Implicit regularisation of gradient algorithms

Scott Pesme

Loucas Pillaud-Vivien

Nicolas Flammarion

TML lab

EPFL

Published as a conference paper at ICLR 2020

Gradient Descent Maximizes the Margin of Homogeneous Neural Networks

Kaifeng Lyu & Jian Li
Institute for Interdisciplinary Information Sciences
Tsinghua University
Beijing, China
vfleaking@gmail.com,lijian83@mail.tsinghua.edu.cn

Journal of Machine Learning Research 19 (2018) 1-57 Submitted 4/18; Published 11/18

The Implicit Bias of Gradient Descent on Separable Data

Daniel Soudry DANIEL.SOUDRY@GMAIL.COM
Elad Hoffer ELAD.HOFFER@GMAIL.COM
Mor Shpigel Nacson MOR.SHPIGEL@GMAIL.COM
Department of Electrical Engineering,Technion
Haifa, 320003, Israel

Suriya Gunasekar SURIYA@TTIC.EDU
Nathan Srebro NATI@TTIC.EDU
Toyota Technological Institute at Chicago
Chicago, Illinois 60637, USA

Proceedings of Machine Learning Research vol 125:1–34, 2020 33rd Annual Conference o

Implicit Bias of Gradient Descent for Wide Two-layer Neural Networks Trained with the Logistic Loss

Lénaïc Chizat LENAIC.CHIZAT@UNIVERSITE-PARIS-SACLAY.FR
Laboratoire de Mathématiques d'Orsay, CNRS, Université Paris-Saclay, France

Francis Bach FRANCIS.BACH@INRIA.FR
search University, Paris, France

Implicit Bias of Gradient Descent on Linear Convolutional Networks

Suriya Gunasekar
TTI at Chicago, USA
suriya@ttic.edu

Jason D. Lee
USC Los Angeles, USA
jasonlee@marshall.usc.edu

Daniel Soudry
Technion, Israel
daniel.soudry@gmail.com

Nathan Srebro
TTI at Chicago, USA
nati@ttic.edu

Implicit Bias in Deep Linear Classification: Initialization Scale vs Training Accuracy

Edward Moroshko
edward.moroshko@gmail.com
Technion

Blake Woodworth
blake@ttic.edu
TTI Chicago

Suriya Gunasekar
suriya@ttic.edu
Microsoft Research

Jason D. Lee
asonlee@princeton.edu
Princeton University

Nathan Srebro
nati@ttic.edu
TTI Chicago

Daniel Soudry
daniel.soudry@gmail.com
Technion

Published as a conference paper at ICLR 2019

Gradient Descent Aligns the Layers of Deep Linear Networks

Ziwei Ji & Matus Telgarsky
Department of Computer Science
University of Illinois at Urbana-Champaign
{ziweiji2,mjt}@illinois.edu

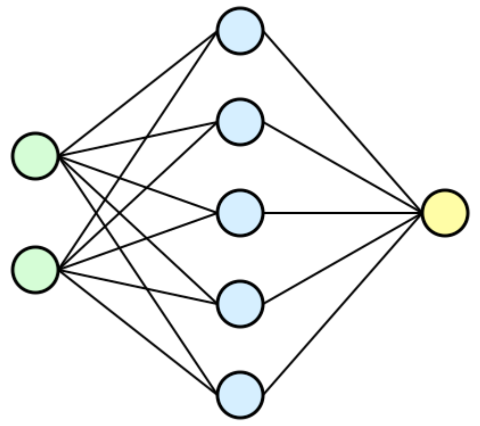
Proceedings of Machine Learning Research vol 134:1–35, 2021 34th Annual Conference on Learning Theory

Implicit Regularization in ReLU Networks with the Square Loss

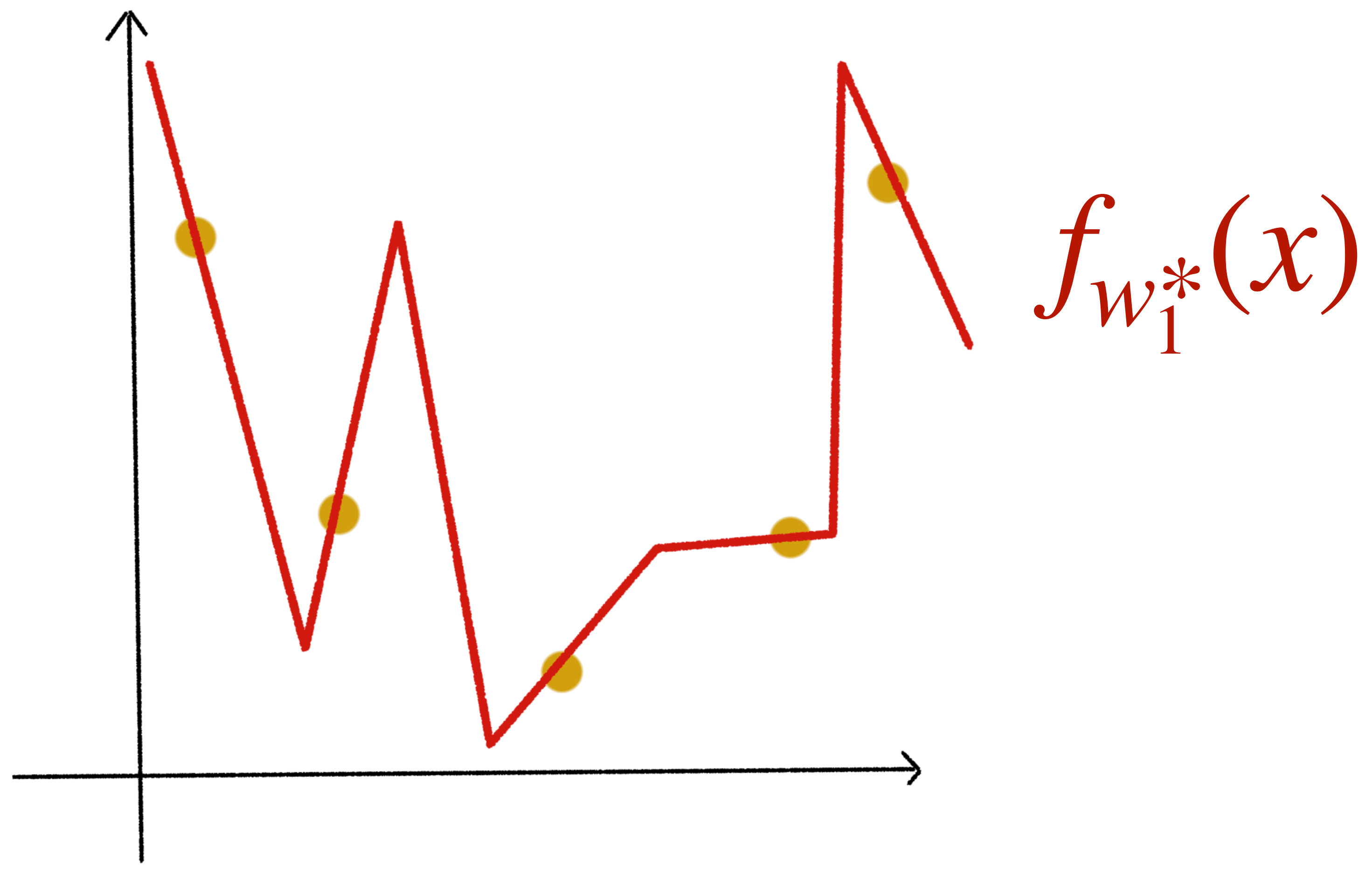
Gal Vardi GAL.VARDI@WEIZMANN.AC.IL and **Ohad Shamir** OHAD.SHAMIR@WEIZMANN.AC.IL
Weizmann Institute of Science

Motivation

Training architecture



An infinity of interpolating solutions

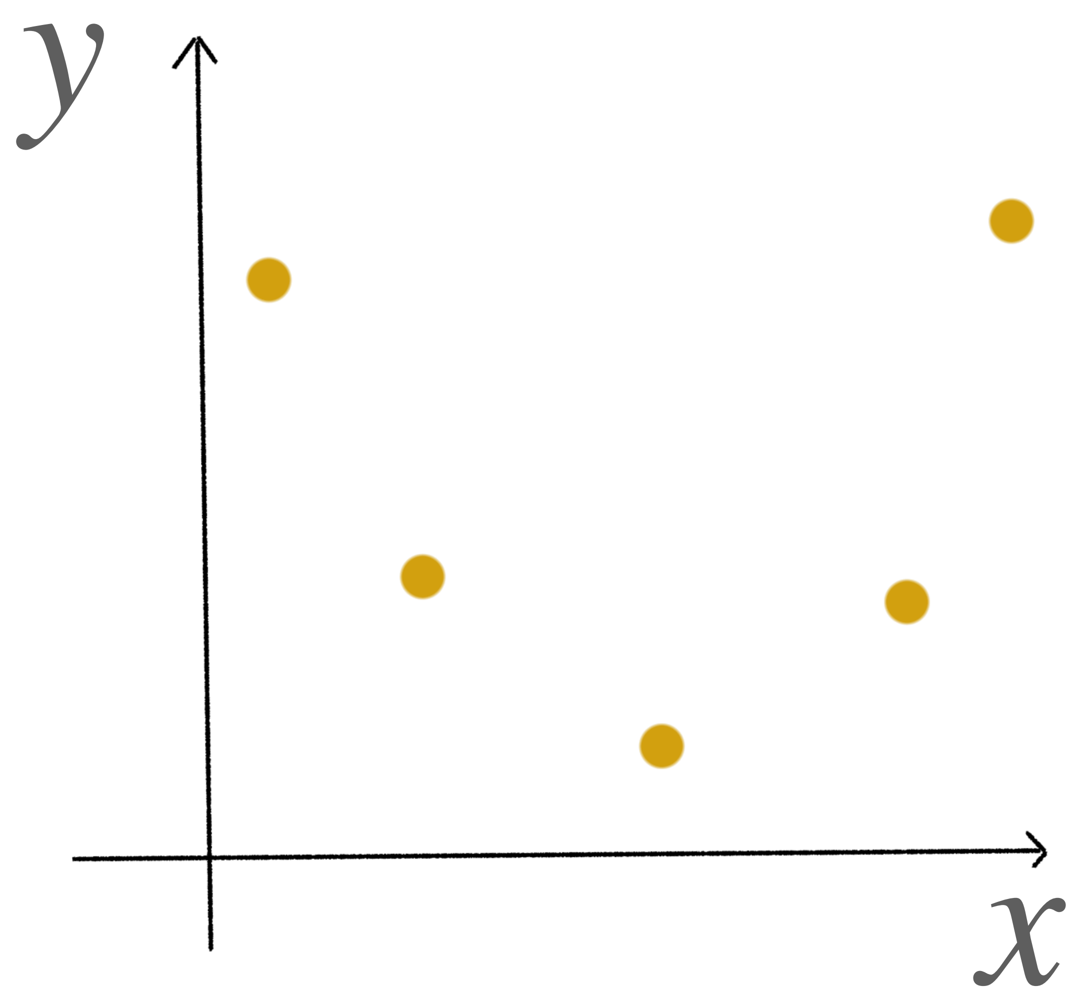


$\mathbb{E}[\ell(f_{w_1^*}(x), y)] \gg 1$

Training dataset

Training algorithm

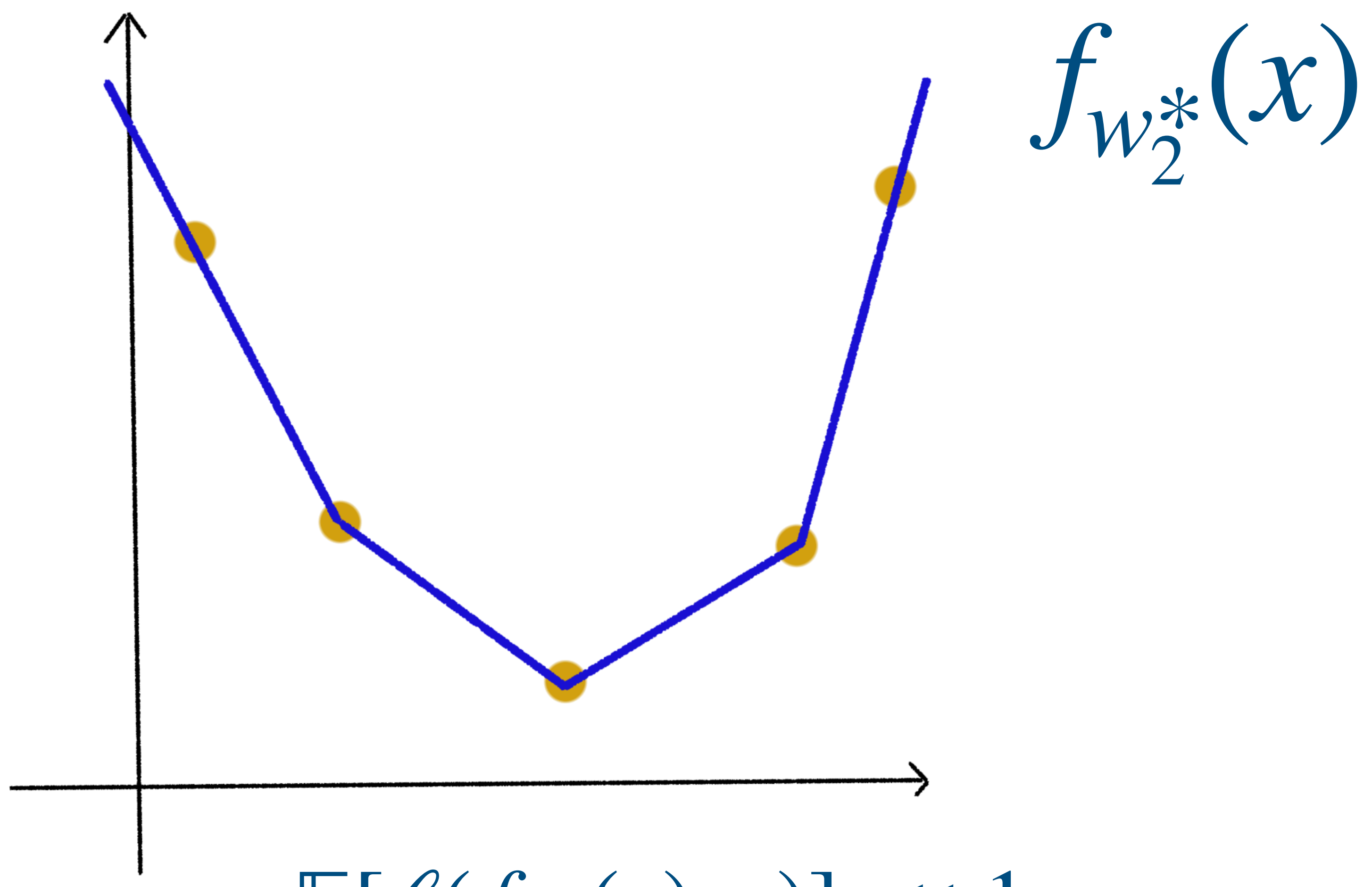
Some ERM



GD

SGD

etc.



$\mathbb{E}[\ell(f_{w_2^*}(x), y)] \ll 1$

A panoply of algorithms:

Some ERM

GD

SGD

+ momentum

Batch Normalisation

Etc.

All can lead to zero training error but do not generalise the same.

Cifar-10 dataset:

Classification task!

ResNet-18 (with BN):

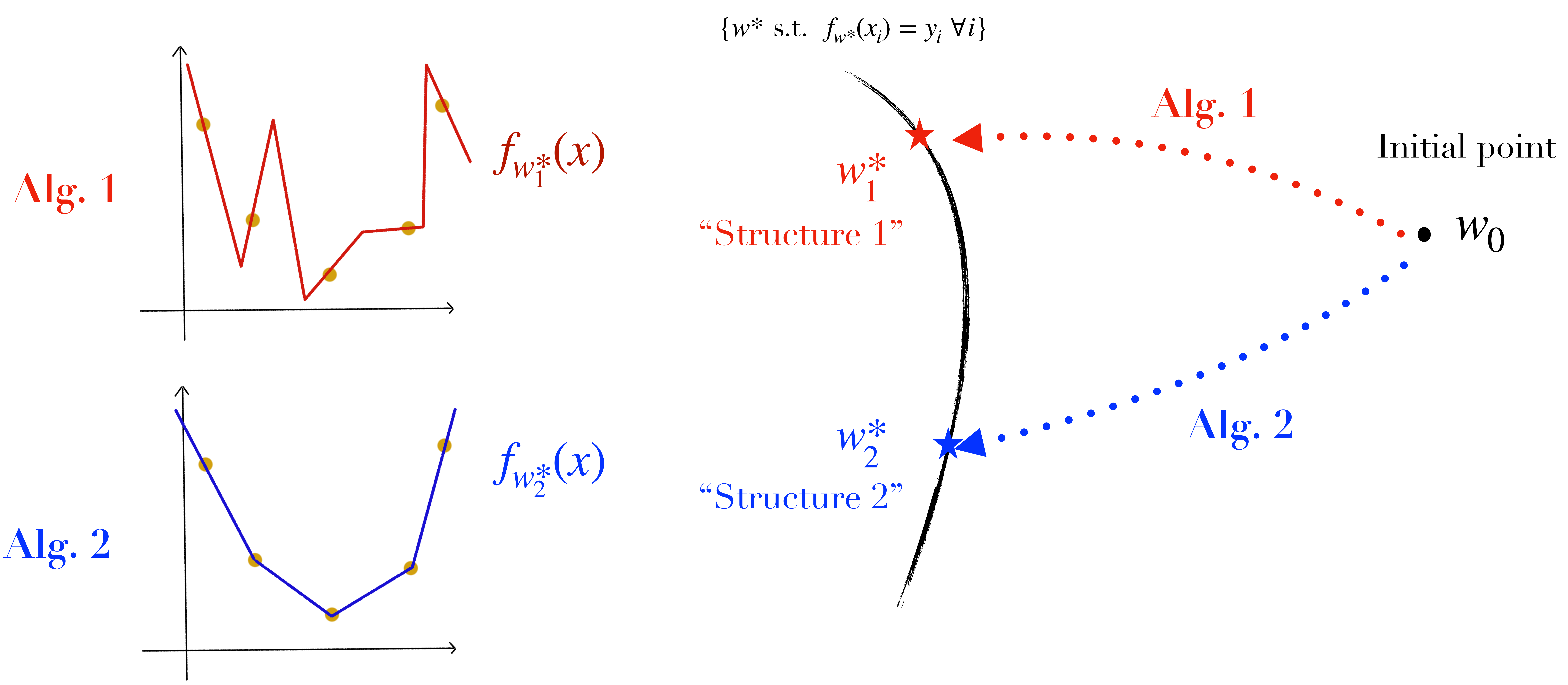
	Train accuracy	**Test accuracy**
GD	100%	~72% (??)
SGD	100%	85%
SGD + momentum	100%	89%
SGD + mom + DA + ℓ_2	100%	95%
GD + mom + DA + ℓ_2	100%	87%

Stochastic Training is Not Necessary for Generalization, Geiping et al. 2021

Bad Global Minima Exist and SGD Can Reach Them, Liu et al. 2020

Revisiting Small Batch Training For Deep Neural Networks, Masters and Luschi 2018

What does all this mean ?



