \mathcal{L}_i(x)$, where $\mathcal{L}_i(x) := |\Phi(x, y_i) - z_i|^2$

RDS in neural networks

A) **Stochastic Gradient Descent (SGD)**:

- Write $\mathcal{L}(x) := N^{-1} \sum_{i=1}^{N} \mathcal{L}_i(x)$, where $\mathcal{L}_i(x) := |\Phi(x, y_i) - z_i|^2$
- Then, for $\omega = (\omega_1, \omega_2, \dots) \in \Omega$ uniform i.i.d. r.v., $(\theta\omega)_i = \omega_{i+1}$,

$$\varphi_\omega^{n+1}(x) = \varphi_\omega^n(x) - \eta \nabla \mathcal{L}_{\omega_{n+1}}(\varphi_\omega^n(x)).$$

RDS in neural networks

A) **Stochastic Gradient Descent (SGD)**:

- Write $\mathcal{L}(x) := N^{-1} \sum_{i=1}^{N} \mathcal{L}_i(x)$, where $\mathcal{L}_i(x) := |\Phi(x, y_i) - z_i|^2$
- Then, for $\omega = (\omega_1, \omega_2, \dots) \in \Omega$ uniform i.i.d. r.v., $(\theta\omega)_i = \omega_{i+1}$,

$$\varphi_\omega^{n+1}(x) = \varphi_\omega^n(x) - \eta \nabla \mathcal{L}_{\omega_{n+1}}(\varphi_\omega^n(x)).$$

B) **Transformers as Stochastic Differential Equations (SDEs)**

$$\mathrm{d}X_t = F_0(X_t)\,\mathrm{d}t + \sum_{j=1}^{m} F_j(X_t)\,\partial W_t^j, \quad X_0 = x \in \mathbb{S}^{n-1}. \qquad \text{(SDE)}$$

RDS in neural networks

A) **Stochastic Gradient Descent (SGD)**:

- Write $\mathcal{L}(x) := N^{-1} \sum_{i=1}^{N} \mathcal{L}_i(x)$, where $\mathcal{L}_i(x) := |\Phi(x, y_i) - z_i|^2$
- Then, for $\omega = (\omega_1, \omega_2, \dots) \in \Omega$ uniform i.i.d. r.v., $(\theta\omega)_i = \omega_{i+1}$,

$$\varphi_\omega^{n+1}(x) = \varphi_\omega^n(x) - \eta \nabla \mathcal{L}_{\omega_{n+1}}(\varphi_\omega^n(x)).$$

B) **Transformers as Stochastic Differential Equations (SDEs)**

$$\mathrm{d}X_t = F_0(X_t)\,\mathrm{d}t + \sum_{j=1}^{m} F_j(X_t)\,\partial W_t^j, \quad X_0 = x \in \mathbb{S}^{n-1}. \qquad \text{(SDE)}$$

- $\Omega = C_0(\mathbb{R}, \mathbb{R}^m)$ with Wiener measure $\mathbb{P}$, $\theta_t\omega(\cdot) := \omega(t + \cdot) - \omega(t)$

RDS in neural networks

A) **Stochastic Gradient Descent (SGD)**:

- Write $\mathcal{L}(x) := N^{-1} \sum_{i=1}^{N} \mathcal{L}_i(x)$, where $\mathcal{L}_i(x) := |\Phi(x, y_i) - z_i|^2$
- Then, for $\omega = (\omega_1, \omega_2, \dots) \in \Omega$ uniform i.i.d. r.v., $(\theta\omega)_i = \omega_{i+1}$,

$$\varphi_\omega^{n+1}(x) = \varphi_\omega^n(x) - \eta \nabla \mathcal{L}_{\omega_{n+1}}(\varphi_\omega^n(x)).$$

B) **Transformers as Stochastic Differential Equations (SDEs)**

$$\mathrm{d}X_t = F_0(X_t)\,\mathrm{d}t + \sum_{j=1}^{m} F_j(X_t)\,\partial W_t^j, \quad X_0 = x \in \mathbb{S}^{n-1}. \qquad \text{(SDE)}$$

- $\Omega = C_0(\mathbb{R}, \mathbb{R}^m)$ with Wiener measure $\mathbb{P}$, $\theta_t\omega(\cdot) := \omega(t + \cdot) - \omega(t)$
- The solution of (SDE) can be written as cocycle $\varphi_\omega^t(x)$.

RDS in neural networks

A) **Stochastic Gradient Descent (SGD)**:

- Write $\mathcal{L}(x) := N^{-1} \sum_{i=1}^{N} \mathcal{L}_i(x)$, where $\mathcal{L}_i(x) := |\Phi(x, y_i) - z_i|^2$
- Then, for $\omega = (\omega_1, \omega_2, \dots) \in \Omega$ uniform i.i.d. r.v., $(\theta\omega)_i = \omega_{i+1}$,

$$\varphi_\omega^{n+1}(x) = \varphi_\omega^n(x) - \eta \nabla \mathcal{L}_{\omega_{n+1}}(\varphi_\omega^n(x)).$$

B) **Transformers as Stochastic Differential Equations (SDEs)**

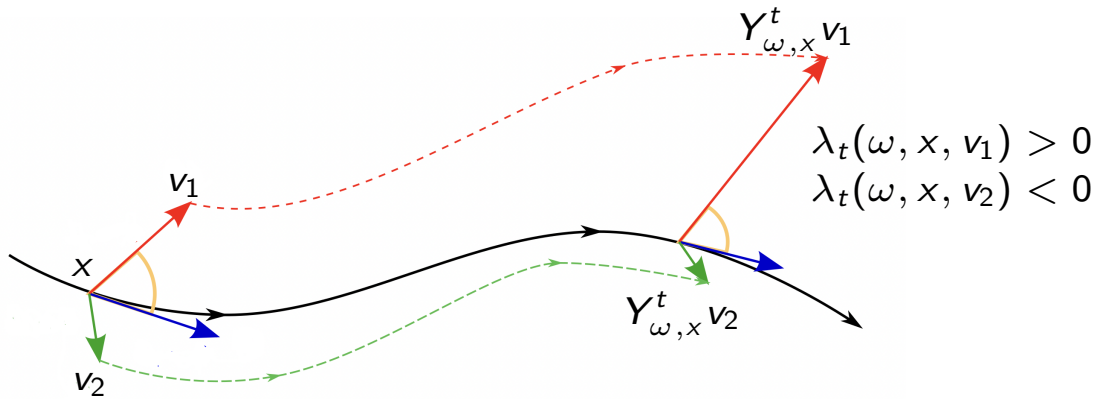
$$\mathrm{d}X_t = F_0(X_t)\,\mathrm{d}t + \sum_{j=1}^{m} F_j(X_t)\,\partial W_t^j, \quad X_0 = x \in \mathbb{S}^{n-1}. \qquad \text{(SDE)}$$

- $\Omega = C_0(\mathbb{R}, \mathbb{R}^m)$ with Wiener measure $\mathbb{P}$, $\theta_t\omega(\cdot) := \omega(t + \cdot) - \omega(t)$
- The solution of (SDE) can be written as cocycle $\varphi_\omega^t(x)$.

Question for A) and B) : What happens with $\varphi_\omega^t(x)$ as $t \to \infty$?