"**Algorithmic implicit bias**" : the algorithm "**chooses**" a particular solution.

Implicit bias and minimal norm solutions (for regression)

Minimise $L(w) = \frac{1}{2n} \sum_{i=1}^{n} (f_w(x_i) - y_i)^2$ with some algorithm.

"It turns out":

$$L(w_\infty^{alg}) = 0 \quad \Leftrightarrow \quad f_{w_\infty^{alg}}(x_i) = y_i$$

and

w_∞^{alg} enjoys a "nice" structure

$$\equiv$$

$$\boxed{w_\infty^{alg} = \underset{w, \forall i,\ f_w(x_i) = y_i}{\arg\min} \ R_{alg}(w)}$$

$\|w\|_2$

$\|w\|_1$

Contrast with explicit regularisation:

$$\min_{w \in \mathbb{R}^d} \ RegL(w) := \frac{1}{n} \sum_{i=1}^{n} \ell_i(w) + \lambda\, R(w)$$

(Often unique)
(Not an interpolator !)

A few disclaimers:

We hope to exhibit: $$\boxed{w_\infty^{alg} = \underset{w, \forall i,\ f_w(x_i) = y_i}{\arg\min}\ R_{alg}(w)}$$

- We will only consider **regression** tasks

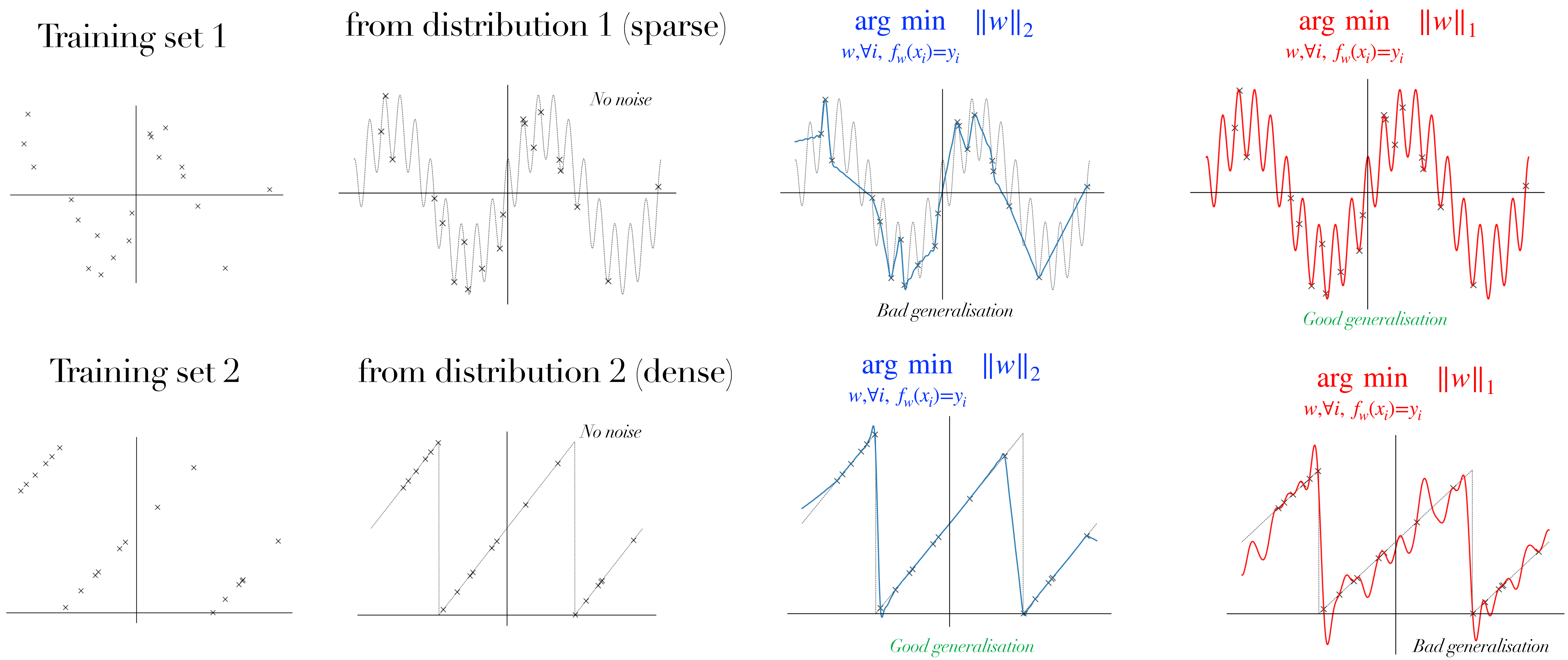
- The characterisation of the solution **does not** (**on its own**) **say anything about generalisation !**

- overfitting the training set is not always good (but often works in practice)
- the benign overfitting literature (partly) covers the generalisation properties of min norm interpolators
- the generalisation questions **depend on the true distribution**, but not the implicit regularisation problem

Toy examples: implicit bias doesn't explain (on its own) generalisation

Samples $(x_i, y_i)_{1 \leq i \leq n} \in \mathbb{R} \times \mathbb{R}$ from some distribution $\mathscr{D}$.

We want to linearly interpolate with feature expansion $\phi(x) = (\frac{1}{i}\cos(2\pi i x), \frac{1}{i}\sin(2\pi i x))_{1 \leq i \leq d/2} \in \mathbb{R}^d$: $\quad f_w(x) = \langle w, \phi(x) \rangle$.



Depending on the true data distribution, "Structure 1" $\lessgtr$ "Structure 2".

Back to implicit bias / regularisation:

Can you give me a simple example showing this phenomenon ?

Simplest example: linear regression

Unique root loss (can be non-convex)

$$L(w) = \frac{1}{2n}\sum_{i=1}^{n}(y_i - \langle w, x_i\rangle)^2$$

But more generally still true with $L(w) = \frac{1}{n}\sum_{i=1}^{n}\ell(y_i, \langle w, x_i\rangle)$

$f_w(x) = \langle w, x\rangle$

GD: $w_{t+1} = w_t + \underbrace{\gamma \frac{1}{n}\sum_i (y_i - \langle x_i, w_t\rangle)x_i}_{\in \operatorname{span}(x_1, \ldots, x_n)}$

SGD: $w_{t+1} = w_t + \underbrace{\gamma(y_{i_t} - \langle x_{i_t}, w_t\rangle)x_{i_t}}_{\in \operatorname{span}(x_1, \ldots, x_n)}$

Implicit regularisation

$w_\infty^{GD/SGD} \in w_0 + \operatorname{span}(x_1, \ldots, x_n)$

$\forall i,\ \langle w_\infty^{GD/SGD}, x_i\rangle = y_i$

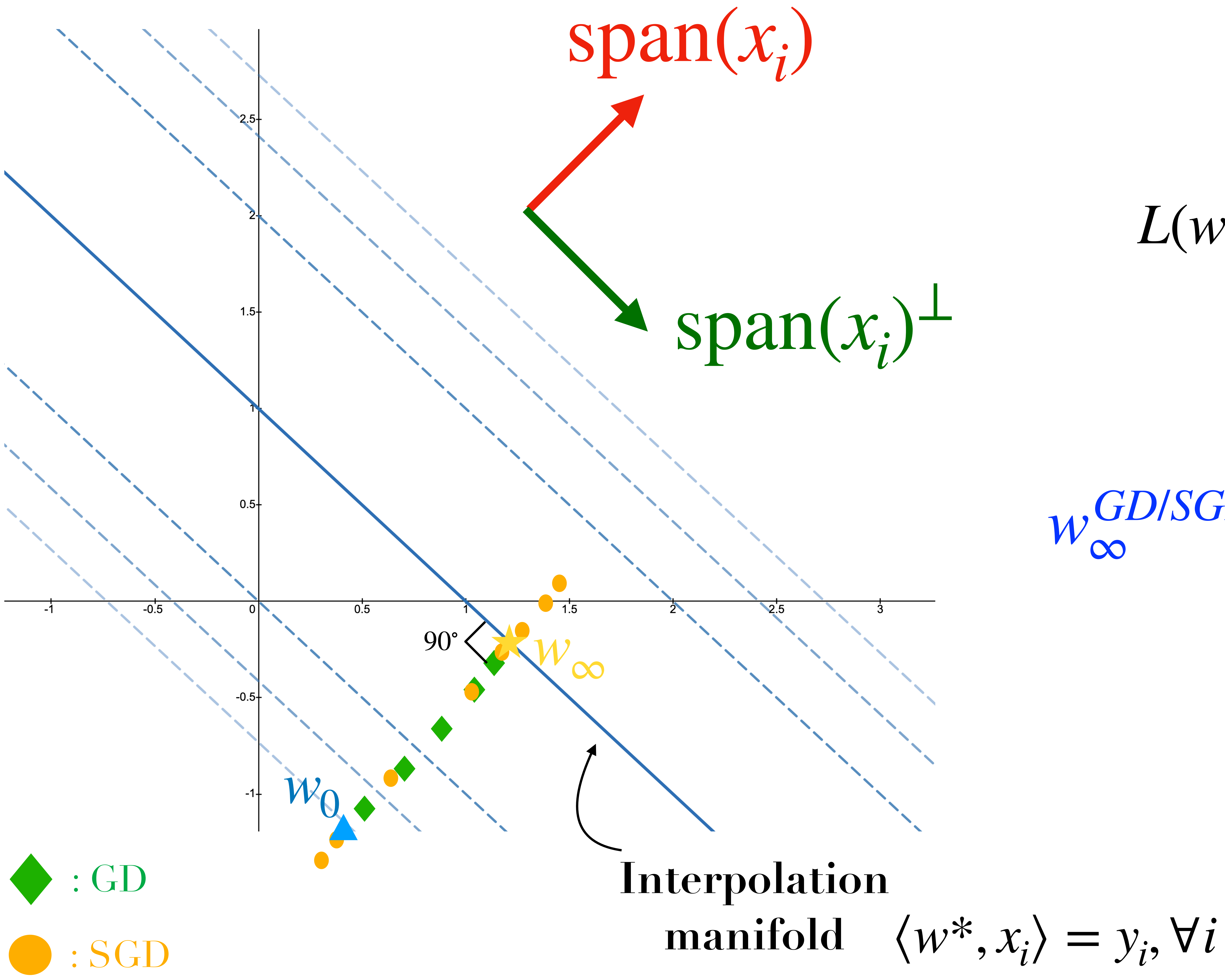
implies

(Pythagorean Theorem)

Pythagoras et al. 500 BC

$$w_\infty^{GD/SGD} = \operatorname*{argmin}_{w, \forall i,\ \langle w, x_i\rangle = y_i} \|w - w_0\|_2^2$$

Simplest example: linear regression



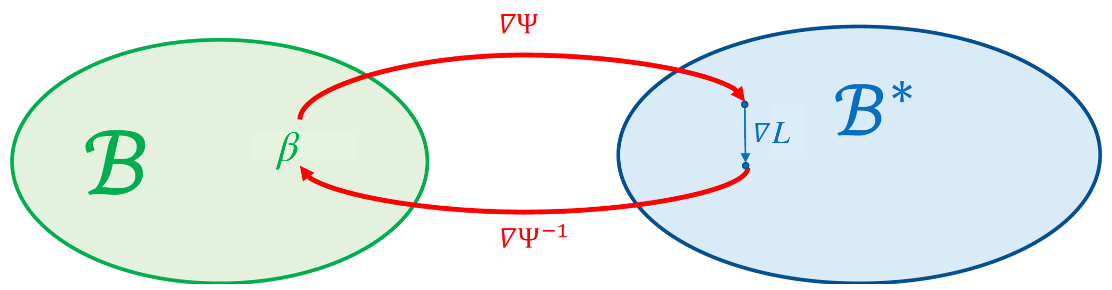
$$L(w) = \frac{1}{2n} \sum_{i=1}^{n} (y_i - \langle w, x_i \rangle)^2$$

Implicit regularisation

$$w_\infty^{GD/SGD} = \underset{w, \forall i,\ \langle w, x_i \rangle = y_i}{\operatorname{argmin}} \|w - w_0\|_2^2$$

Second simplest: Mirror descent

$$\nabla\Psi(\beta_{t+1}) = \nabla\Psi(\beta_t) - \gamma\nabla L(\beta_t)$$



Ψ is a convex and differentiable potential.

$\Psi(\beta) = \|\beta\|_2^2$: back to GD

MD: $\nabla\Psi(\beta_{t+1}) = \nabla\Psi(\beta_t) + \underbrace{\gamma\frac{1}{n}\sum_i (y_i - \langle x_i, \beta_t\rangle)x_i}_{\in\, \mathrm{span}(x_1,\ldots,x_n)}$

SMD: $\nabla\Psi(\beta_{t+1}) = \nabla\Psi(\beta_t) + \underbrace{\gamma(y_{i_t} - \langle x_{i_t}, \beta_t\rangle)x_{i_t}}_{\in\, \mathrm{span}(x_1,\ldots,x_n)}$

β_∞^{MD} ?

$\nabla\Psi(\beta_\infty^{MD}) \in \nabla\Psi(\beta_0) + \mathrm{span}(x_i)$

$\forall i,\ \langle \beta_\infty^{MD}, x_i\rangle = y_i$

$\Longrightarrow$ (Pythagorean Theorem)

$$\beta_\infty^{MD} = \underset{\beta,\ \langle\beta, x_i\rangle = y_i}{\mathrm{argmin}}\ D_\Psi(\beta, \beta_0)$$

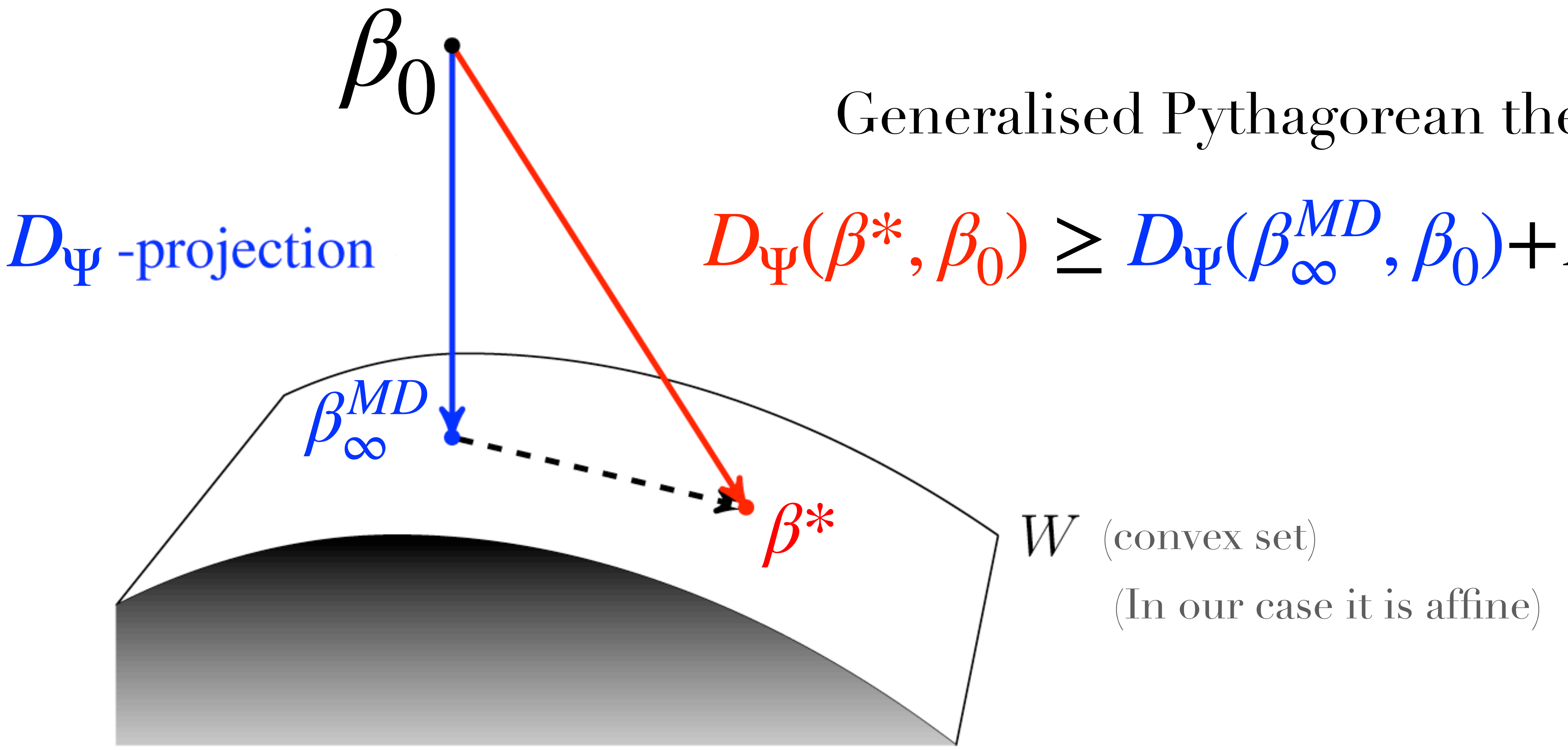
Gunasekar, S., Lee, J., Soudry, D., and Srebro, N. Characterizing implicit bias in terms of optimization geometry. ICML2018

Regression with linear models, recap.

$$L(w) = \frac{1}{2n}\sum_{i=1}^{n}(y_i - \langle w, x_i\rangle)^2$$

$$w_\infty^{SGD} = w_\infty^{GD} = \underset{w,\ \forall i,\ \langle w,x_i\rangle = y_i}{\arg\min} \ \|w - w_0\|_2^2 \quad (= w_\infty^{mom} = w^{GF} = \ldots)$$

$$\beta_\infty^{MD} = \beta_\infty^{SMD} = \underset{w,\ \forall i,\ \langle \beta,x_i\rangle = y_i}{\arg\min} \ D_\Psi(\beta, \beta_0) \quad (= \beta_\infty^{mom} = \beta^{MF} = \ldots) \quad (= \mathrm{Proj}^{\Psi}_{\{\langle \beta,x_i\rangle = y_i\}}(\beta_0))$$



Generalised Pythagorean theorem:

$$D_\Psi(\beta^*, \beta_0) \geq D_\Psi(\beta_\infty^{MD}, \beta_0) + D_\Psi(\beta^*, \beta_\infty^{MD})$$

MD on linear models:

$$\beta_\infty^{MD} = \beta_\infty^{SMD} = \underset{w,\ \forall i,\ \langle \beta, x_i \rangle = y_i}{\arg\min}\ D_\Psi(\beta, \beta_0)$$

For the intuition, recall that $\beta_{t+1} = \underset{\beta \in \mathbb{R}^d}{\operatorname{argmin}}\ L(\beta_t) + \langle \nabla L(\beta_t), \beta - \beta_t \rangle + \frac{1}{\gamma} D_\Psi(\beta, \beta_t)$

➡ Independent of stochasticity

But who (in DL) cares about mirror descent ?

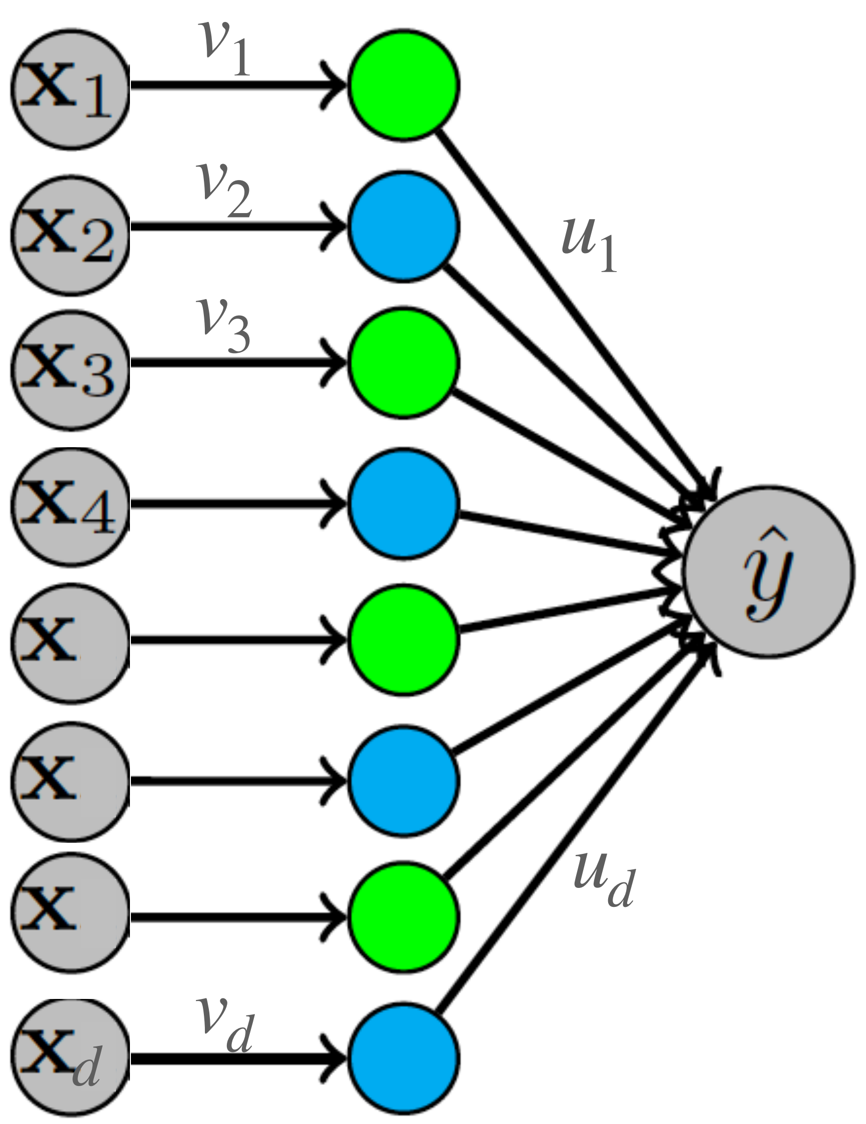
Mirror descent is a framework in which things are easy:

- Convergence of the iterates and of the training loss
- Tight rates
- Implicit bias

A "practical example": 2-layer diagonal linear network.

Architecture

Diagonal linear network :



$$f_w(x) = \langle u \odot v, x \rangle$$

$$w = \begin{pmatrix} u \\ v \end{pmatrix} \in \mathbb{R}^{2d}$$

Square-loss

$$\min_{w \in \mathbb{R}^{2d}} L(w) = \frac{1}{4n} \sum_{i=1}^{n} (y_i - \langle \underbrace{u \odot v}_{\beta_w}, x_i \rangle)^2$$

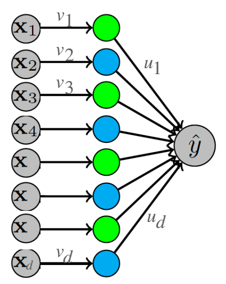
Non-convex in w

Final model is linear: but training is changed.

Gunasekar, S., Woodworth, B., Bhojanapalli, S., Neyshabur, B., Srebro, N. . Implicit regularization in matrix factorization NeurIPS 2017
Li, Y., Ma, T. and Zhang, H. Algorithmic Regularization in Over-parameterized Matrix Recovery. *COLT 2018*
Vaskevicius, T., Kanade, V., and Rebeschini, P. Implicit regularization for optimal sparse recovery. NeurIPS 2019, etc...

Hidden mirror descent

Setting $\quad L(w) = \frac{1}{4n}\sum_{i=1}^{n}(y_i - \langle u \odot v, x_i\rangle)^2$



$w = \begin{pmatrix} u \\ v \end{pmatrix} \in \mathbb{R}^{2d} \qquad \beta_w := u \odot v \in \mathbb{R}^d$

Gradient **flow** on the neurons:

$$\mathrm{d}u_t = -\nabla_u L(w_t)\ \mathrm{d}t \qquad u_{t=0} = \alpha \mathbf{1} \in \mathbb{R}^d$$

$$\mathrm{d}v_t = -\nabla_v L(w_t)\ \mathrm{d}t \qquad v_{t=0} = \mathbf{0} \in \mathbb{R}^d$$

$$\beta_{w_{t=0}} = \mathbf{0} \in \mathbb{R}^d$$

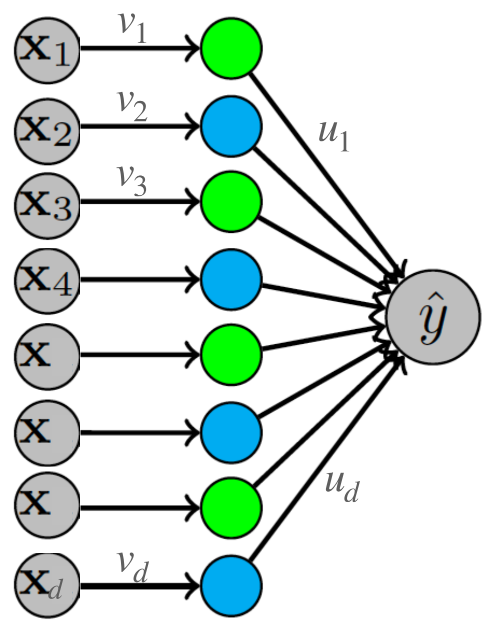
What about the dynamics of $\beta_t := \beta_{w_t} = u_t \odot v_t$? Turns out that:

$$\frac{\mathrm{d}\nabla\phi_\alpha(\beta_t)}{\mathrm{d}t} = -\nabla_\beta L(\beta_t)$$

i.e. continuous mirror descent with $\Psi = \phi_{\underbrace{\alpha}_{\text{Initialisation scale !}}}$

Implicit bias of the gradient flow for 2 layer diagonal linear network

Architecture: $$\min_{w \in \mathbb{R}^{2d}} L(w) = \frac{1}{4n} \sum_{i=1}^{n} (y_i - \langle \underbrace{u \odot v}_{\beta_w}, x_i \rangle)^2$$



Algorithm: Gradient flow **on the neurons** w_t starting at $u_{t=0} = \alpha \mathbf{1} \in \mathbb{R}^d$, $v_{t=0} = \mathbf{0} \in \mathbb{R}^d$

Implicit bias: $$\beta_\infty^\alpha = \arg\min_{\beta,\ \langle \beta, x_i \rangle = y_i} \phi_\alpha(\beta) \quad (= D_{\phi_\alpha}(\beta, \beta_0 = 0))$$

where $\phi_\alpha \underset{\alpha \to \infty}{\sim} \|\cdot\|_2$ and $\phi_\alpha \underset{\alpha \to 0}{\sim} \|\cdot\|_1$

(+ **convergence, rates etc.**)

Woodworth, B., Gunasekar, S., Lee, J. D., Moroshko, E., Savarese, P., Golan, I., Soudry, D., and Srebro, N. Kernel and rich regimes in overparametrized models. COLT 2020

Implicit bias:

$$\beta_\infty^\alpha = \arg\min_{\beta,\ \langle\beta, x_i\rangle = y_i} \phi_\alpha(\beta)$$

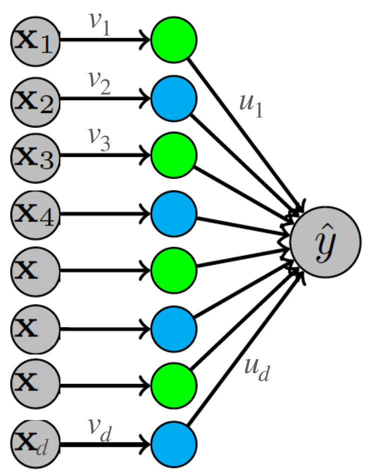
GF on the neurons

$u_{t=0} = \alpha \mathbf{1} \in \mathbb{R}^d$

$v_{t=0} = \mathbf{0} \in \mathbb{R}^d$

MD on the predictor

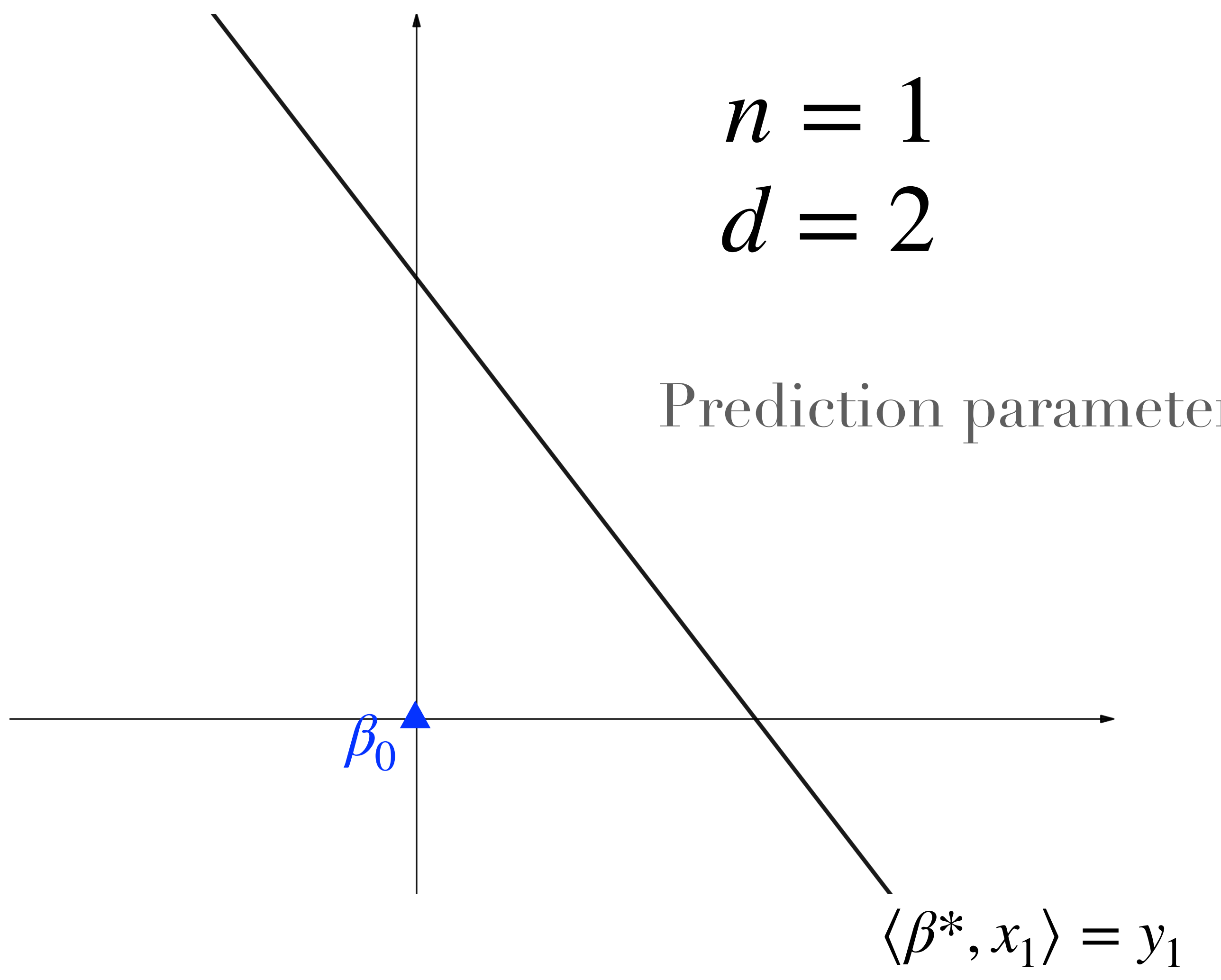
$\beta_t := u_t \odot v_t$?