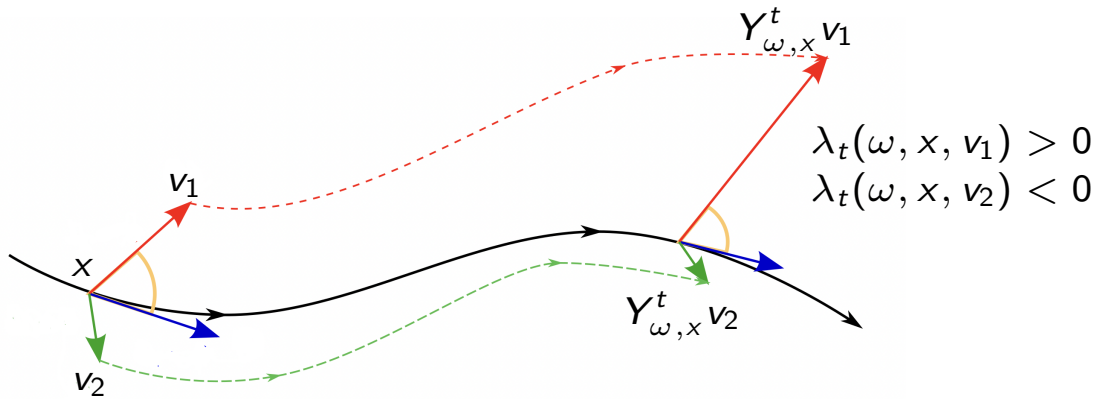
Lyapunov exponents

- We denote $Y^t_{\omega,x} := \mathrm{D}_x\varphi^t_\omega(x)$ and set $\lambda_t(\omega, x, v) = \frac{1}{t}\log\|Y^t_{\omega,x}v\|$.



Lyapunov exponents

- We denote $Y^t_{\omega,x} := \mathrm{D}_x\varphi^t_\omega(x)$ and set $\lambda_t(\omega, x, v) = \frac{1}{t}\log\|Y^t_{\omega,x}v\|$.



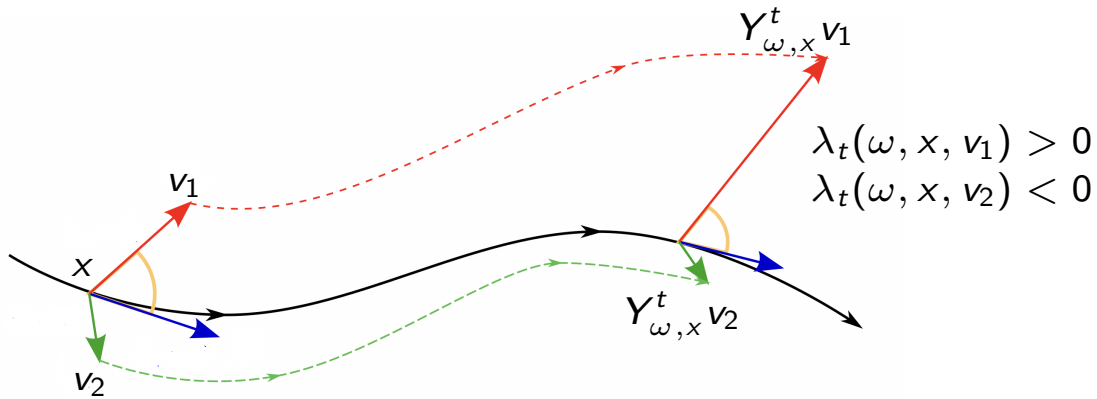
- For ergodic measure μ of (θ, φ)

$$\lambda = \lim_{t\to\infty} \frac{1}{t}\log\|Y^t_{\omega,x}\|, \quad \text{for } \mu\text{-almost all } (\omega, x),$$

is the maximal **Lyapunov exponent**.

Lyapunov exponents

- We denote $Y^t_{\omega,x} := \mathrm{D}_x\varphi^t_\omega(x)$ and set $\lambda_t(\omega, x, v) = \frac{1}{t}\log\|Y^t_{\omega,x}v\|$.



- For ergodic measure μ of (θ, φ)

$$\lambda = \lim_{t\to\infty} \frac{1}{t}\log\|Y^t_{\omega,x}\|, \quad \text{for } \mu\text{-almost all } (\omega, x),$$

is the maximal **Lyapunov exponent**.

Typical Paradigm:

- $\lambda < 0 \Rightarrow$ Synchronization/Accumulation
- $\lambda > 0 \Rightarrow$ Instability/Repulsion/Chaos

A) Characterizing dynamical stability of stochastic gradient descent in overparameterized learning

[Chemnitz/E., Journal of Machine Learning Research 2025]

Overdetermined vs. Overparameterized regime

D = number of parameters, N = size of the training data set

Overdetermined $D < N$:

Overparameterized $D > N$:

Overdetermined vs. Overparameterized regime

D = number of parameters, N = size of the training data set

Overdetermined $D < N$:

- "classical" regime (- 2010s)

Overparameterized $D > N$:

Overdetermined vs. Overparameterized regime

D = number of parameters, N = size of the training data set

Overdetermined $D < N$:

- "classical" regime (- 2010s)
- no interpolation solutions
  $\{x : \mathcal{L}(x) = 0\} = \emptyset$

Overparameterized $D > N$:

Overdetermined vs. Overparameterized regime

D = number of parameters, N = size of the training data set

Overdetermined $D < N$:

- "classical" regime (- 2010s)
- no interpolation solutions
  $\{x : \mathcal{L}(x) = 0\} = \emptyset$
- global minimum unique

Overparameterized $D > N$:

Overdetermined vs. Overparameterized regime

D = number of parameters, N = size of the training data set

Overdetermined $D < N$:

- "classical" regime (- 2010s)
- no interpolation solutions $\{x : \mathcal{L}(x) = 0\} = \emptyset$
- global minimum unique
- Generalization depends on network (mostly)

Overparameterized $D > N$:

Overdetermined vs. Overparameterized regime

D = number of parameters, N = size of the training data set

Overdetermined $D < N$:

- "classical" regime (- 2010s)
- no interpolation solutions $\{x : \mathcal{L}(x) = 0\} = \emptyset$
- global minimum unique
- Generalization depends on network (mostly)

Overparameterized $D > N$:

- "modern" regime (2010s -)

Overdetermined vs. Overparameterized regime

D = number of parameters, N = size of the training data set

Overdetermined $D < N$:

- "classical" regime (- 2010s)
- no interpolation solutions $\{x : \mathcal{L}(x) = 0\} = \emptyset$
- global minimum unique
- Generalization depends on network (mostly)

Overparameterized $D > N$:

- "modern" regime (2010s -)
- $(D - N)$-dim. manifold of interp. sol. $\{x : \mathcal{L}(x) = 0\}$

Overdetermined vs. Overparameterized regime

D = number of parameters, N = size of the training data set

Overdetermined $D < N$:

- "classical" regime (- 2010s)
- no interpolation solutions $\{x : \mathcal{L}(x) = 0\} = \emptyset$
- global minimum unique
- Generalization depends on network (mostly)

Overparameterized $D > N$:

- "modern" regime (2010s -)
- $(D - N)$-dim. manifold of interp. sol. $\{x : \mathcal{L}(x) = 0\}$
- global min. not unique

Overdetermined vs. Overparameterized regime

D = number of parameters, N = size of the training data set

Overdetermined $D < N$:

- "classical" regime (- 2010s)
- no interpolation solutions $\{x : \mathcal{L}(x) = 0\} = \emptyset$
- global minimum unique
- Generalization depends on network (mostly)

Overparameterized $D > N$:

- "modern" regime (2010s -)
- $(D - N)$-dim. manifold of interp. sol. $\{x : \mathcal{L}(x) = 0\}$
- global min. not unique
- Generalization depends on network and **learning**