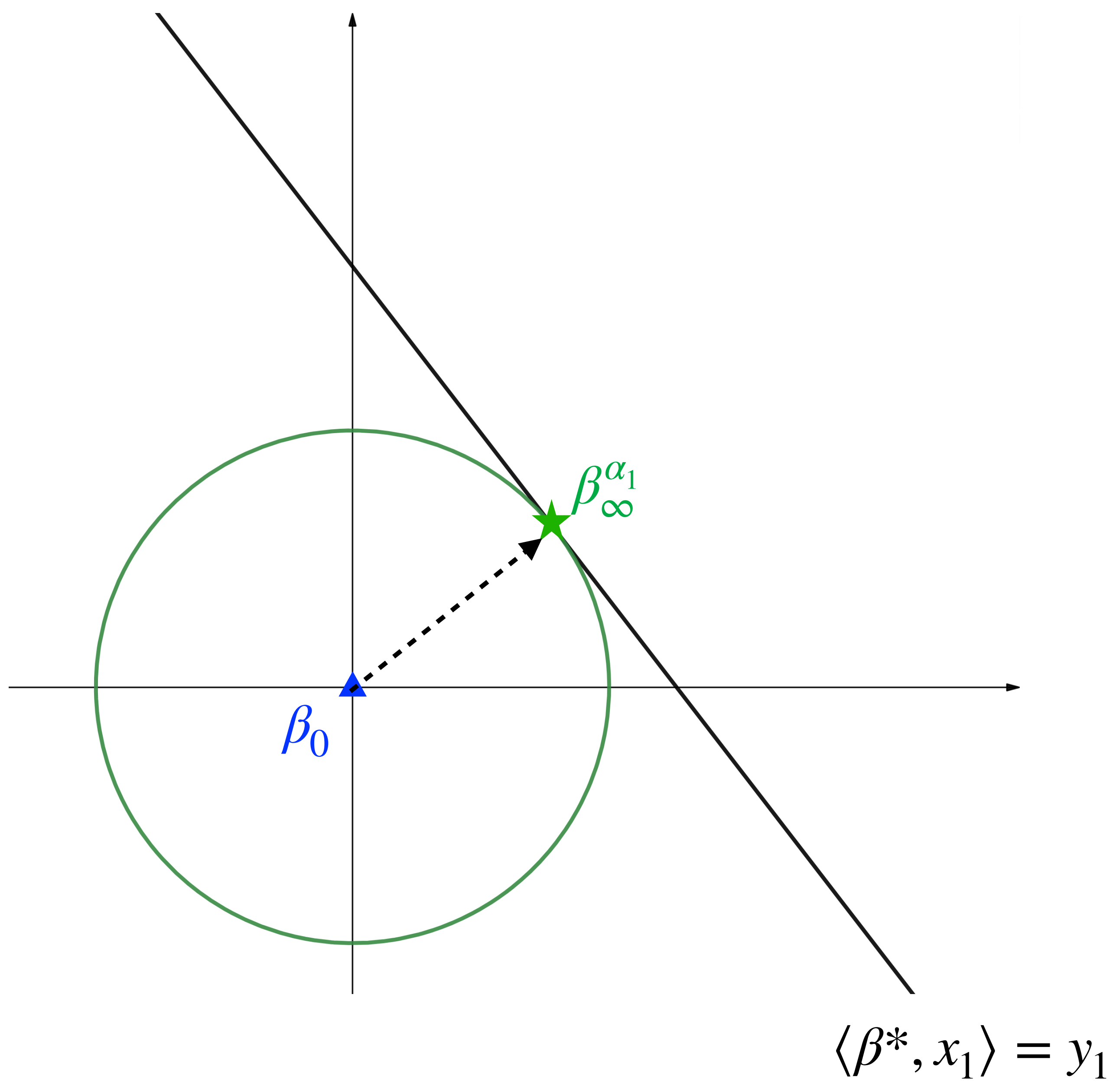
where $\phi_\alpha \underset{\alpha \to 0}{\sim} \|\cdot\|_1$

$\phi_\alpha \underset{\alpha \to \infty}{\sim} \|\cdot\|_2$

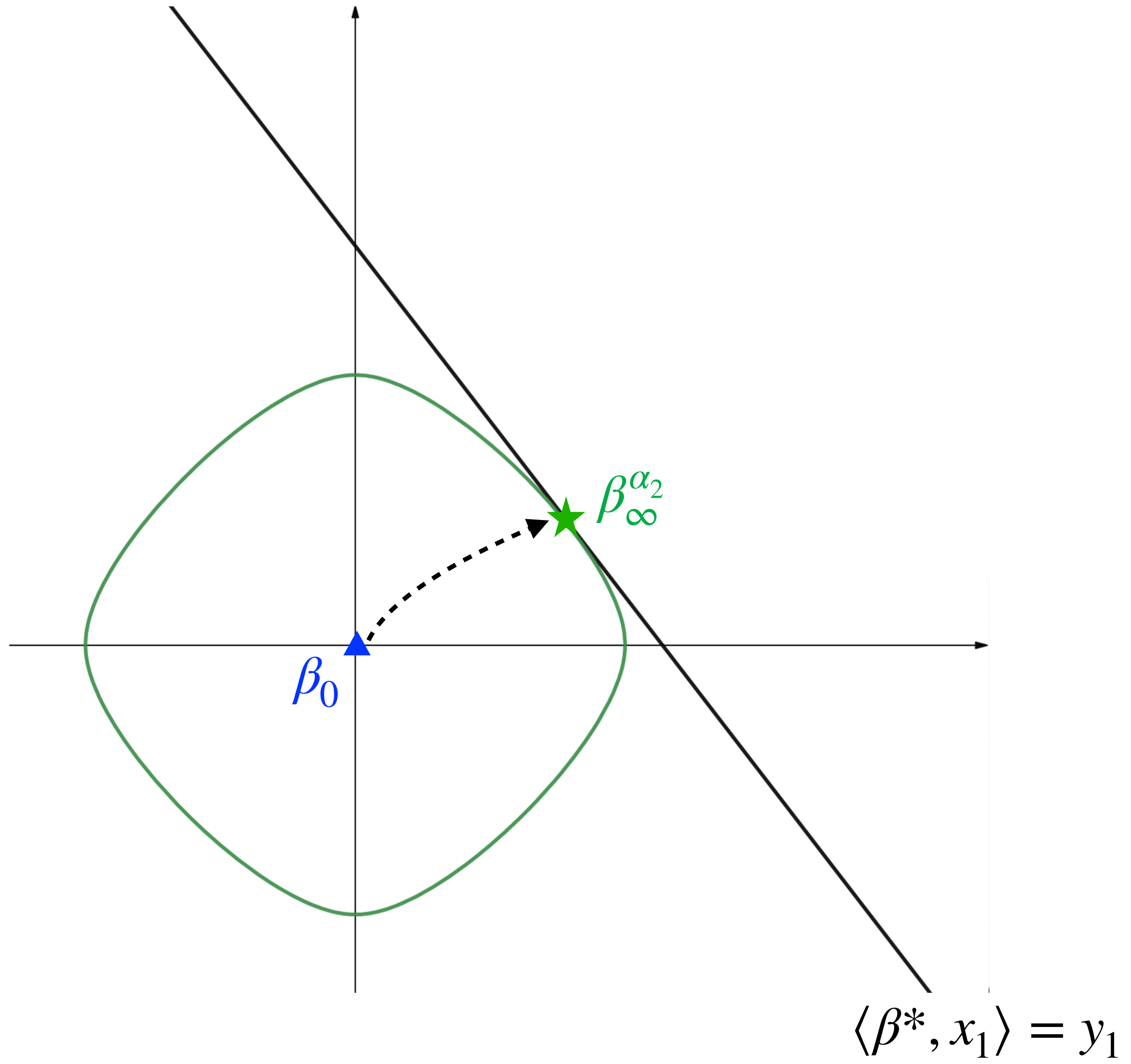
Toy illustration



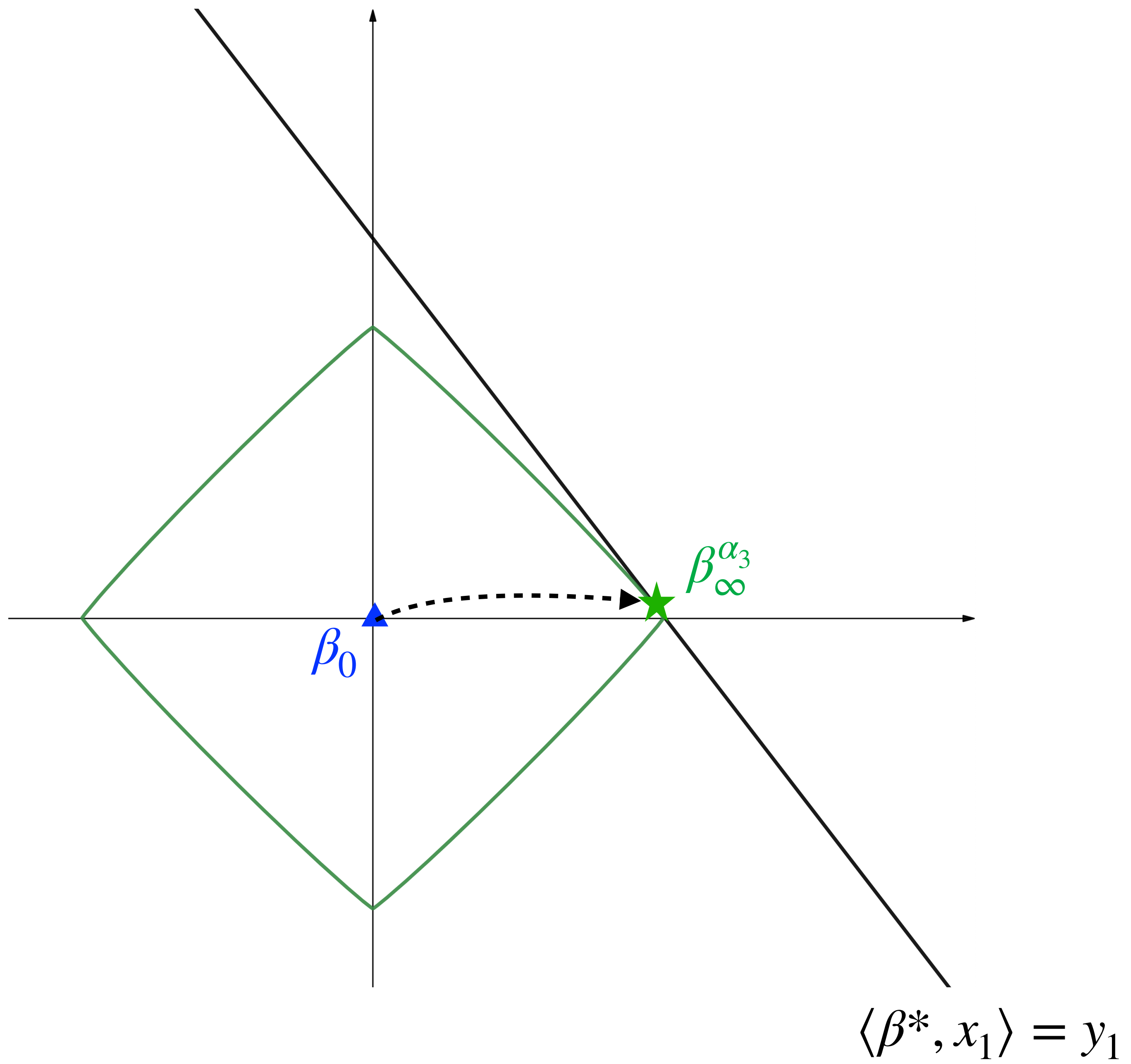
"Big" initialisation α_1



"Intermediate" initialisation α_2



"Small" initialisation α_3



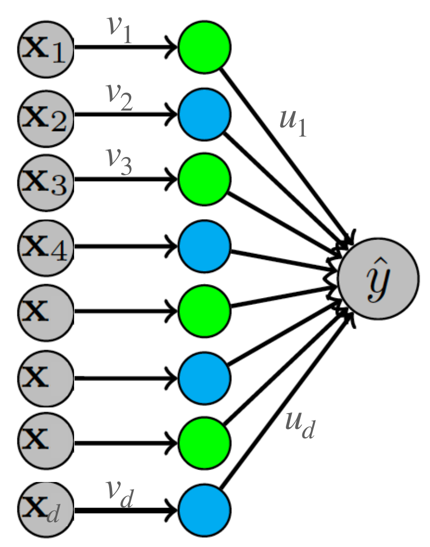
Numerical illustration:

Sparse overparametrised regression with

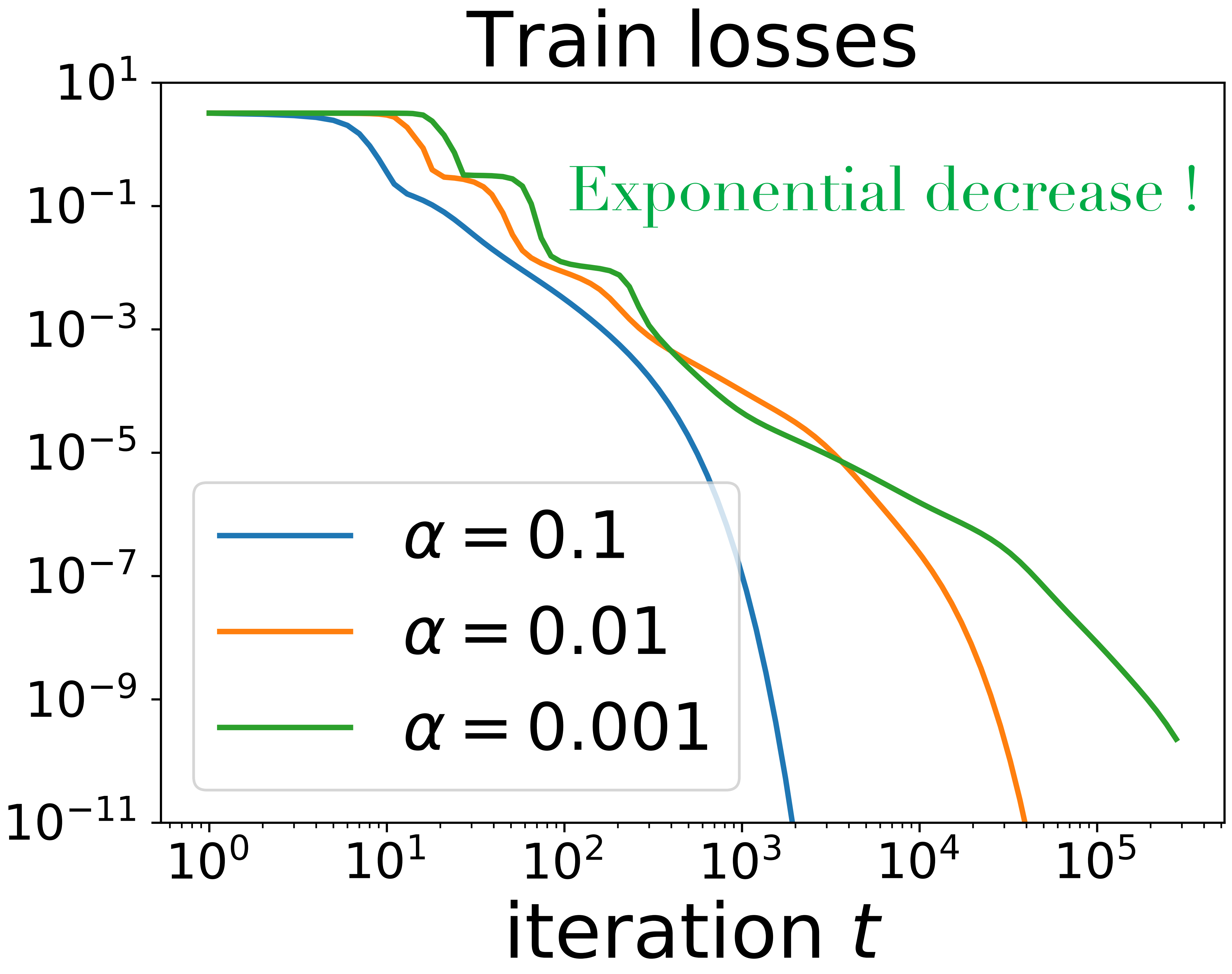
$x_i \sim \mathcal{N}(0, I_d) \quad y_i = \langle x_i, \beta^*_{\ell_0} \rangle \quad \|\beta^*_{\ell_0}\|_0 = 5$

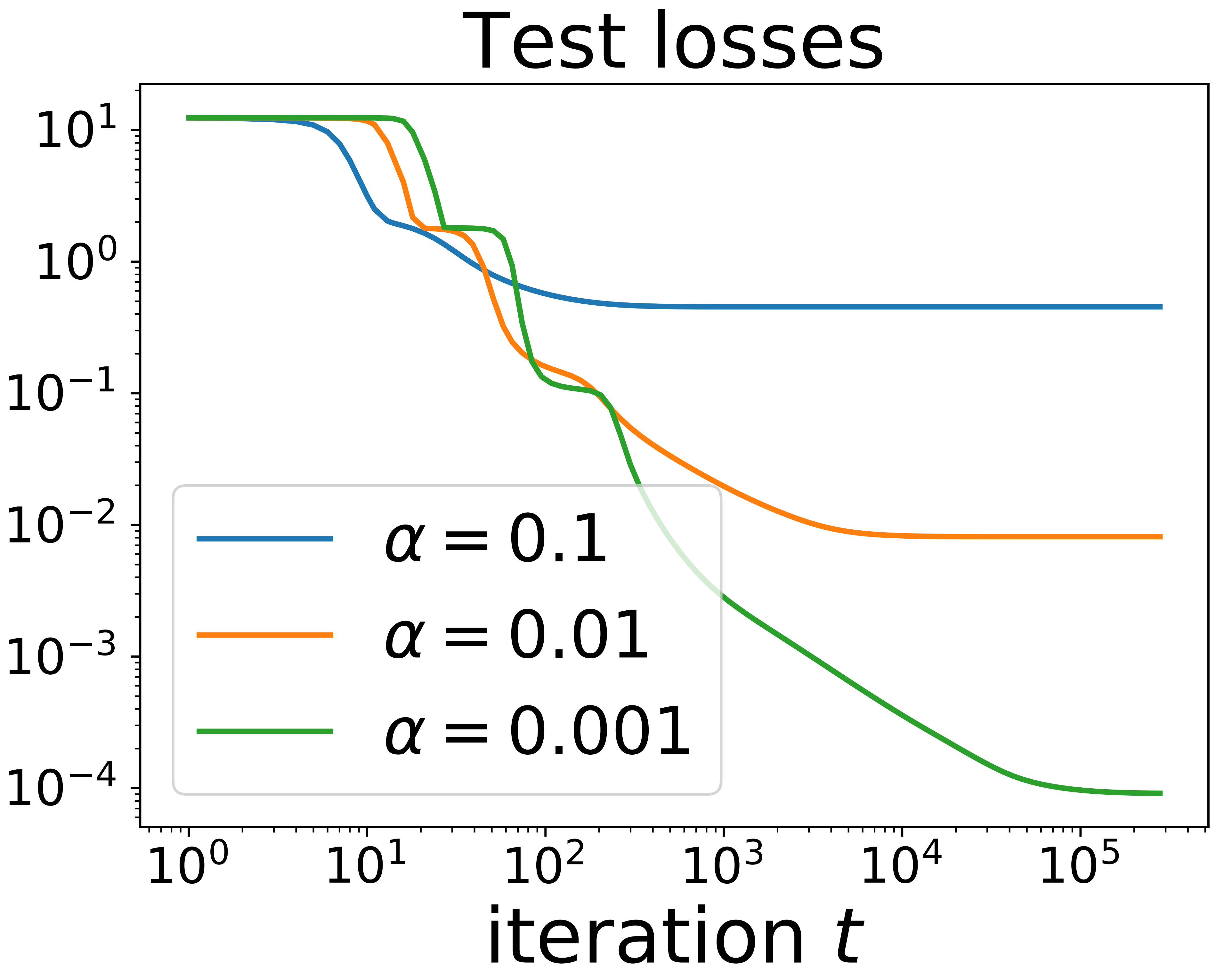
$n = 40 \quad d = 100 \quad d \gg n$

$$\min_{w \in \mathbb{R}^{2d}} L(w) = \frac{1}{4n} \sum_{i=1}^{n} (y_i - \langle \underbrace{u \odot v}_{\beta_w}, x_i \rangle)^2$$



Gradient descent with **fixed** step-size and $u_{t=0} = \alpha \mathbf{1} \in \mathbb{R}^d$, $v_{t=0} = \mathbf{0} \in \mathbb{R}^d$:





$$\beta^\alpha_\infty = \underset{\beta,\ \langle \beta, x_i \rangle = y_i}{\arg\min}\ \phi_\alpha(\beta)$$

Initialisation gets smaller

$$\beta^\alpha_\infty \xrightarrow[\alpha \to 0]{} \underset{\langle x_i, \beta \rangle = y_i}{\arg\min}\ \|\beta\|_1$$

$(= \beta^*_{\ell_0}$ *due to* ℓ_1 *magic !*$)$

Can we intuitively understand where the ℓ_1 norm comes from ?

$$L(w) = \frac{1}{4n} \sum_{i=1}^{n} (y_i - \langle u \odot v,\, x_i \rangle)^2$$

$$w = \binom{u}{v} \in \mathbb{R}^{2d} \quad \beta_w := u \odot v \in \mathbb{R}^d$$

Recall that we do **gradient flow on the neurons $w = (u, v) \in \mathbb{R}^{2d}$**.

This leads to a **mirror descent on the prediction parameter $\beta \in \mathbb{R}^d$** .

As in the linear case, we "would expect" that $w_\infty^{\alpha \to 0}$ is the solution of $\min_{w \in \mathbb{R}^{2d}, \langle u \odot v, x_i \rangle = y_i} \|w\|_2^2$

And:

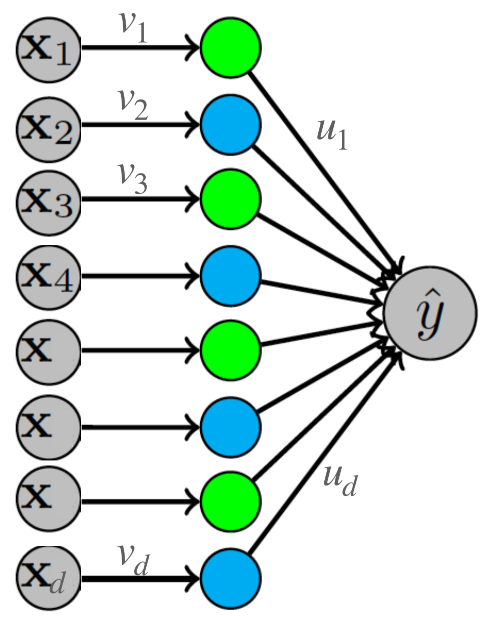
$$\min_{w \in \mathbb{R}^{2d}, \langle u \odot v, x_i \rangle = y_i} \|w\|_2^2 = \min_{(u,v), \langle u \odot v, x_i \rangle = y_i} \|u\|_2^2 + \|v\|_2^2$$
$$= 2 \min_{\beta \in \mathbb{R}^d, \langle \beta, x_i \rangle = y_i} \|\beta\|_1$$

However:

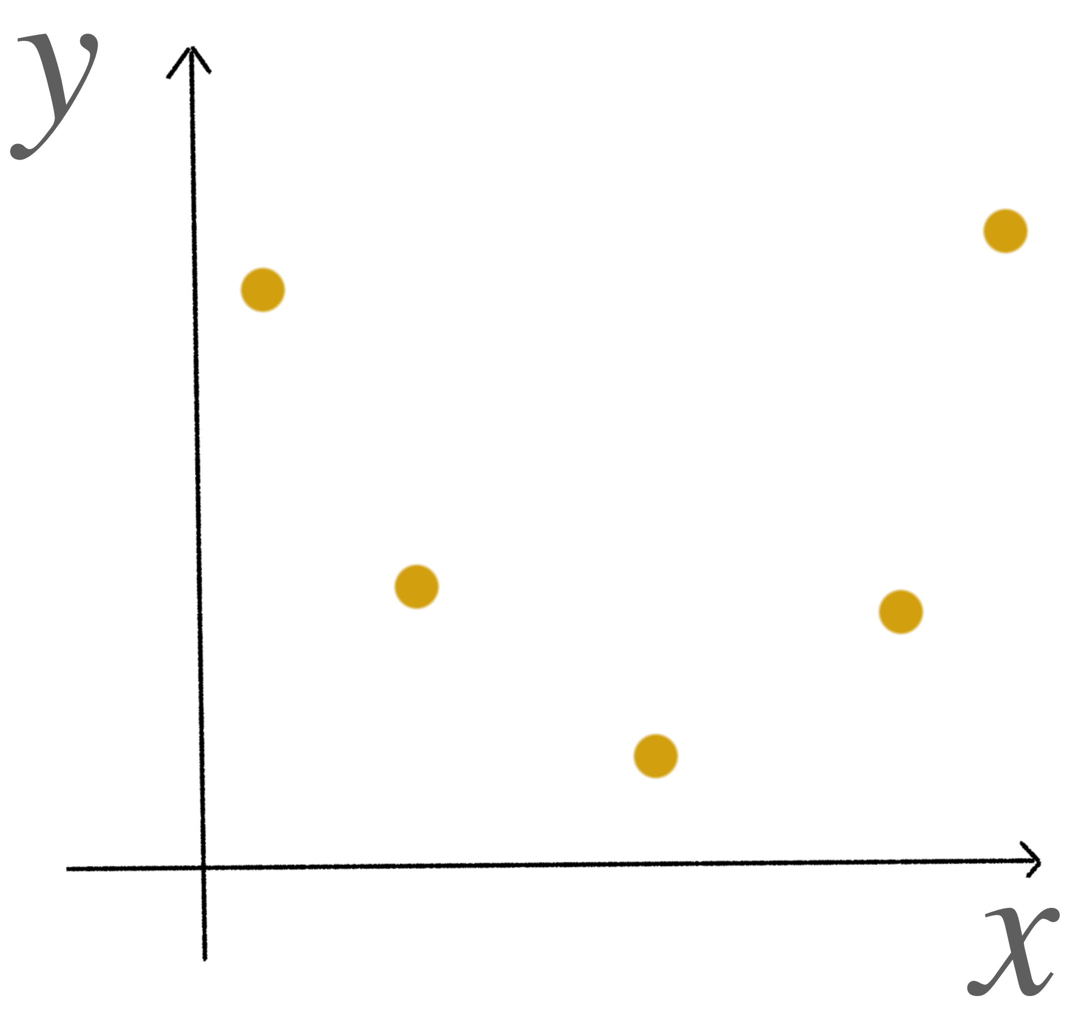
$$\min_{(u,v) \in \mathbb{R}^{2d}, \langle u \odot v, x_i \rangle = y_i} \|u - \alpha \mathbf{1}\|_2^2 + \|v\|_2^2 \neq \min_{\beta \in \mathbb{R}^d, \langle \beta, x_i \rangle = y_i} \phi_\alpha(\beta) \ !$$

Recap.

Training architecture



Training dataset



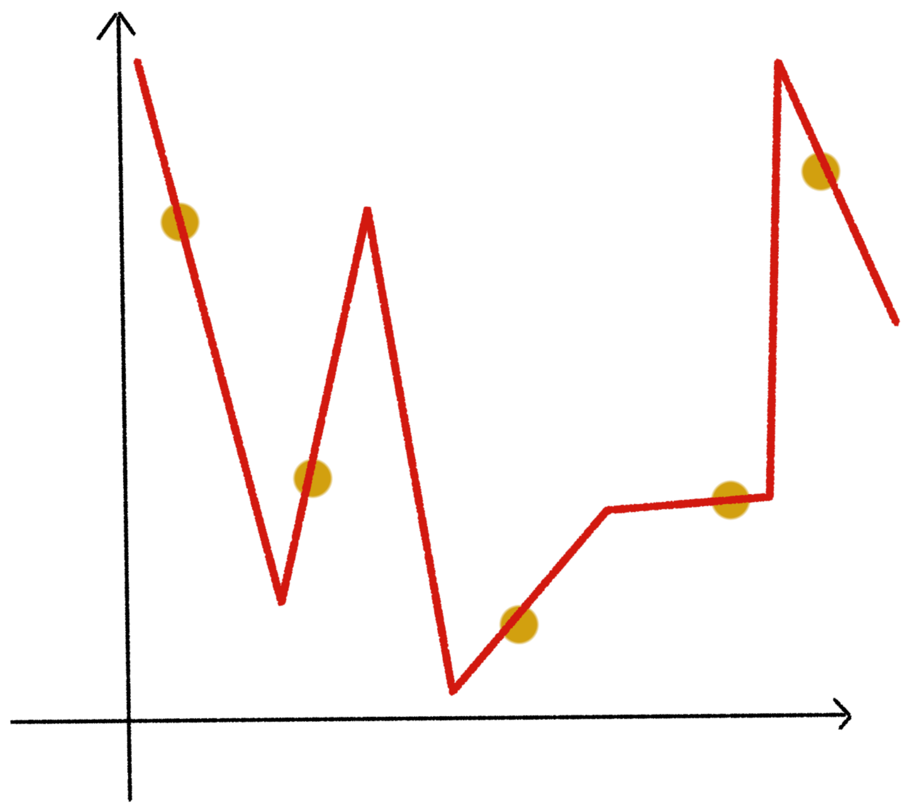
Training algorithm

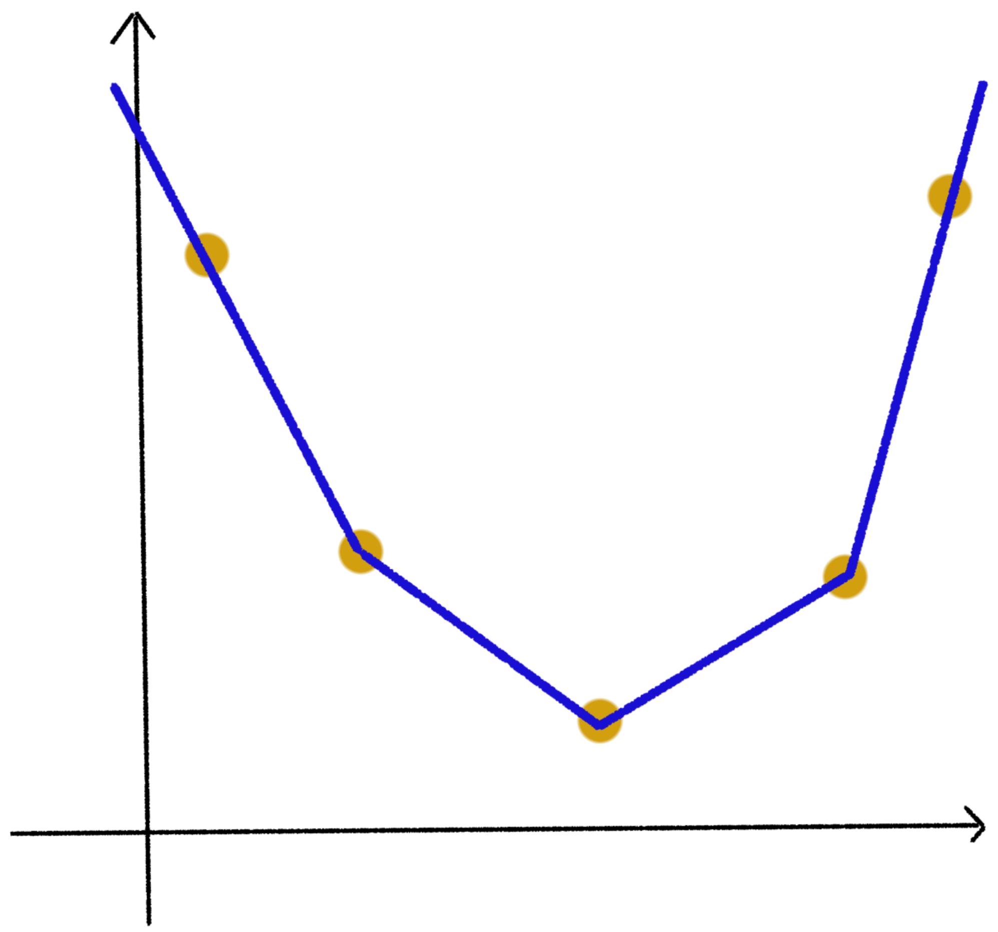
Some ERM

GF

Initialisation scale α

An infinity of interpolating solutions





$$\beta_\infty^\alpha = \arg\min_{\beta,\ \langle \beta, x_i\rangle = y_i} \ \phi_\alpha(\beta)$$

where $\phi_\alpha \underset{\alpha\to\infty}{\sim} \|\cdot\|_2$ and $\phi_\alpha \underset{\alpha\to 0}{\sim} \|\cdot\|_1$

Second part of the talk, lets talk noise.

Main question : is there a difference of implicit bias between SGD and GD ?

- We already saw that it is **not the case** when training linear models
- What about non-linear ? Neural-networks ?

Some empirical evidence that SGD often outputs models which generalise better than GD:

Efficient BackProp 1998

Yann LeCun[1], Leon Bottou[1], Genevieve B. Orr[2], and Klaus-Robert Müller[3]

Advantages of Stochastic Learning
1. Stochastic learning is usually *much* faster than batch learning.
2. Stochastic learning also often results in better solutions.
3. Stochastic learning can be used for tracking changes.

On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima

Keskar et al. 2017

Model Name	Testing Accuracy SB (Small batch)	LB (Large batch)
F_1	$98.03\% \pm 0.07\%$	$97.81\% \pm 0.07\%$
F_2	$64.02\% \pm 0.2\%$	$59.45\% \pm 1.05\%$
C_1	$80.04\% \pm 0.12\%$	$77.26\% \pm 0.42\%$
C_2	$89.24\% \pm 0.12\%$	$87.26\% \pm 0.07\%$
C_3	$49.58\% \pm 0.39\%$	$46.45\% \pm 0.43\%$
C_4	$63.08\% \pm 0.5\%$	$57.81\% \pm 0.17\%$

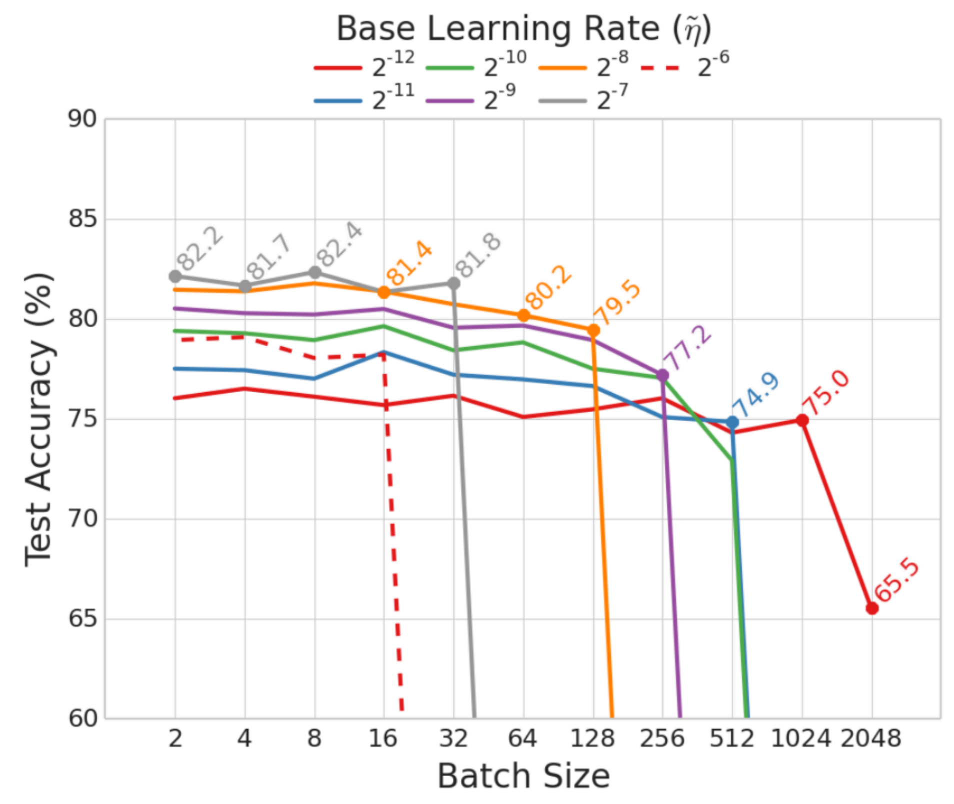
$\approx -2\,\%$

Classification task !
Train ~ 100%
For Adam !

"Experiments with other optimizers for the large-batch experiments, including SGD, led to similar results."

Revisiting Small Batch Training for Deep Neural Networks 2018

Dominic Masters and Carlo Luschi



Vanilla SGD, AlexNet, Cifar 10

What about our toy neural network ?