Overdetermined vs. Overparameterized regime

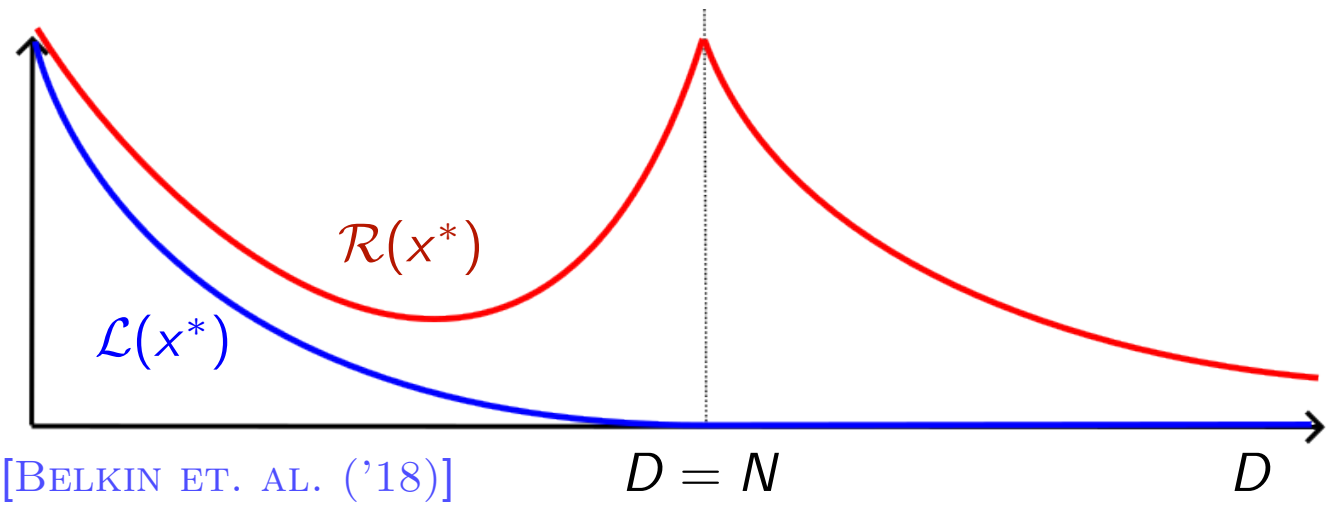
D = number of parameters, N = size of the training data set

Overdetermined $D < N$:

- "classical" regime (- 2010s)
- no interpolation solutions $\{x : \mathcal{L}(x) = 0\} = \emptyset$
- global minimum unique
- Generalization depends on network (mostly)

Overparameterized $D > N$:

- "modern" regime (2010s -)
- $(D - N)$-dim. manifold of interp. sol. $\{x : \mathcal{L}(x) = 0\}$
- global min. not unique
- Generalization depends on network and **learning**



[Belkin et. al. ('18)]

Learning in the overparameterized regime

Loss: $\mathcal{L}_i(x) := (\Phi(x, y_i) - z_i)^2$, $\mathcal{L}(x) := N^{-1} \sum_{i=1}^{N} \mathcal{L}_i(x)$

Simplest algorithm: *Gradient descent (GD)*

Learning in the overparameterized regime

Loss: $\mathcal{L}_i(x) := (\Phi(x, y_i) - z_i)^2$, $\mathcal{L}(x) := N^{-1} \sum_{i=1}^{N} \mathcal{L}_i(x)$

Simplest algorithm: *Gradient descent (GD)*

1. **Initialization:** $X_0^{\mathsf{GD}} \in \mathbb{R}^D$ random with dist ν_{init}

Learning in the overparameterized regime

Loss: $\mathcal{L}_i(x) := (\Phi(x, y_i) - z_i)^2$, $\mathcal{L}(x) := N^{-1} \sum_{i=1}^{N} \mathcal{L}_i(x)$

Simplest algorithm: *Gradient descent (GD)*

1. **Initialization:** $X_0^{\mathsf{GD}} \in \mathbb{R}^D$ random with dist ν_{init}
2. **Update:** $\eta > 0$ learning rate

$$X_{n+1}^{\mathsf{GD}} = X_n^{\mathsf{GD}} - \eta \nabla \mathcal{L}(X_n^{\mathsf{GD}})$$

Learning in the overparameterized regime

Loss: $\mathcal{L}_i(x) := (\Phi(x, y_i) - z_i)^2$, $\mathcal{L}(x) := N^{-1} \sum_{i=1}^{N} \mathcal{L}_i(x)$

Simplest algorithm: *Gradient descent (GD)*

1. **Initialization:** $X_0^{\mathrm{GD}} \in \mathbb{R}^D$ random with dist ν_{init}
2. **Update:** $\eta > 0$ learning rate

$$X_{n+1}^{\mathrm{GD}} = X_n^{\mathrm{GD}} - \eta \nabla \mathcal{L}(X_n^{\mathrm{GD}})$$

3. **Output:** $X_{\lim}^{\mathrm{GD}} := \begin{cases} \lim_{n\to\infty} X_n^{\mathrm{GD}}, & \text{if convergent} \\ \emptyset, & \text{else.} \end{cases}$

Learning in the overparameterized regime

Loss: $\mathcal{L}_i(x) := (\Phi(x, y_i) - z_i)^2$, $\mathcal{L}(x) := N^{-1} \sum_{i=1}^{N} \mathcal{L}_i(x)$

Simplest algorithm: *Gradient descent (GD)*
Most common: Variants of *stochastic gradient descent (SGD)*

1. **Initialization:** $X_0^{\mathrm{GD}} \in \mathbb{R}^D$ random with dist ν_{init}
2. **Update:** $\eta > 0$ learning rate

$$X_{n+1}^{\mathrm{GD}} = X_n^{\mathrm{GD}} - \eta \nabla \mathcal{L}(X_n^{\mathrm{GD}})$$

3. **Output:** $X_{\lim}^{\mathrm{GD}} := \begin{cases} \lim_{n\to\infty} X_n^{\mathrm{GD}}, & \text{if convergent} \\ \emptyset, & \text{else.} \end{cases}$

Learning in the overparameterized regime

Loss: $\mathcal{L}_i(x) := (\Phi(x, y_i) - z_i)^2$, $\mathcal{L}(x) := N^{-1} \sum_{i=1}^{N} \mathcal{L}_i(x)$

Simplest algorithm: *Gradient descent (GD)*
Most common: Variants of *stochastic gradient descent (SGD)*

1. **Initialization:** $X_0^{\mathrm{GD/SGD}} \in \mathbb{R}^D$ random with dist ν_{init}
2. **Update:** $\eta > 0$ learning rate, $\omega_n \in [N]$ uniform i.i.d.,

$$X_{n+1}^{\mathrm{GD}} = X_n^{\mathrm{GD}} - \eta \nabla \mathcal{L}(X_n^{\mathrm{GD}})$$
$$X_{n+1}^{\mathrm{SGD}} = X_n^{\mathrm{SGD}} - \eta \nabla \mathcal{L}_{\omega_{n+1}}(X_n^{\mathrm{SGD}})$$

3. **Output:** $X_{\mathrm{lim}}^{\mathrm{GD/SGD}} := \begin{cases} \lim_{n\to\infty} X_n^{\mathrm{GD/SGD}}, & \text{if convergent} \\ \emptyset, & \text{else.} \end{cases}$

Learning in the overparameterized regime

Loss: $\mathcal{L}_i(x) := (\Phi(x, y_i) - z_i)^2$, $\mathcal{L}(x) := N^{-1} \sum_{i=1}^{N} \mathcal{L}_i(x)$

Simplest algorithm: *Gradient descent (GD)*
Most common: Variants of *stochastic gradient descent (SGD)*

1. **Initialization:** $X_0^{\mathrm{GD/SGD}} \in \mathbb{R}^D$ random with dist ν_{init}
2. **Update:** $\eta > 0$ learning rate, $\omega_n \in [N]$ uniform i.i.d.,

$$X_{n+1}^{\mathrm{GD}} = X_n^{\mathrm{GD}} - \eta \nabla \mathcal{L}(X_n^{\mathrm{GD}})$$
$$X_{n+1}^{\mathrm{SGD}} = X_n^{\mathrm{SGD}} - \eta \nabla \mathcal{L}_{\omega_{n+1}}(X_n^{\mathrm{SGD}})$$