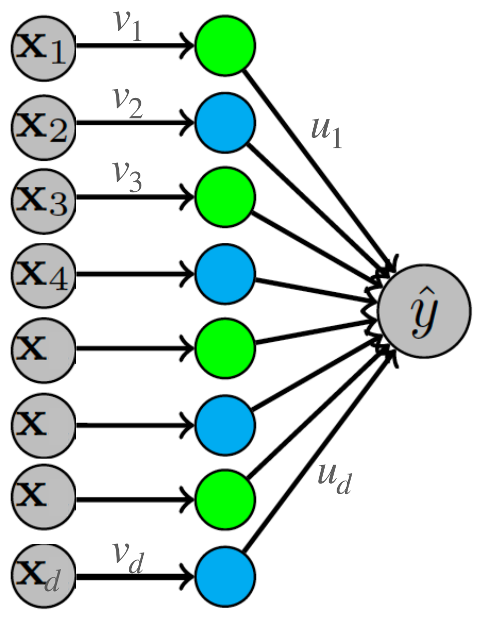
Back to our toy neural network.

Sparse overparametrised regression with

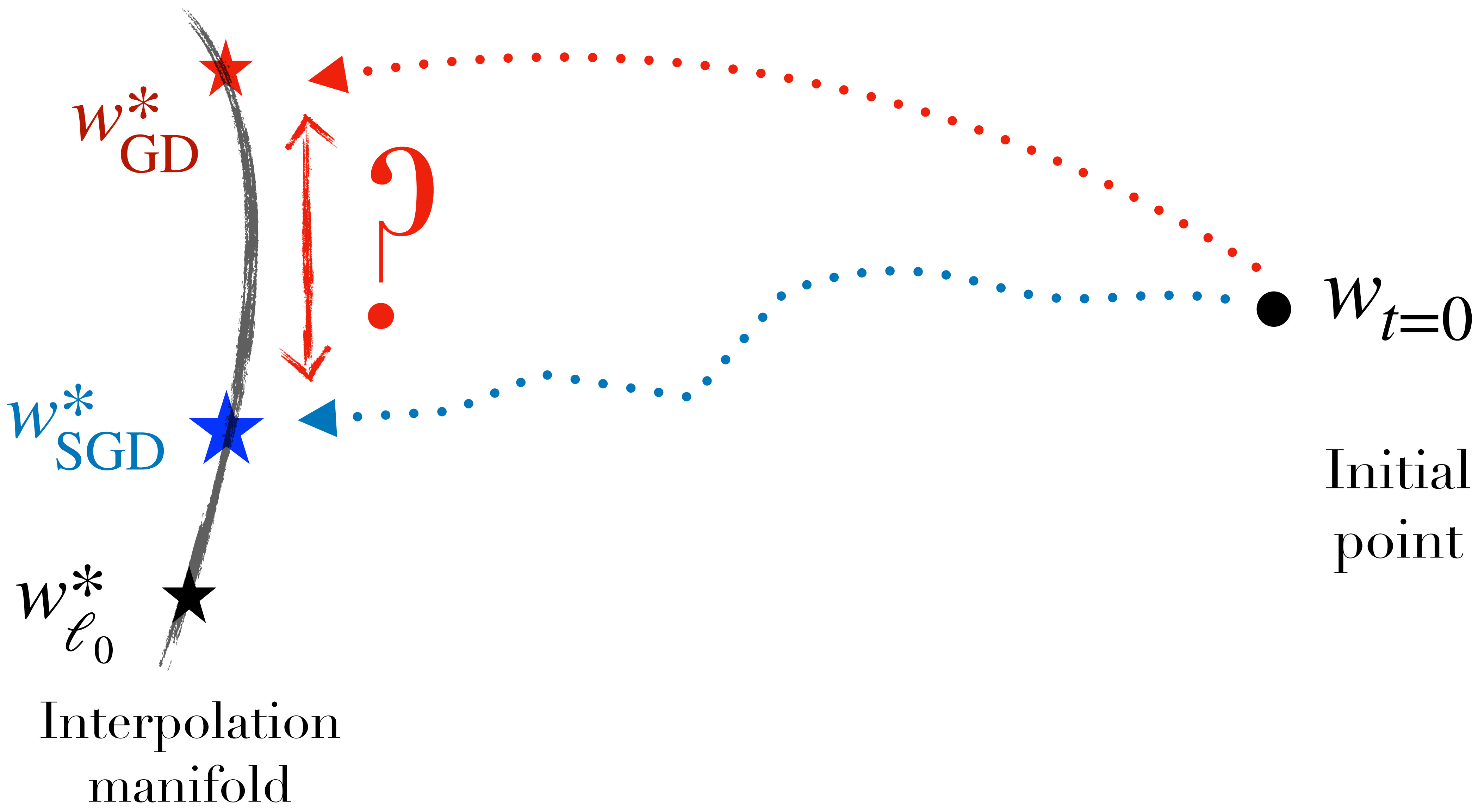
$x_i \sim \mathcal{N}(0, I_d) \qquad y_i = \langle x_i, \beta^*_{\ell_0} \rangle \qquad \|\beta^*_{\ell_0}\|_0 = 5$

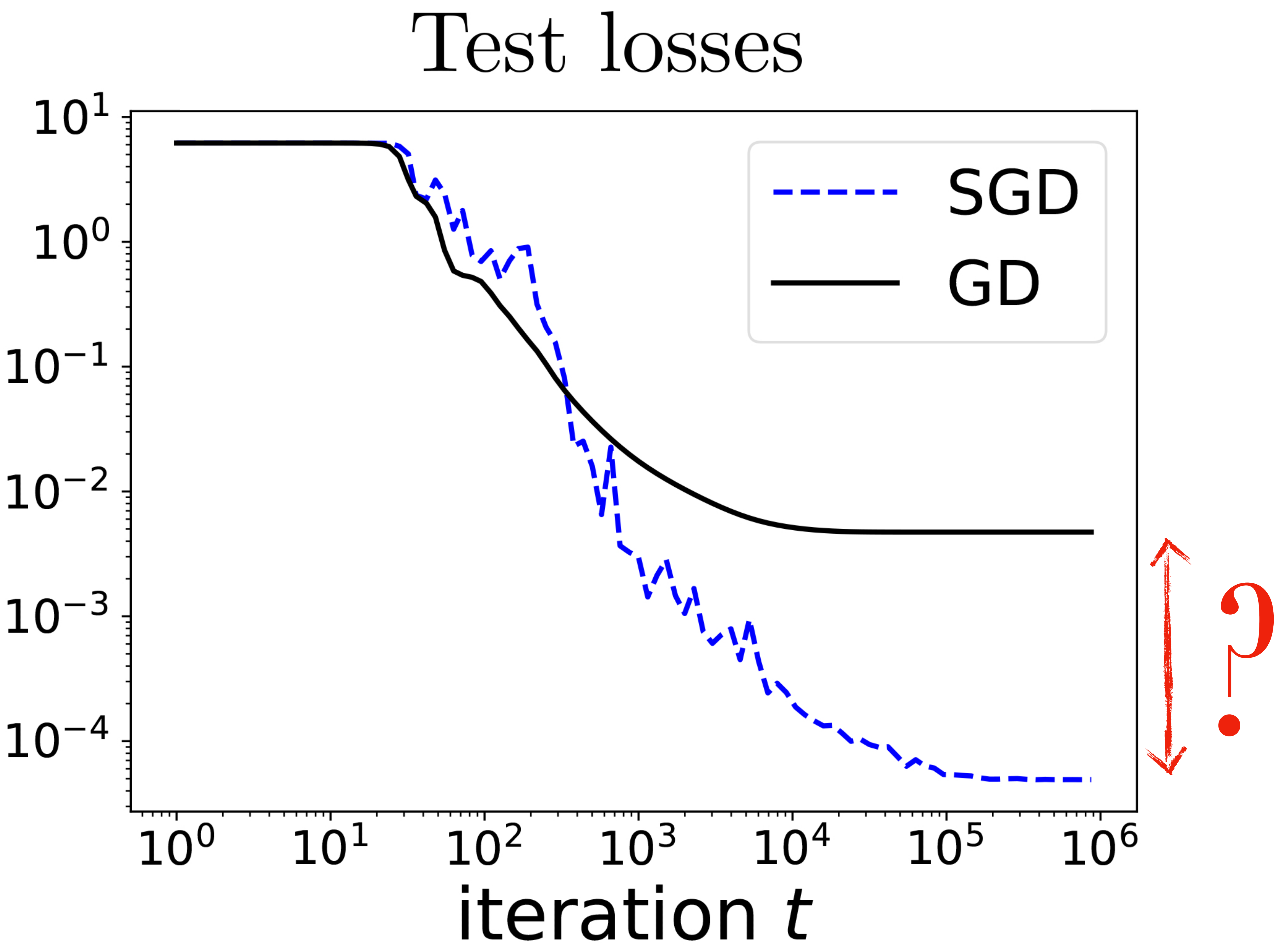
$n = 40 \qquad d = 100 \qquad d \gg n$

$$\min_{w \in \mathbb{R}^{2d}} L(w) = \frac{1}{4n} \sum_{i=1}^{n} (y_i - \langle \underbrace{u \odot v}_{\beta}, x_i \rangle)^2$$

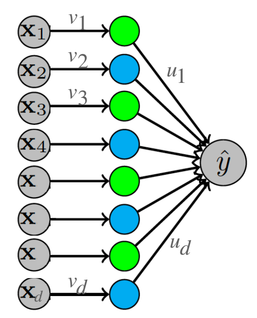


GD and SGD with **fixed** step-size and $u_{t=0} = \alpha \mathbf{1} \in \mathbb{R}^d$, $v_{t=0} = \mathbf{0} \in \mathbb{R}^d$.





SGD to Stochastic Gradient Flow



$$L(w) = \frac{1}{4n}\sum_{i=1}^{n}(y_i - \langle u \odot v, x_i\rangle)^2$$

$$w = \begin{pmatrix} u \\ v \end{pmatrix} \in \mathbb{R}^{2d} \qquad \beta_w := u \odot v \in \mathbb{R}^d$$

SGD: $w_{t+1} = w_t - \gamma \nabla L_{i_t}(w_t) \quad \Longrightarrow \quad u_{t+1} = u_t - \gamma\langle \beta_{w_t} - \beta^*, \ x_{i_t}\rangle \ x_{i_t} \odot v_t$ (Similar for v_t)

What is the "best" continuous version of this recursion ? $\mathrm{d}u_t = -\nabla_u L(w_t)\mathrm{d}t + \Sigma(w_t)\mathrm{d}B_t$

Crucial part is to correctly model the noise's structure.

Re-writing SGD: $u_{t+1} = u_t - \gamma \nabla_u L(w_t) + \underbrace{\gamma v_t \odot [X^\top \xi_{i_t}(w_t)]}$

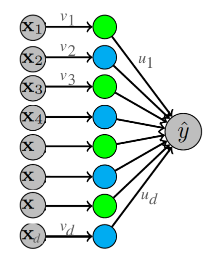
Zero mean, state dependent, vanishing, sampling noise

Two key properties of the noise: (i) belongs to $\mathrm{span}(x_1 \odot v, \ldots, x_n \odot v)$

(ii) has covariance $\Sigma_{SGD}(w) := \gamma^2 \mathrm{diag}(v) X^\top \mathrm{Cov}_{i_t}(\xi_{i_t}(\beta)) X \mathrm{diag}(v) \in \mathbb{R}^{d \times d}$

Wojtowytsch, S., Stochastic gradient descent with noise of machine learning type. Part I: Discrete time analysis.

SGD to Stochastic Gradient Flow



$L(w) = \frac{1}{4n}\sum_{i=1}^{n}(y_i - \langle u \odot v, x_i\rangle)^2$

$w = \binom{u}{v} \in \mathbb{R}^{2d}$ $\beta_w := u \odot v \in \mathbb{R}^d$

What is the "best" continuous version of the SGD recursion ? $\mathrm{d}u_t = -\nabla_u L(w_t)\mathrm{d}t + \Sigma(w_t)\mathrm{d}B_t$

Crucial part is to correctly model the noise's structure.

Re-writing SGD: $u_{t+1} = u_t - \gamma\nabla_u L(w_t) + \gamma v_t \odot [X^\top \underbrace{\xi_{i_t}(w_t)}]$

Zero mean, state dependent, vanishing, sampling noise

Two key properties of the noise: (i) belongs to $\mathrm{span}(x_1 \odot v, \dots, x_n \odot v)$

(ii) has covariance $\Sigma_{SGD}(w) := \gamma^2 \mathrm{diag}(v) X^\top \mathrm{Cov}_{i_t}(\xi_{i_t}(\beta)) X \mathrm{diag}(v) \in \mathbb{R}^{d\times d}$

We consider the following stochastic differential equation:

$$\mathrm{d}u_t = -\nabla_u L(w_t)\mathrm{d}t + 2\sqrt{\gamma n^{-1} L(w_t)}\, v_t \odot [X^\top \mathrm{d}B_t]$$

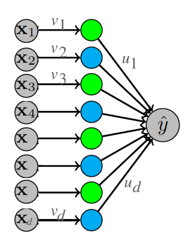
Because it conserves the two key properties: (i) structure

(ii) (nearly) matching covariance

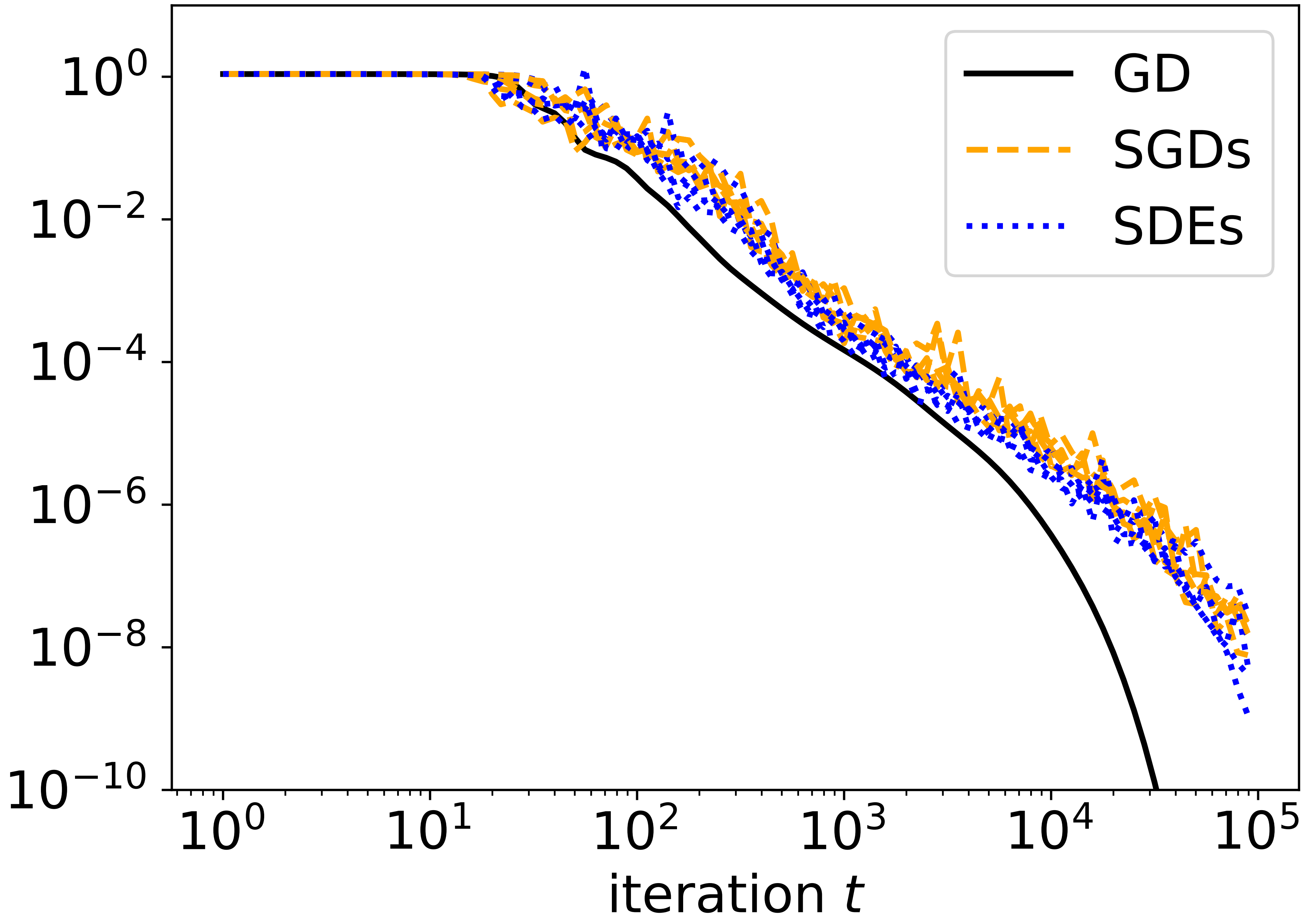
Numerical "validation"

$x_i \sim \mathcal{N}(0, I_d)$ $y_i = \langle x_i, \beta^*_{\ell_0} \rangle$ $\|\beta^*_{\ell_0}\|_0 = 5$

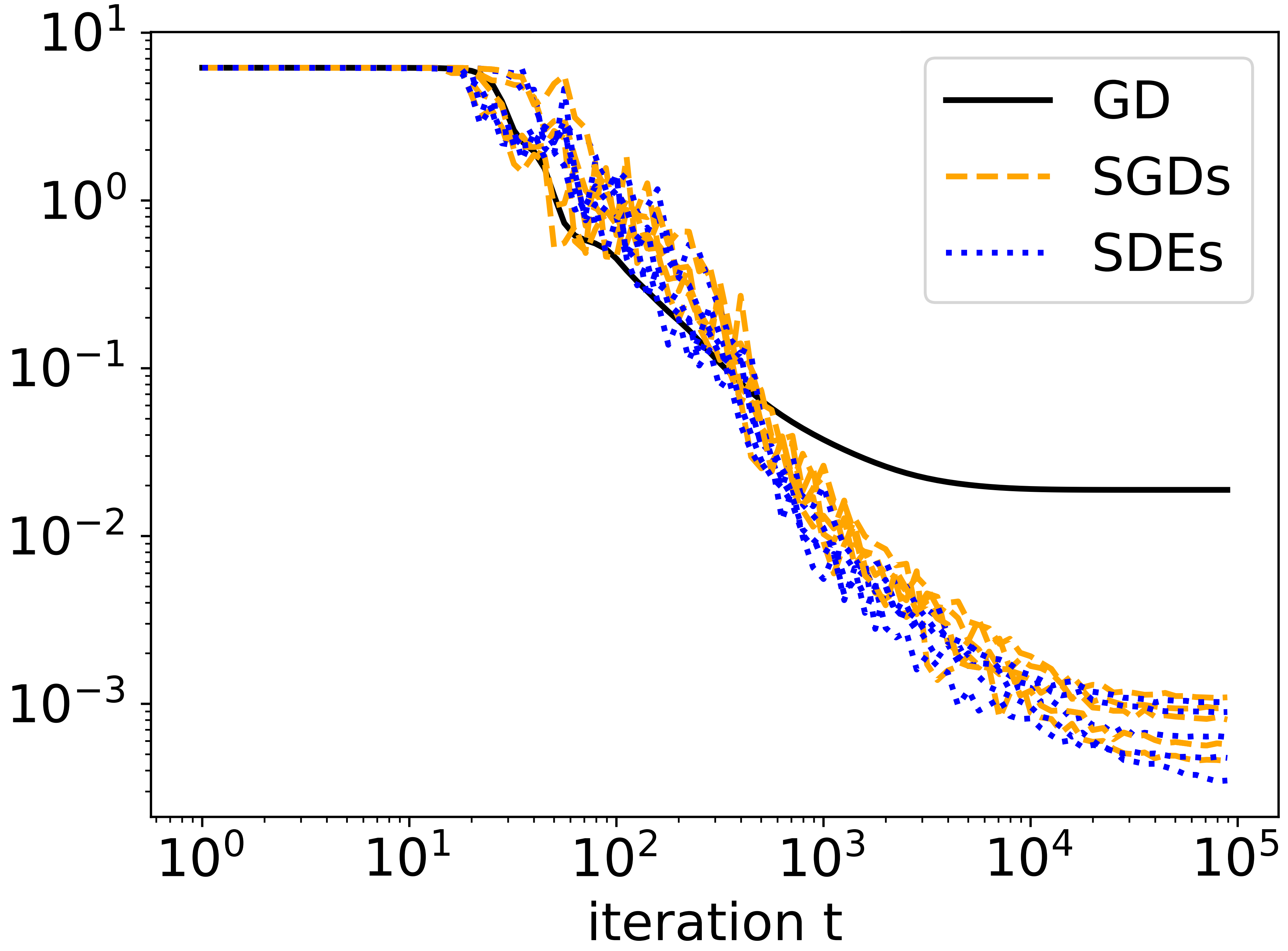
$$\min_{w \in \mathbb{R}^{2d}} L(w) = \frac{1}{4n} \sum_{i=1}^{n} (y_i - \langle u \odot v, x_i \rangle)^2$$



Train losses



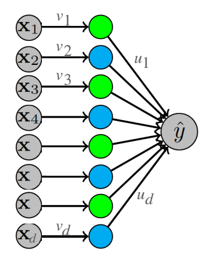
Test losses



The SDE seems to faithfully capture SGD's behaviour **for macroscopic step-sizes !**

Keep in mind:
- **This is a model !**
- There are unfortunately no theoretical guarantees for macroscopic step-sizes (**as for GF!**)
- **However it captures the key ingredients to understand the implicit bias of SGD.**

Implicit bias of the stochastic gradient flow



$L(w) = \frac{1}{4n}\sum_{i=1}^{n}(y_i - \langle u \odot v, x_i\rangle)^2$

$w = \binom{u}{v} \in \mathbb{R}^{2d}$ $\quad \beta_w := u \odot v \in \mathbb{R}^d$

Assumptions: probability $p \in (0,1)$ and initialisation $u_{t=0} = \alpha \in \mathbb{R}^d, v_{t=0} = 0$.

Step-size $\gamma \leq \tilde{O}\left(\frac{1}{\ln(4/p)\lambda_{\max}\|\beta^*_{\ell_1}\|_1}\right)$ where

$$\lambda_{\max} = \lambda_{\max}(X^\top X/n)$$

$$\beta^*_{\ell_1} = \underset{\beta \ s.t. \ X\beta = y}{\operatorname{argmin}} \|\beta\|_1$$

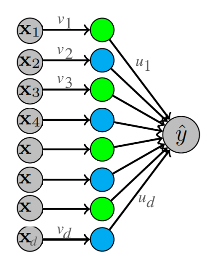
Result: With probability $1-p$, the Stochastic Gradient Flow (u_t, v_t) is such that:

Convergence $\longrightarrow$ • The flow $(\beta_t)_{t\geq 0} = (u_t \odot v_t)_{t \geq 0}$ converges towards a zero-training error solution $\beta_\infty^{\alpha,\mathrm{SGF}}$

Implicit Bias $\longrightarrow$ • This solution $\beta_\infty^{\alpha,\mathrm{SGF}}$ satisfies

$$\beta_\infty^{\alpha,\mathrm{SGF}} = \underset{\beta \in \mathbb{R}^d,\ \langle \beta, x_i\rangle = y_i}{\arg\min} \ \phi_{\alpha_\infty}(\beta) \quad \text{where} \quad \underbrace{\alpha_\infty}_{\text{“effective” initialisation}} = \alpha \odot \exp\left(-2\gamma \mathrm{diag}\left(\frac{X^\top X}{n}\right) \overbrace{\int_0^{+\infty} \underbrace{L(\beta_s)}_{\text{training loss}} \mathrm{d}s}^{\text{stochastic !}}\right) < \overbrace{\alpha}^{\text{initialisation scale}}$$

What does this mean ?



$$L(w) = \frac{1}{4n}\sum_{i=1}^{n}(y_i - \langle u \odot v, x_i\rangle)^2$$

$$w = \binom{u}{v} \in \mathbb{R}^{2d} \qquad \beta_w := u \odot v \in \mathbb{R}^d$$

$$\beta_\infty^{\alpha,\mathrm{SGF}} = \underset{\beta\in\mathbb{R}^d,\ \langle\beta,x_i\rangle=y_i}{\arg\min}\ \phi_{\alpha_\infty}(\beta) \quad \text{where} \quad \underbrace{\alpha_\infty}_{\text{“effective” initialisation}} = \alpha \odot \exp\left(-2\gamma\,\mathrm{diag}\left(\frac{X^\top X}{n}\right)\overbrace{\int_0^{+\infty}\underbrace{L(\beta_s)\mathrm{d}s}_{\text{training loss}}}^{\text{stochastic !}}\right) < \overbrace{\alpha}^{\text{initialisation scale}}$$

GF vs SGF: Recall that: $\beta_\infty^{\alpha,\mathrm{GF}} = \underset{\beta\in\mathbb{R}^d,\ \langle\beta,x_i\rangle=y_i}{\arg\min}\ \phi_\alpha(\beta)$

- Implicit bias of SGF is the same as GF but with an effective initialisation
- Since $\alpha_\infty < \alpha$, ϕ_{α_∞} is closer to the ℓ_1 norm than ϕ_α and $\boxed{\beta_\infty^{\alpha,\mathrm{SGF}} \text{ is “sparser” than } \beta_\infty^{\alpha,\mathrm{GF}}}$

The slower the convergence, the “better” the bias:

$$\int_0^{+\infty} L(\beta_s)\mathrm{d}s \gg 1 \implies \alpha_\infty \ll \alpha$$

The bigger the step-size, the “better” the bias

Convergence for fixed step-size !

What does this mean ?

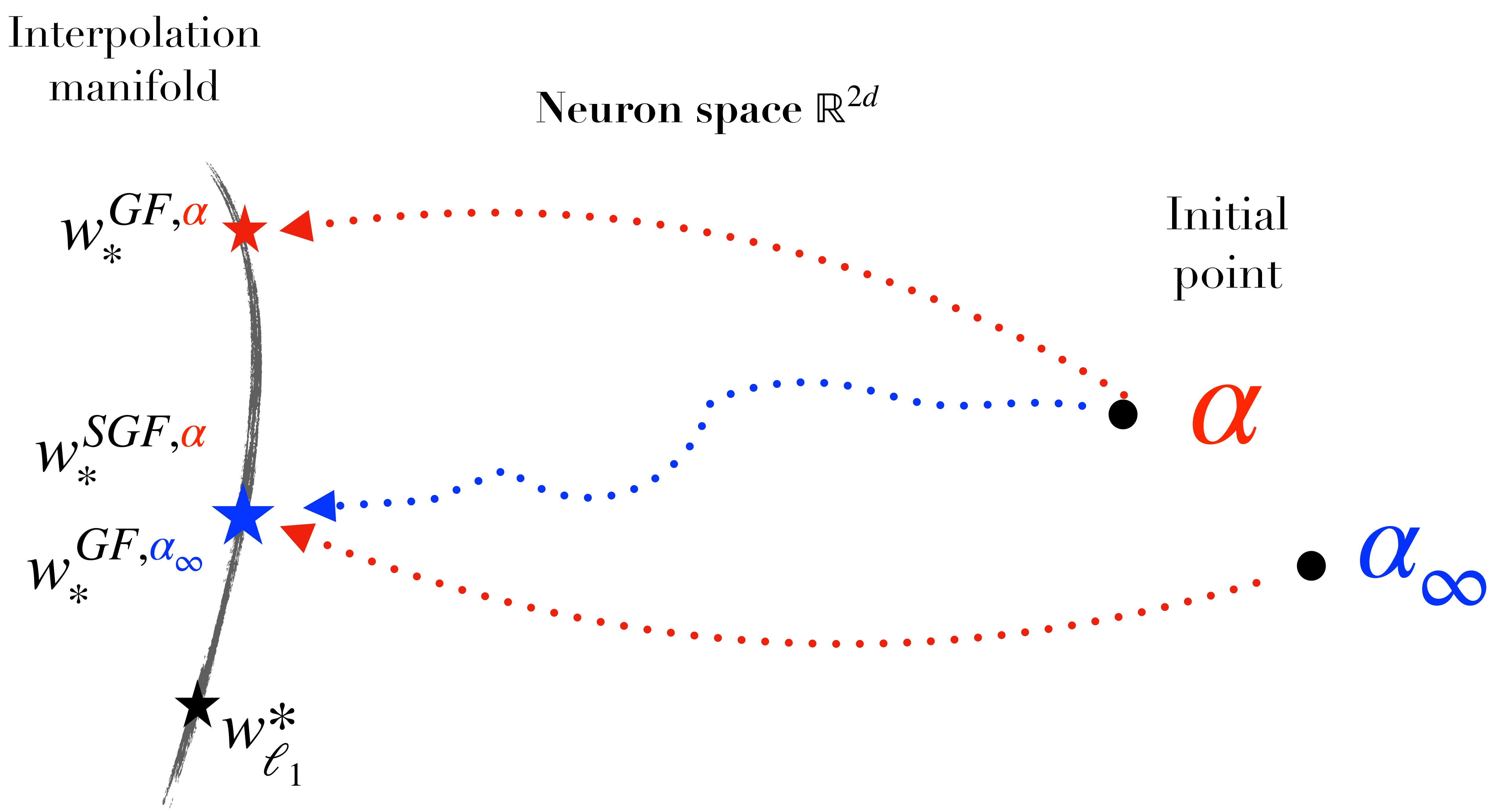
$$\beta_\infty^{\alpha,\mathrm{SGF}} = \underset{\beta\in\mathbb{R}^d,\ \langle\beta,x_i\rangle=y_i}{\arg\min} \phi_{\alpha_\infty}(\beta)$$

$$\beta_\infty^{\alpha,\mathrm{GF}} = \underset{\beta\in\mathbb{R}^d,\ \langle\beta,x_i\rangle=y_i}{\arg\min} \phi_{\alpha}(\beta)$$

$$\underbrace{\alpha_\infty}_{\text{"effective" initialisation}} = \alpha \odot \exp\left(-2\gamma \mathrm{diag}\left(\frac{X^\top X}{n}\right) \overbrace{\int_0^{+\infty} \underbrace{L(\beta_s)}_{\text{training loss}} \mathrm{d}s}^{\text{stochastic !}}\right) < \alpha$$

Neuron initialisation scale

$$\beta_\infty^{\alpha,\mathrm{SGF}} = \beta_\infty^{\alpha_\infty,\mathrm{GF}}$$