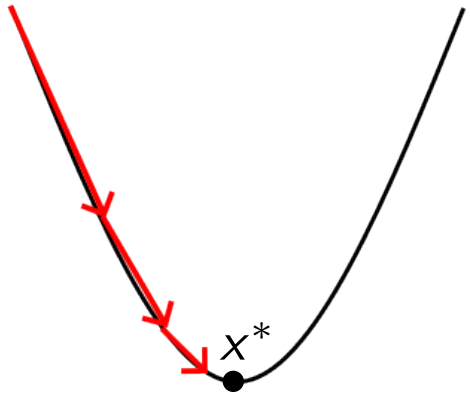
3. **Output:** $X_{\mathrm{lim}}^{\mathrm{GD/SGD}} := \begin{cases} \lim_{n\to\infty} X_n^{\mathrm{GD/SGD}}, & \text{if convergent} \\ \emptyset, & \text{else.} \end{cases}$

Question:

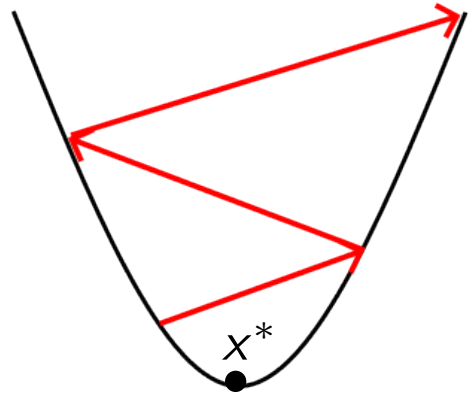
Do we converge and, if yes, whereto?

Dynamical stability in gradient descent

$$x \mapsto x - \eta \nabla \mathcal{L}(x)$$



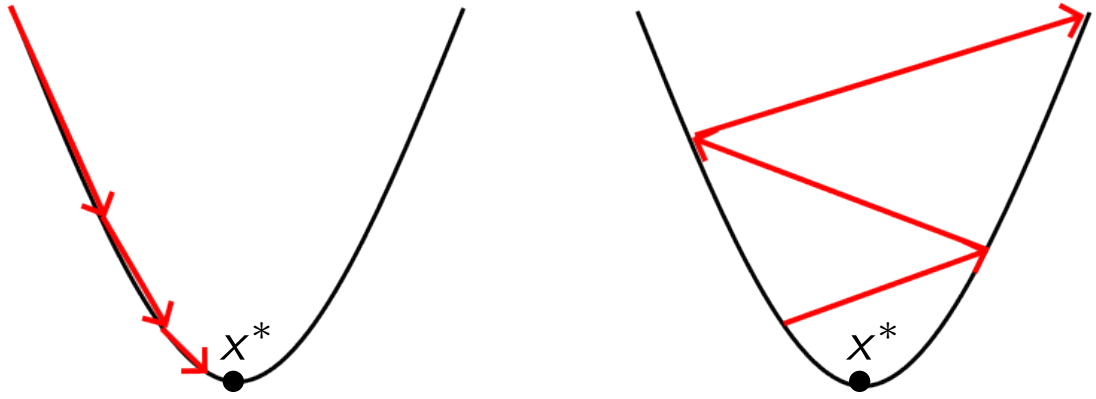
$$\eta < \frac{2}{\|\text{Hess } \mathcal{L}(x^*)\|}$$



$$\eta > \frac{2}{\|\text{Hess } \mathcal{L}(x^*)\|}$$

Dynamical stability in gradient descent

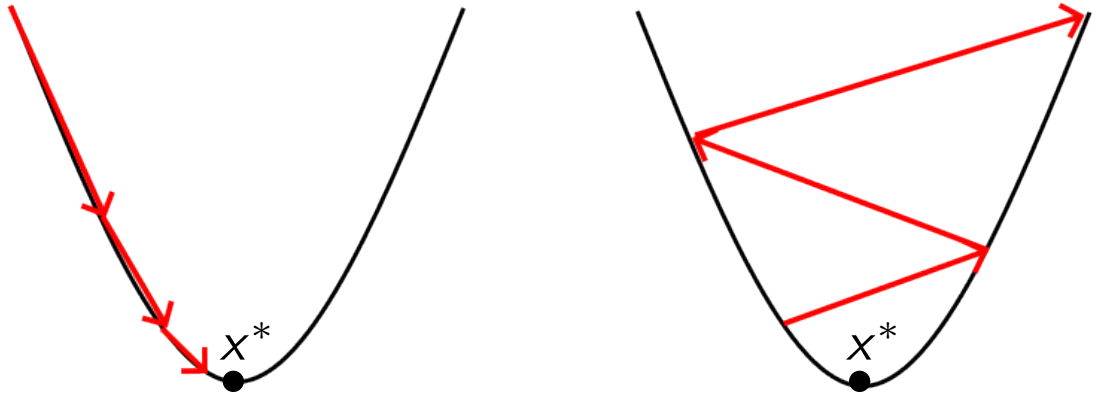
$$x \mapsto x - \eta \nabla \mathcal{L}(x)$$



Overdetermined regime: Choose $\eta \ll 1$.

Dynamical stability in gradient descent

$$x \mapsto x - \eta \nabla \mathcal{L}(x)$$

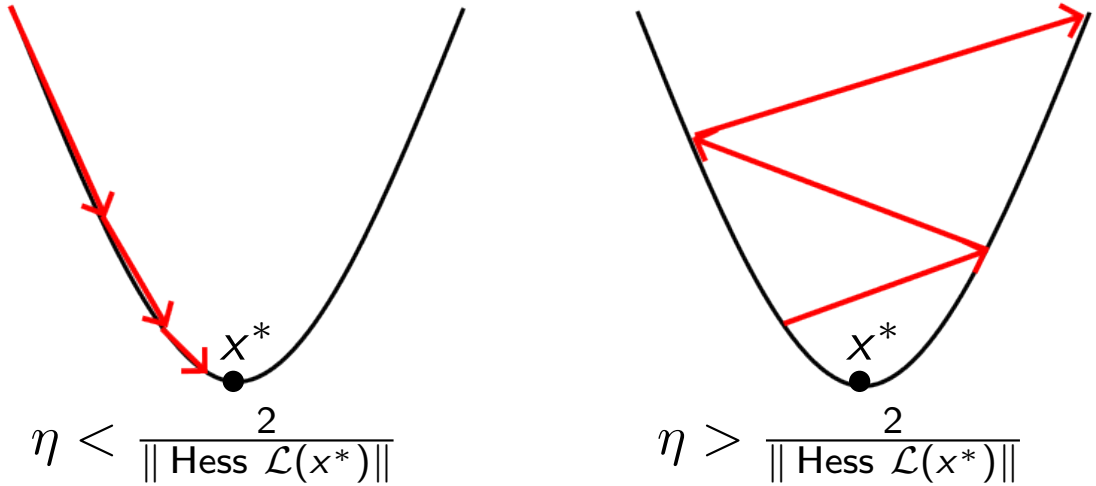


Overdetermined regime: Choose $\eta \ll 1$.

Overparameterized regime: $\| \text{Hess } \mathcal{L}(x^*) \| < 2\eta^{-1}$

Dynamical stability in gradient descent

$$x \mapsto x - \eta \nabla \mathcal{L}(x)$$



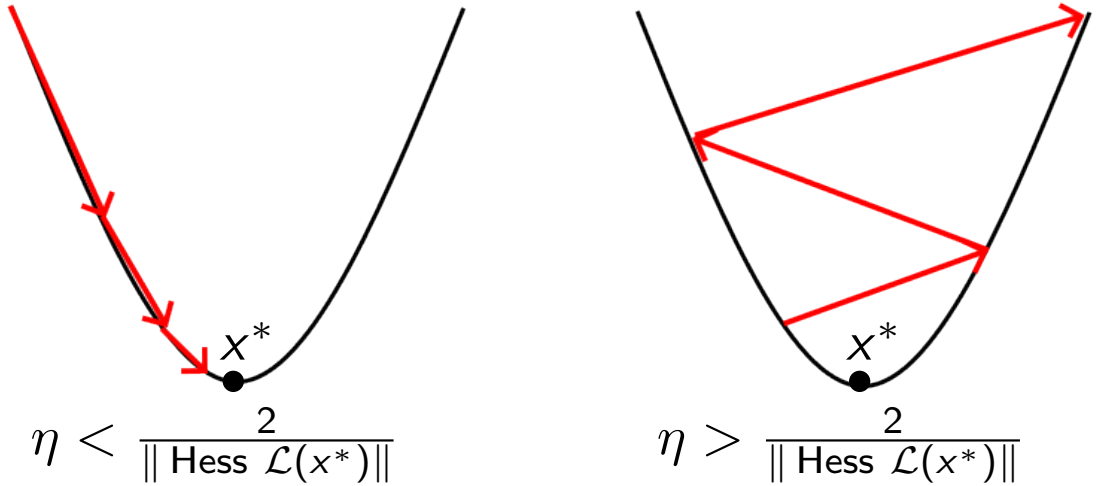
Overdetermined regime: Choose $\eta \ll 1$.

Overparameterized regime: $\| \text{Hess } \mathcal{L}(x^*) \| < 2\eta^{-1}$

- Instability restricts the **effective hypothesis class**

Dynamical stability in gradient descent

$$x \mapsto x - \eta \nabla \mathcal{L}(x)$$



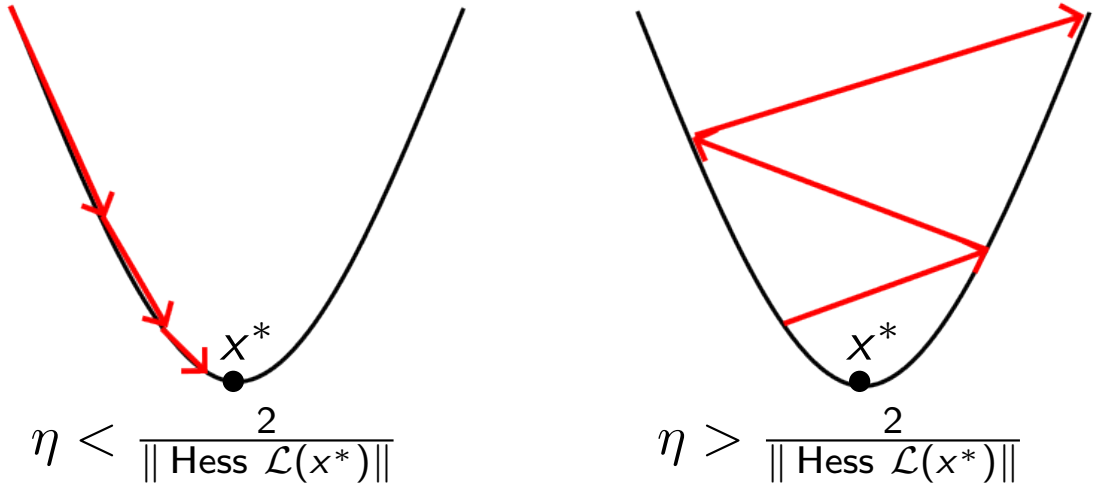
Overdetermined regime: Choose $\eta \ll 1$.

Overparameterized regime: $\|\text{Hess } \mathcal{L}(x^*)\| < 2\eta^{-1}$

- Instability restricts the **effective hypothesis class**
- $\|\text{Hess } \mathcal{L}(x^*)\|$ small $\rightsquigarrow$ Good generalization
  [Hochreiter, Schmidthuber ('97)]

Dynamical stability in gradient descent

$$x \mapsto x - \eta \nabla \mathcal{L}(x)$$

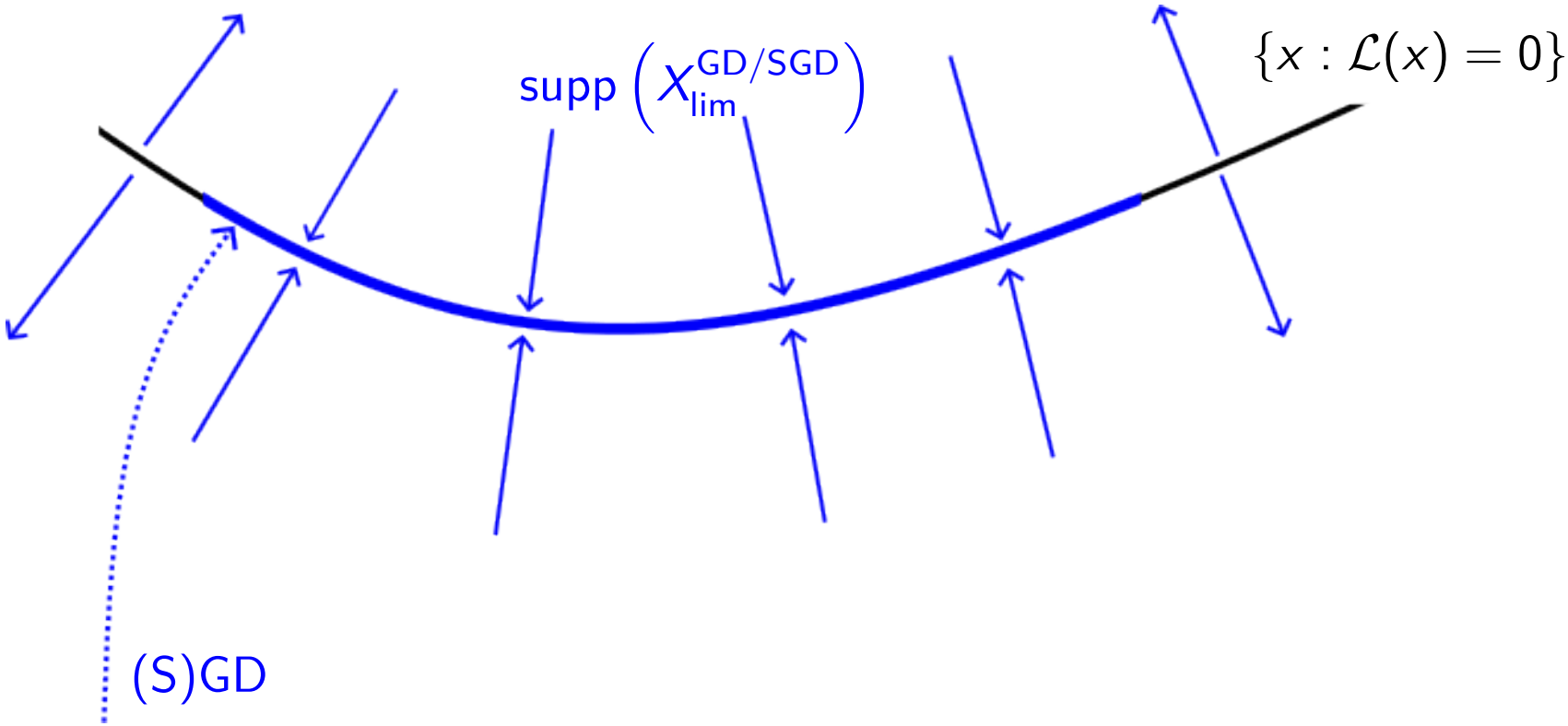


Overdetermined regime: Choose $\eta \ll 1$.

Overparameterized regime: $\|\text{Hess } \mathcal{L}(x^*)\| < 2\eta^{-1}$

- Instability restricts the **effective hypothesis class**
- $\|\text{Hess } \mathcal{L}(x^*)\|$ small $\rightsquigarrow$ Good generalization
  [Hochreiter, Schmidthuber ('97)]
- "Edge of stability": $\|\text{Hess } \mathcal{L}(x^*)\| \approx 2\eta^{-1}$
  [Wu et al. ('18)], [Cohen et al. ('20)]

Dynamical stability in gradient descent



Goal: Characterize

$$\text{supp}\left(X_{\text{lim}}^{\text{GD/SGD}}\right) := \text{supp}\left(\text{Law}\left(X_{\text{lim}}^{\text{GD/SGD}}\right)\right).$$

Main Results - GD

Global min.: $\mathcal{M} = \{x : \mathcal{L}(x) = 0\}$, Update: $\varphi(x) = x - \eta \nabla \mathcal{L}(x)$

Main Results - GD

Global min.: $\mathcal{M} = \{x : \mathcal{L}(x) = 0\}$, Update: $\varphi(x) = x - \eta \nabla \mathcal{L}(x)$

Assume

- $\mathcal{M}$ manifold [Cooper ('21)],
- ν_{init} equivalent to Lebesgue (usually Gaussian),
- φ non-singular [Crăciun, Ghoshdastidar ('24)].

Main Results - GD

Global min.: $\mathcal{M} = \{x : \mathcal{L}(x) = 0\}$, Update: $\varphi(x) = x - \eta \nabla \mathcal{L}(x)$

Assume

- $\mathcal{M}$ manifold [Cooper ('21)],
- ν_{init} equivalent to Lebesgue (usually Gaussian),
- φ non-singular [Crăciun, Ghoshdastidar ('24)].

For $x^* \in \mathcal{M}$, define

$$\mu(x^*) := \log \|\mathrm{D}\varphi(x^*)|_{(\mathrm{T}_{x^*}\mathcal{M})^\perp}\|,$$

such that $\mu(x^*) < 0 \Leftrightarrow \eta < 2\|\operatorname{Hess} \mathcal{L}(x^*)\|^{-1}$.

Main Results - GD

Global min.: $\mathcal{M} = \{x : \mathcal{L}(x) = 0\}$, Update: $\varphi(x) = x - \eta\nabla\mathcal{L}(x)$

Assume

- $\mathcal{M}$ manifold [Cooper ('21)],
- ν_{init} equivalent to Lebesgue (usually Gaussian),
- φ non-singular [Crăciun, Ghoshdastidar ('24)].

For $x^* \in \mathcal{M}$, define

$$\mu(x^*) := \log \|\mathrm{D}\varphi(x^*)|_{(\mathrm{T}_{x^*}\mathcal{M})^\perp}\|,$$

such that $\mu(x^*) < 0 \Leftrightarrow \eta < 2\|\operatorname{Hess}\mathcal{L}(x^*)\|^{-1}$.

Theorem (Chemnitz/E. 2025)

Let $x^* \in \mathcal{M}$. Then

$$\mu(x^*) < 0 \Rightarrow x^* \in \operatorname{supp}(X_{\text{lim}}^{\text{GD}}),$$
$$\mu(x^*) > 0 \Rightarrow x^* \notin \operatorname{supp}(X_{\text{lim}}^{\text{GD}}).$$

Linearization - SGD

Consider the **random dynamical system** given by

$$\hat{\varphi}_i : \mathbb{R}^D \to \mathbb{R}^D,\ x \mapsto x - \eta \nabla \mathcal{L}_i(x).$$

Let $\varphi_\omega^n = \hat{\varphi}_{\omega_n} \circ \cdots \circ \hat{\varphi}_{\omega_1}$, where $\omega = (\omega_1, \omega_2, \dots)$ uniform i.i.d. r.v..

Linearization - SGD

Consider the **random dynamical system** given by

$$\hat{\varphi}_i : \mathbb{R}^D \to \mathbb{R}^D,\ x \mapsto x - \eta \nabla \mathcal{L}_i(x).$$

Let $\varphi_\omega^n = \hat{\varphi}_{\omega_n} \circ \cdots \circ \hat{\varphi}_{\omega_1}$, where $\omega = (\omega_1, \omega_2, \dots)$ uniform i.i.d. r.v..

Note: $x^* \in \mathcal{M} \Rightarrow \hat{\varphi}_i(x^*) = x^*$ for all $1 \leq i \leq N$.

Linearization - SGD

Consider the **random dynamical system** given by

$$\hat{\varphi}_i : \mathbb{R}^D \to \mathbb{R}^D,\ x \mapsto x - \eta \nabla \mathcal{L}_i(x).$$

Let $\varphi_\omega^n = \hat{\varphi}_{\omega_n} \circ \cdots \circ \hat{\varphi}_{\omega_1}$, where $\omega = (\omega_1, \omega_2, \dots)$ uniform i.i.d. r.v..

Note: $x^* \in \mathcal{M} \Rightarrow \hat{\varphi}_i(x^*) = x^*$ for all $1 \leq i \leq N$.

For $x^* \in \mathcal{M}$, let $A_i(x^*) : (\mathrm{T}_{x^*}\mathcal{M})^\perp \to (\mathrm{T}_{x^*}\mathcal{M})^\perp$ by given by

$$A_i(x^*) := \mathrm{D}\hat{\varphi}_i(x^*)|_{(\mathrm{T}_{x^*}\mathcal{M})^\perp}.$$

Linearization - SGD

Consider the **random dynamical system** given by

$$\hat{\varphi}_i : \mathbb{R}^D \to \mathbb{R}^D, \; x \mapsto x - \eta \nabla \mathcal{L}_i(x).$$

Let $\varphi_\omega^n = \hat{\varphi}_{\omega_n} \circ \cdots \circ \hat{\varphi}_{\omega_1}$, where $\omega = (\omega_1, \omega_2, \dots)$ uniform i.i.d. r.v..

Note: $x^* \in \mathcal{M} \Rightarrow \hat{\varphi}_i(x^*) = x^*$ for all $1 \leq i \leq N$.

For $x^* \in \mathcal{M}$, let $A_i(x^*) : (\mathrm{T}_{x^*}\mathcal{M})^\perp \to (\mathrm{T}_{x^*}\mathcal{M})^\perp$ by given by

$$A_i(x^*) := \mathrm{D}\hat{\varphi}_i(x^*)|_{(\mathrm{T}_{x^*}\mathcal{M})^\perp}.$$

Theorem (Furstenberg-Kesten)

The Lyapunov exponent

$$\lambda(x^*) := \lim_{n\to\infty} \frac{1}{n} \log \|A_{\omega_n}(x^*) \dots A_{\omega_1}(x^*)\|$$

exists for almost all ω and is essentially constant.

Main Results - SGD

Assume

- $\mathcal{M}$ manifold [Cooper ('21)],
- ν_{init} equivalent to Lebesgue,
- $\hat{\varphi}_i$ non-singular [Crăciun, Ghoshdastidar ('24)].

Main Results - SGD

Assume

- $\mathcal{M}$ manifold [Cooper ('21)],
- ν_{init} equivalent to Lebesgue,
- $\hat{\varphi}_i$ non-singular [Crăciun, Ghoshdastidar ('24)].

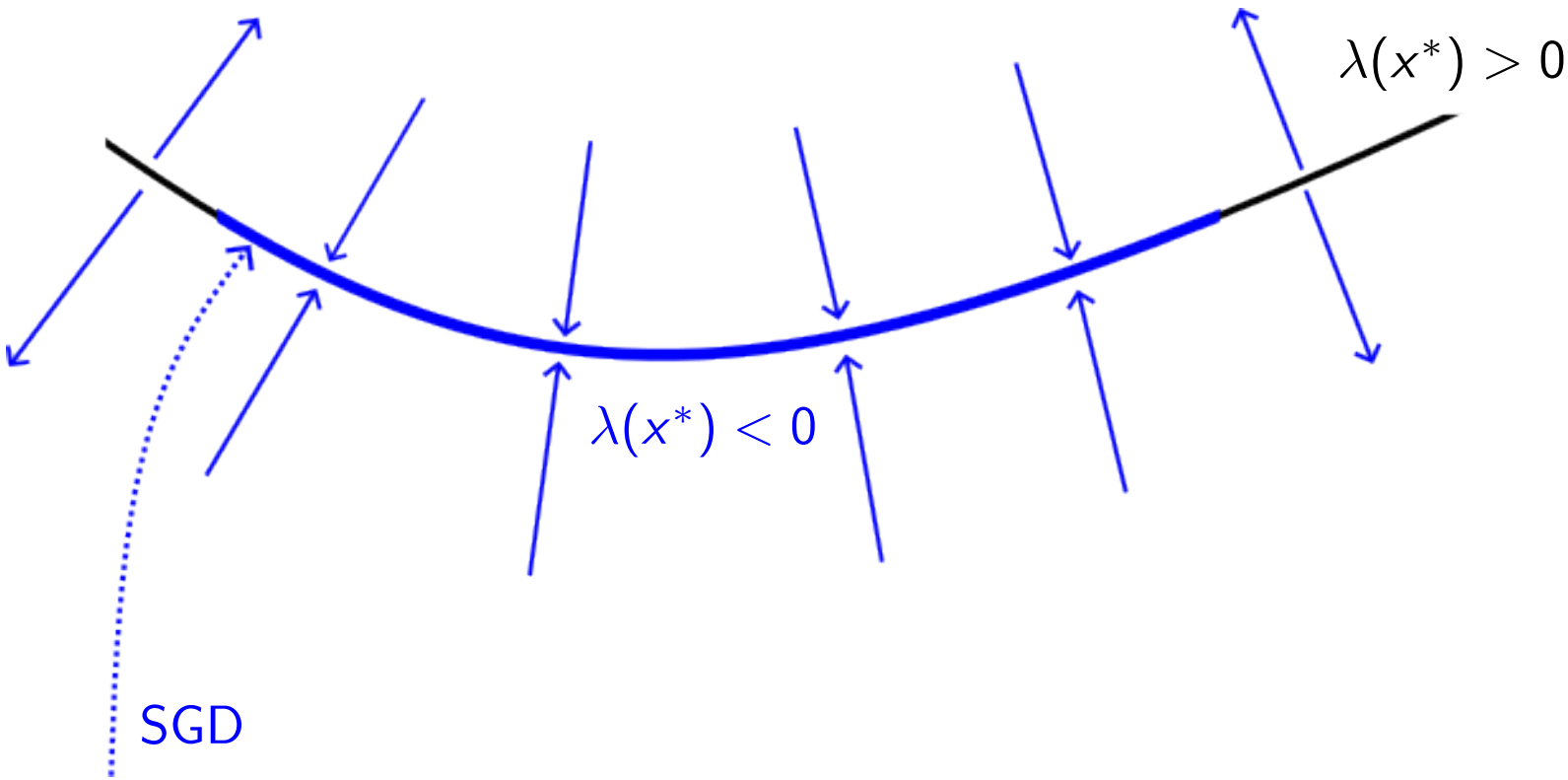
Theorem (Chemnitz/E. 2025)

Let $x^* \in \mathcal{M}$ be a "regular minimum". Then

$$\lambda(x^*) < 0 \Rightarrow x^* \in \operatorname{supp}(X_{\text{lim}}^{\text{SGD}}),$$
$$\lambda(x^*) > 0 \Rightarrow x^* \notin \operatorname{supp}(X_{\text{lim}}^{\text{SGD}}).$$

Proof strategies

- For $\lambda(x^*) < 0$, **Quenched:** "Pathwise dynamics" for typical ω
- For $\lambda(x^*) > 0$, **Annealed:** construct Lyapunov function for Markov process away from x^*



B) Random Quadratic Form on a Sphere: Synchronization by Common Noise

[E./Shalova, arxiv:2603.06187, 2026+]

Transformer models [Vaswani et al. 2017]

- Sentence of length d represented via *tokens* $(x_i)_{1 \leq i \leq d}$, $x_i \in \mathbb{R}^n$.

Transformer models [Vaswani et al. 2017]

- Sentence of length d represented via *tokens* $(x_i)_{1\leq i\leq d}$, $x_i \in \mathbb{R}^n$.
- Neural map at k-th transformer layer

$$x_i^{k+1} = \hat{\Phi}_k\left((x_i)_{1\leq i\leq d}\right),$$

with **self-attention**, **feed-forward** and projection to $\mathbb{S}^{n-1}$.

Transformer models [Vaswani et al. 2017]

- Sentence of length d represented via *tokens* $(x_i)_{1\le i\le d}$, $x_i \in \mathbb{R}^n$.
- Neural map at k-th transformer layer

$$x_i^{k+1} = \hat{\Phi}_k\left((x_i)_{1\le i\le d}\right),$$

with **self-attention**, **feed-forward** and projection to $\mathbb{S}^{n-1}$.

Continuous-time model [Geshkovski et al. 2024, 2025]

$$\dot{x}_i = P_{x_i}\left(\mathrm{FF}(x_i) + \mathrm{Attn}(x_i; x_1, x_2 \dots x_d)\right),$$
$$\mathrm{FF}(x_i) = \sigma(Mx_i + B), \quad P_{x_i} : \mathbb{R}^n \to \mathbb{S}^{n-1}.$$

Transformer models [Vaswani et al. 2017]

- Sentence of length d represented via *tokens* $(x_i)_{1\le i\le d}$, $x_i \in \mathbb{R}^n$.
- Neural map at k-th transformer layer

$$x_i^{k+1} = \hat{\Phi}_k\left((x_i)_{1\le i\le d}\right),$$

with **self-attention**, **feed-forward** and projection to $\mathbb{S}^{n-1}$.

Continuous-time model [Geshkovski et al. 2024, 2025]

$$\dot{x}_i = P_{x_i}\left(\mathrm{FF}(x_i) + \mathrm{Attn}(x_i; x_1, x_2 \dots x_d)\right),$$
$$\mathrm{FF}(x_i) = \sigma(Mx_i + B), \quad P_{x_i} : \mathbb{R}^n \to \mathbb{S}^{n-1}.$$

Simplified model ($\mathrm{Attn} = 0 = B$) for random parametrization at each k:

Transformer models [Vaswani et al. 2017]

- Sentence of length d represented via *tokens* $(x_i)_{1\le i\le d}$, $x_i \in \mathbb{R}^n$.
- Neural map at k-th transformer layer

$$x_i^{k+1} = \hat{\Phi}_k\left((x_i)_{1\le i\le d}\right),$$

with **self-attention**, **feed-forward** and projection to $\mathbb{S}^{n-1}$.

Continuous-time model [Geshkovski et al. 2024, 2025]

$$\dot{x}_i = P_{x_i}\left(\mathrm{FF}(x_i) + \mathrm{Attn}(x_i; x_1, x_2 \dots x_d)\right),$$
$$\mathrm{FF}(x_i) = \sigma(Mx_i + B), \quad P_{x_i} : \mathbb{R}^n \to \mathbb{S}^{n-1}.$$

Simplified model ($\mathrm{Attn} = 0 = B$) for random parametrization at each k:

$$\mathrm{d}X_i(t) = P_{X_i(t)}\partial Q(t)X_i(t), \qquad \textbf{(Random Quadratic Form (RQF))}$$
$$Q_t = \frac{1}{2}(B_t + B_t^T),$$

where $\{B_t^{ij} : i, j \in 1 \dots n\}$ are independent Brownian motions.

Synchronization of tokens purely by common noise?

Synchronization of tokens purely by common noise?

Consider the RDS solutions $\varphi_\omega^t(x)$ of

$$\mathrm{d}X(t) = P_{X(t)}\partial Q(t)X(t),\ X(0) = x \in \mathbb{S}^{n-1}. \qquad \textbf{(RQF)}$$

Synchronization of tokens purely by common noise?

Consider the RDS solutions $\varphi_\omega^t(x)$ of

$$\mathrm{d}X(t) = P_{X(t)}\partial Q(t)X(t),\ X(0) = x \in \mathbb{S}^{n-1}. \qquad \textbf{(RQF)}$$

Theorem (E./Shalova 2026+)

For any $x, y \in \mathbb{S}^{n-1}$

$$\mathrm{dist}(\varphi_\omega^t(y), \varphi_\omega^t(x)) \to 0 \quad \textit{or} \quad \mathrm{dist}(\varphi_\omega^t(y), -\varphi_\omega^t(x)) \to 0,$$

Ω-*almost surely.*

Synchronization of tokens purely by common noise?

Consider the RDS solutions $\varphi_\omega^t(x)$ of

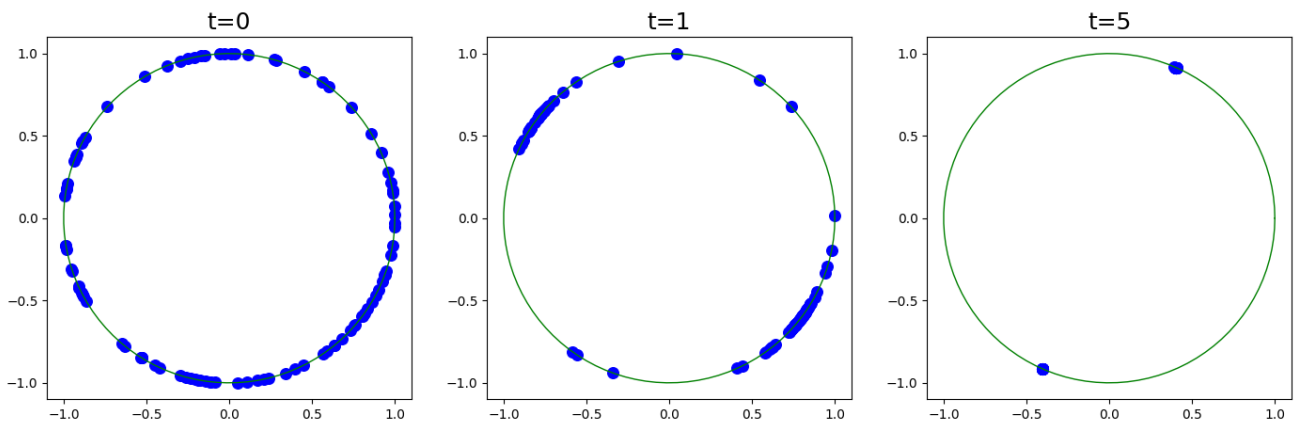
$$\mathrm{d}X(t) = P_{X(t)}\partial Q(t)X(t),\ X(0) = x \in \mathbb{S}^{n-1}. \qquad \textbf{(RQF)}$$

Theorem (E./Shalova 2026+)

For any $x, y \in \mathbb{S}^{n-1}$

$$\operatorname{dist}(\varphi_\omega^t(y), \varphi_\omega^t(x)) \to 0 \quad \textit{or} \quad \operatorname{dist}(\varphi_\omega^t(y), -\varphi_\omega^t(x)) \to 0,$$

Ω-almost surely.



Proof strategies

First Proof:

- For solutions $X_t,\ Y_t$ of (RQF), write $Z_t = \langle X_t, Y_t \rangle \in [-1, 1]$.

Proof strategies

First Proof:

- For solutions $X_t,\ Y_t$ of (RQF), write $Z_t = \langle X_t, Y_t \rangle \in [-1, 1]$.
- Derive that Z_t has same law as process

$$\mathrm{d}\hat{Z}_t = 2\hat{Z}_t(1 - \hat{Z}_t^2)\mathrm{d}t + \sqrt{2}(1 - \hat{Z}_t^2)\partial B_t,$$

Proof strategies

First Proof:

- For solutions $X_t,\ Y_t$ of (RQF), write $Z_t = \langle X_t, Y_t \rangle \in [-1, 1]$.
- Derive that Z_t has same law as process

$$\mathrm{d}\hat{Z}_t = 2\hat{Z}_t(1 - \hat{Z}_t^2)\mathrm{d}t + \sqrt{2}(1 - \hat{Z}_t^2)\partial B_t,$$

- Use **Feller criterion** to show that

$$\mathbb{P}(\hat{Z}_t \to 1) + \mathbb{P}(\hat{Z}_t \to -1) = 1.$$

Proof strategies

First Proof:

- For solutions $X_t,\ Y_t$ of (RQF), write $Z_t = \langle X_t, Y_t \rangle \in [-1, 1]$.
- Derive that Z_t has same law as process

$$\mathrm{d}\hat{Z}_t = 2\hat{Z}_t(1 - \hat{Z}_t^2)\mathrm{d}t + \sqrt{2}(1 - \hat{Z}_t^2)\partial B_t,$$

- Use **Feller criterion** to show that

$$\mathbb{P}(\hat{Z}_t \to 1) + \mathbb{P}(\hat{Z}_t \to -1) = 1.$$

Second Proof:

- Show that **Lyapunov exponent** $\lambda < 0$ for RDS induced by (RQF)

Proof strategies

First Proof:

- For solutions $X_t,\ Y_t$ of (RQF), write $Z_t = \langle X_t, Y_t \rangle \in [-1, 1]$.
- Derive that Z_t has same law as process

$$\mathrm{d}\hat{Z}_t = 2\hat{Z}_t(1 - \hat{Z}_t^2)\mathrm{d}t + \sqrt{2}(1 - \hat{Z}_t^2)\partial B_t,$$

- Use **Feller criterion** to show that

$$\mathbb{P}(\hat{Z}_t \to 1) + \mathbb{P}(\hat{Z}_t \to -1) = 1.$$

Second Proof:

- Show that **Lyapunov exponent** $\lambda < 0$ for RDS induced by (RQF)
- Write process on projective space $\mathbb{RP}^n$

Proof strategies

First Proof:

- For solutions $X_t,\ Y_t$ of (RQF), write $Z_t = \langle X_t, Y_t \rangle \in [-1, 1]$.
- Derive that Z_t has same law as process

$$\mathrm{d}\hat{Z}_t = 2\hat{Z}_t(1 - \hat{Z}_t^2)\mathrm{d}t + \sqrt{2}(1 - \hat{Z}_t^2)\partial B_t,$$

- Use **Feller criterion** to show that

$$\mathbb{P}(\hat{Z}_t \to 1) + \mathbb{P}(\hat{Z}_t \to -1) = 1.$$

Second Proof:

- Show that **Lyapunov exponent** $\lambda < 0$ for RDS induced by (RQF)
- Write process on projective space $\mathbb{RP}^n$
- Use synchronization criteria [Baxendale 1991], [Flandoli et al. 2017]