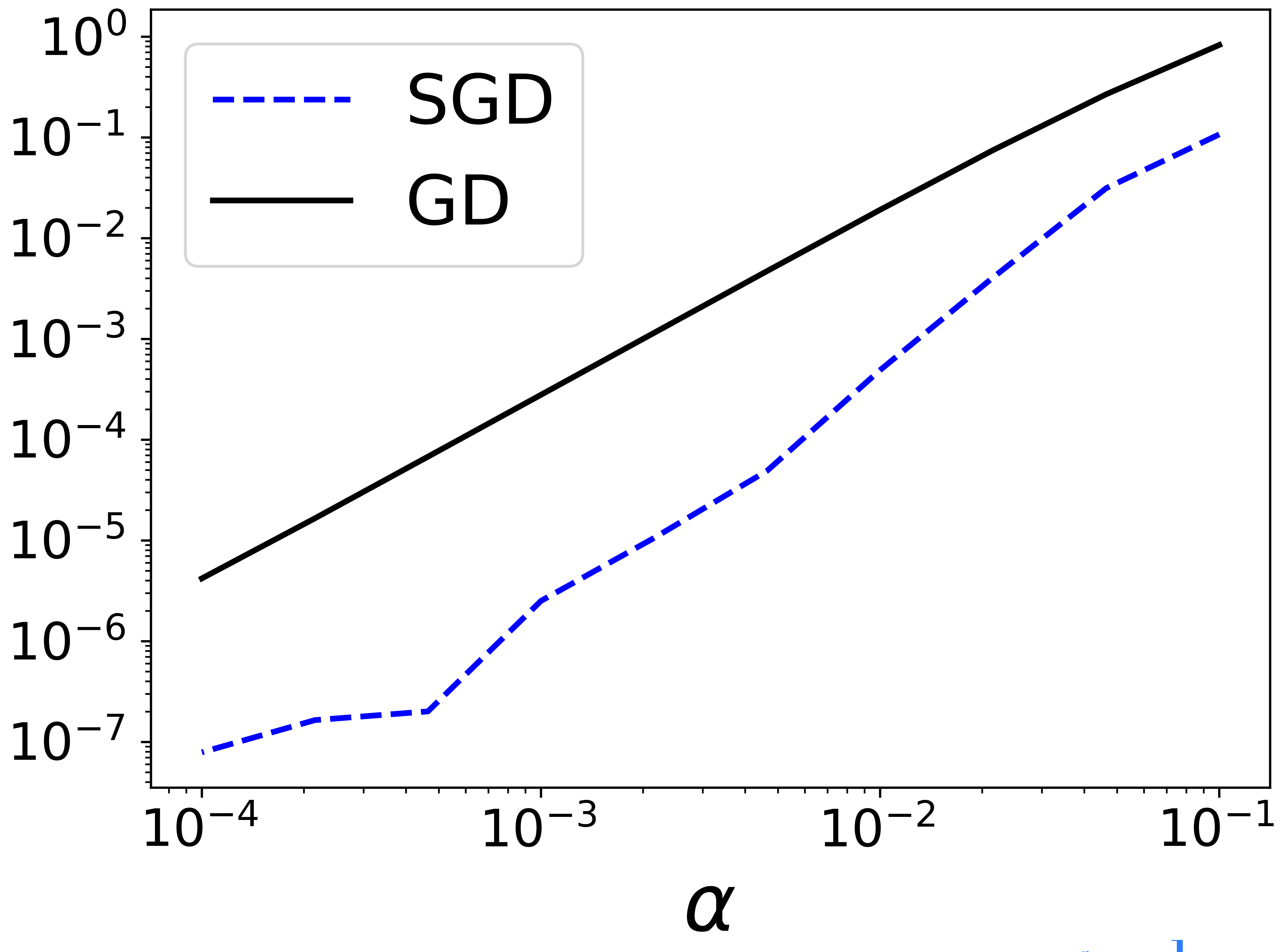


Benefit of stochasticity

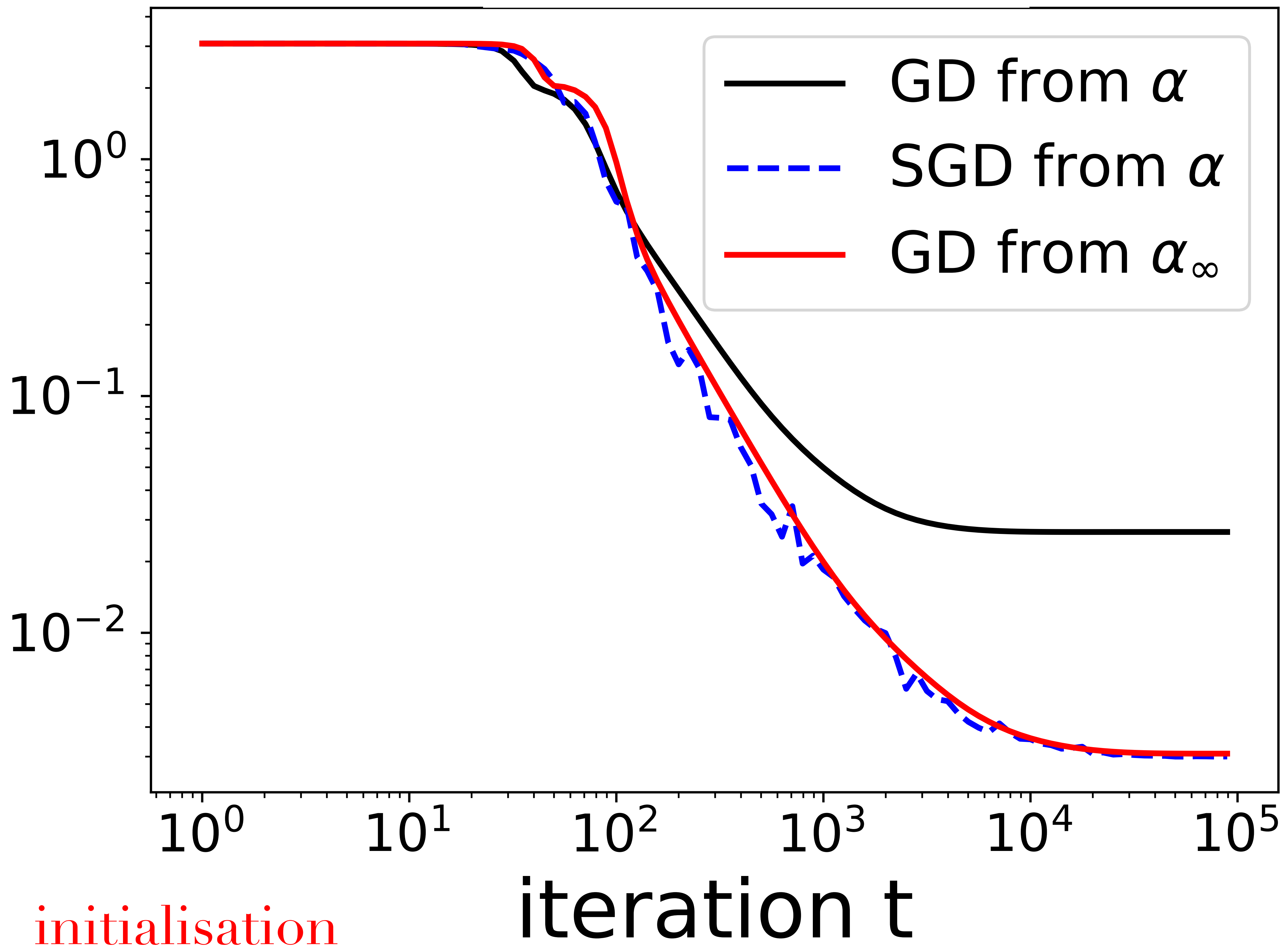
Setting: $n = 40$ $d = 100$ $\|\beta^*_{\ell_0}\|_0 = 5$

$x_i \sim \mathcal{N}(0, I)$ $y_i = \langle x_i, \beta^*_{\ell_0} \rangle$ $\alpha = 0.1$

Test losses at convergence



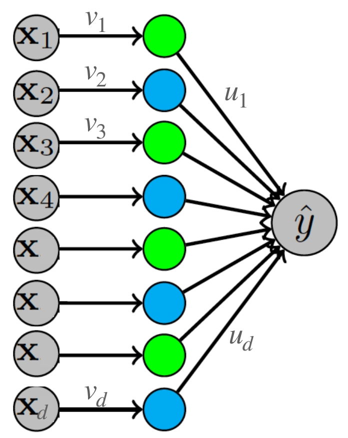
Test losses



$$\underbrace{\alpha_\infty}_{\text{“effective” initialisation}} = \alpha \odot \exp\Big(-2\gamma \mathrm{diag}\big(\tfrac{X^\top X}{n}\big) \overbrace{\int_0^{+\infty} \underbrace{L(\beta_s)}_{\text{training loss}} \mathrm{d}s}^{\text{stochastic !}}\Big) < \overbrace{\alpha}^{\text{initialisation scale}}$$

Benefit of stochasticity

Samples $(x_i, y_i)_{1 \le i \le n} \in \mathbb{R} \times \mathbb{R}$ from some distribution $\mathscr{D}$.
We want to linearly interpolate with feature expansion $\phi(x) = (\cos(2\pi i x), \sin(2\pi i x))_{1 \le i \le d/2} \in \mathbb{R}^d$.



$f_w(x) = \langle u \odot v, \phi(x) \rangle$

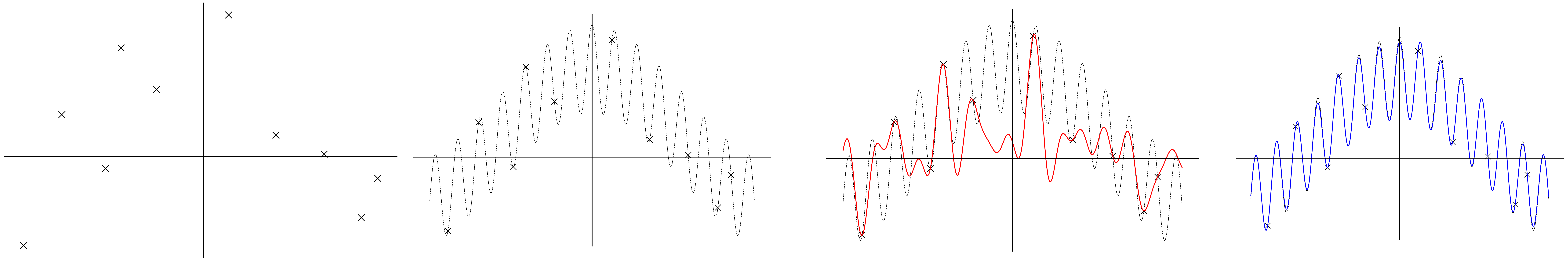
Training set 1

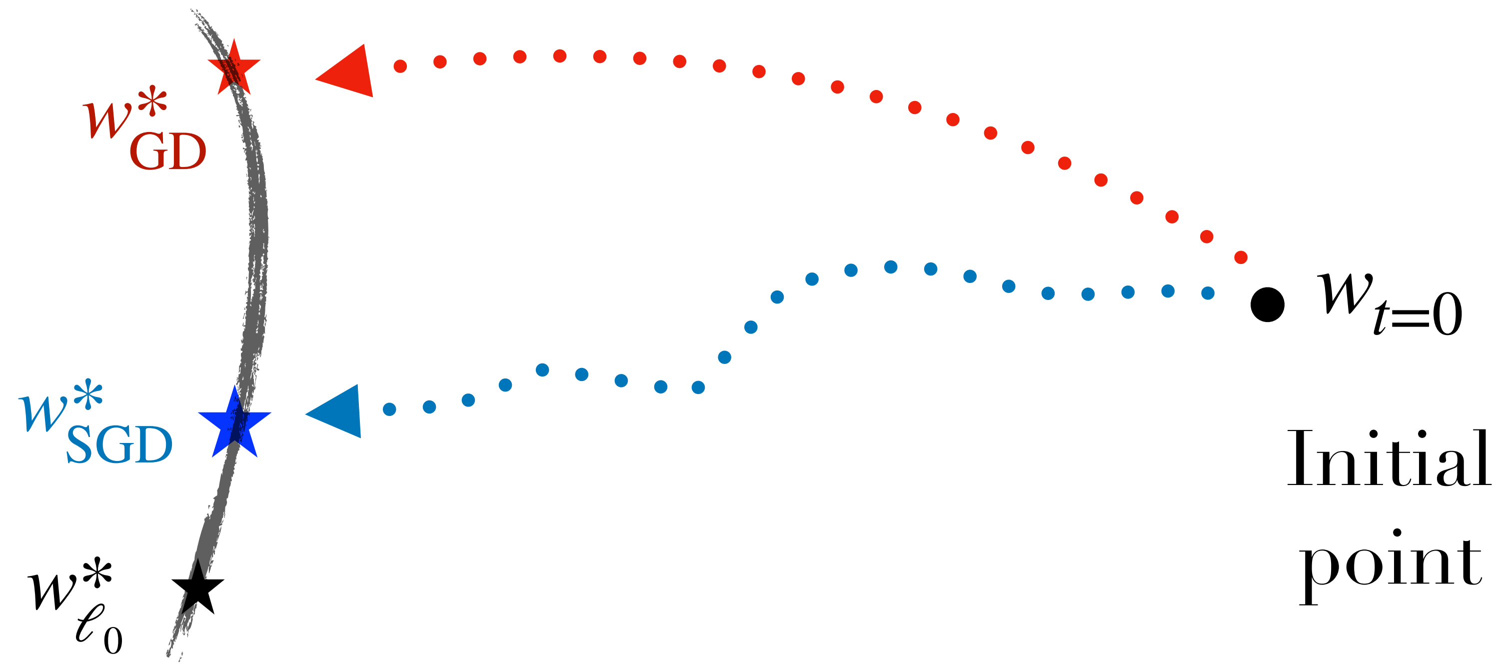
From sparse distribution

GD from initialisation α

SGD from initialisation α

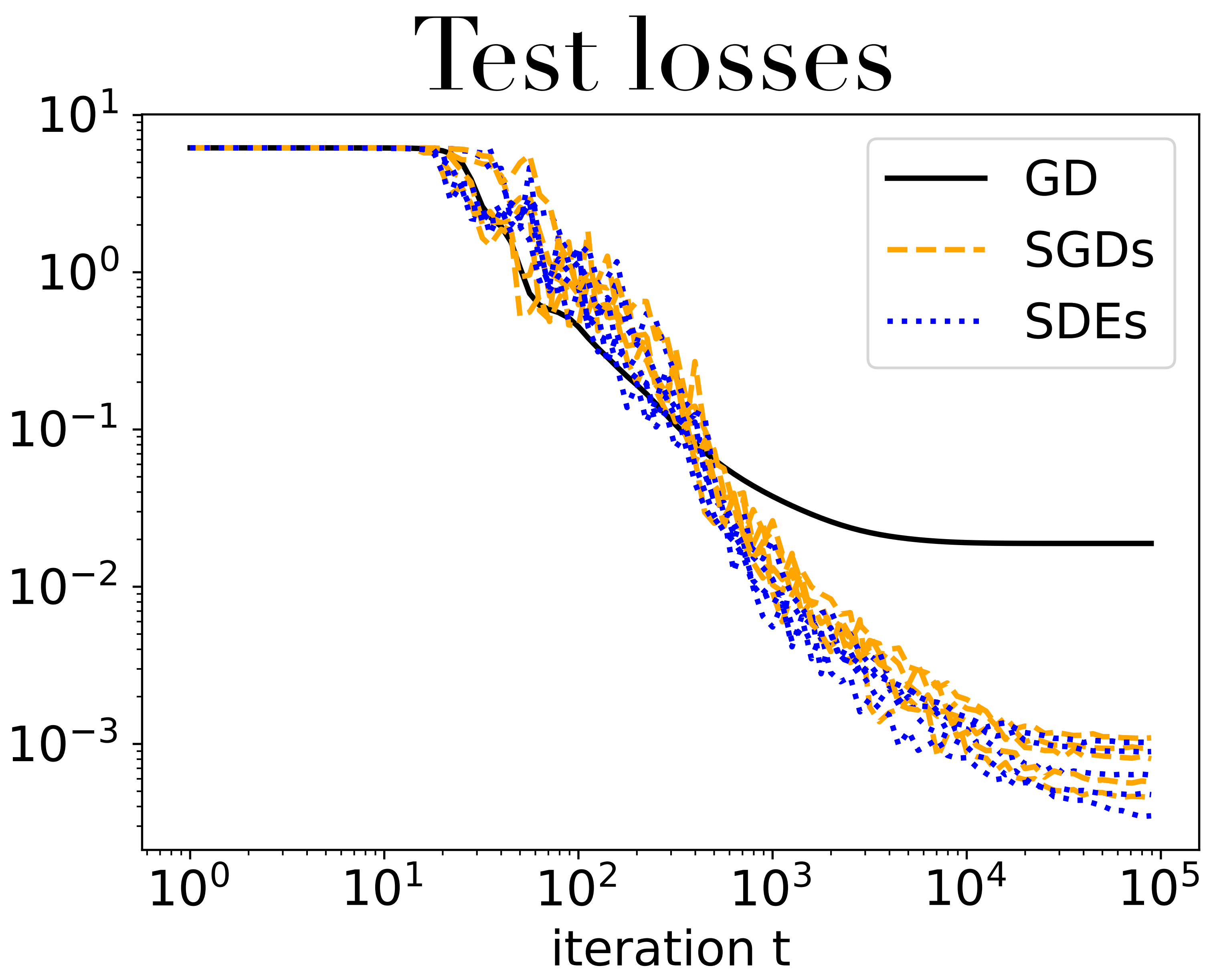
(+label noise)



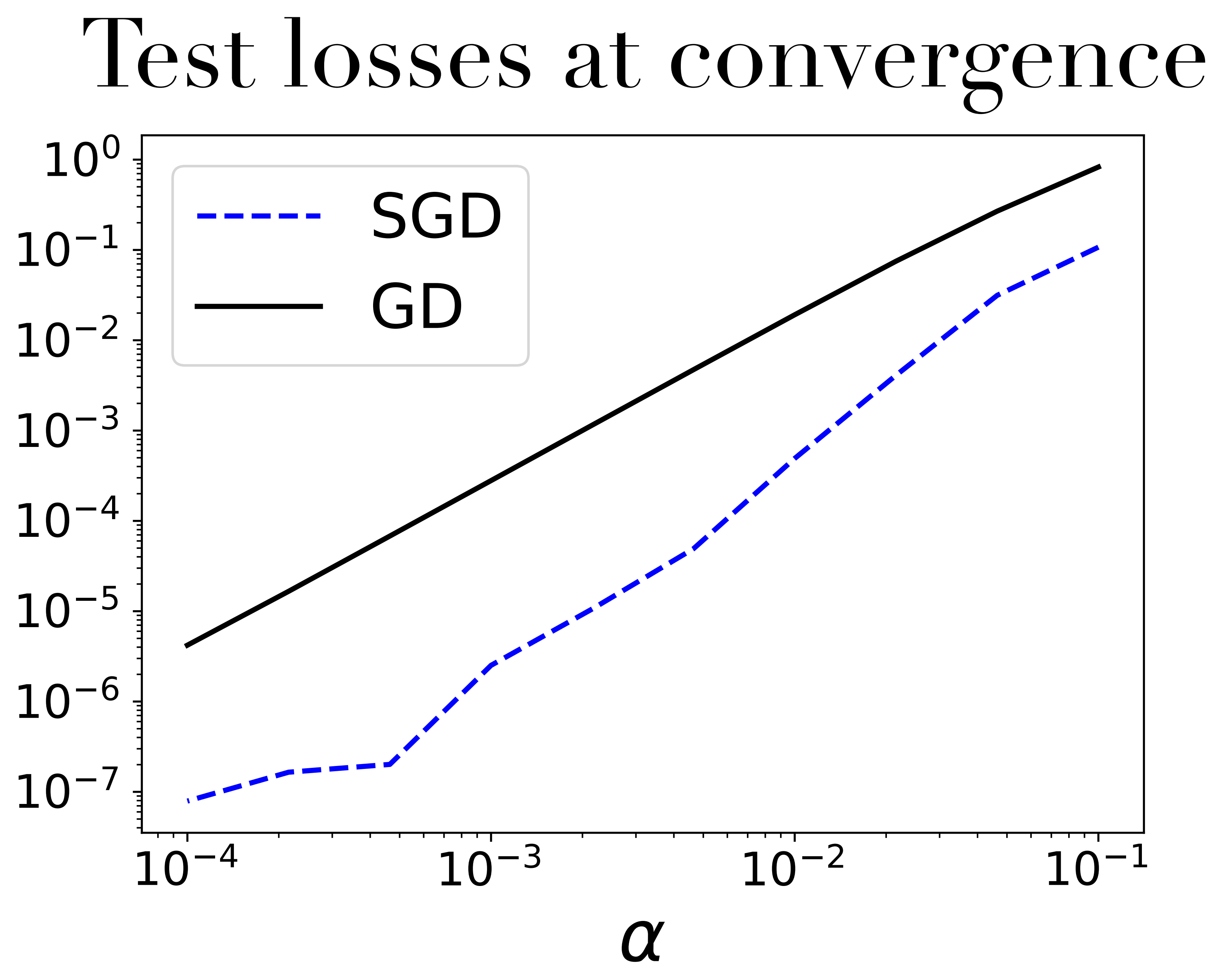


Take home messages

1. Considering (appropriate) stochastic gradient flows can lead to interesting and pertinent results



2. For a very specific toy problem, the noise inherent to SGD's stochasticity helps recover a solution which has better sparsity properties than that of GD.



Implicit regularisation of gradient algorithms

Scott Pesme

Loucas Pillaud-Vivien

Nicolas Flammarion

TML lab

EPFL

Bonus 0: other SDE modelisations

SGD: $u_{t+1} = u_t - \gamma \nabla_u L(w_t) + \gamma v_t \odot [X^\top \xi_{i_t}(w_t)]$

Our SDE:

$$\mathrm{d}u_t = -\nabla_u L(w_t)\mathrm{d}t + 2\sqrt{\gamma n^{-1} L(w_t)}v_t \odot [X^\top \mathrm{d}B_t]$$

Overdamped Langevin:

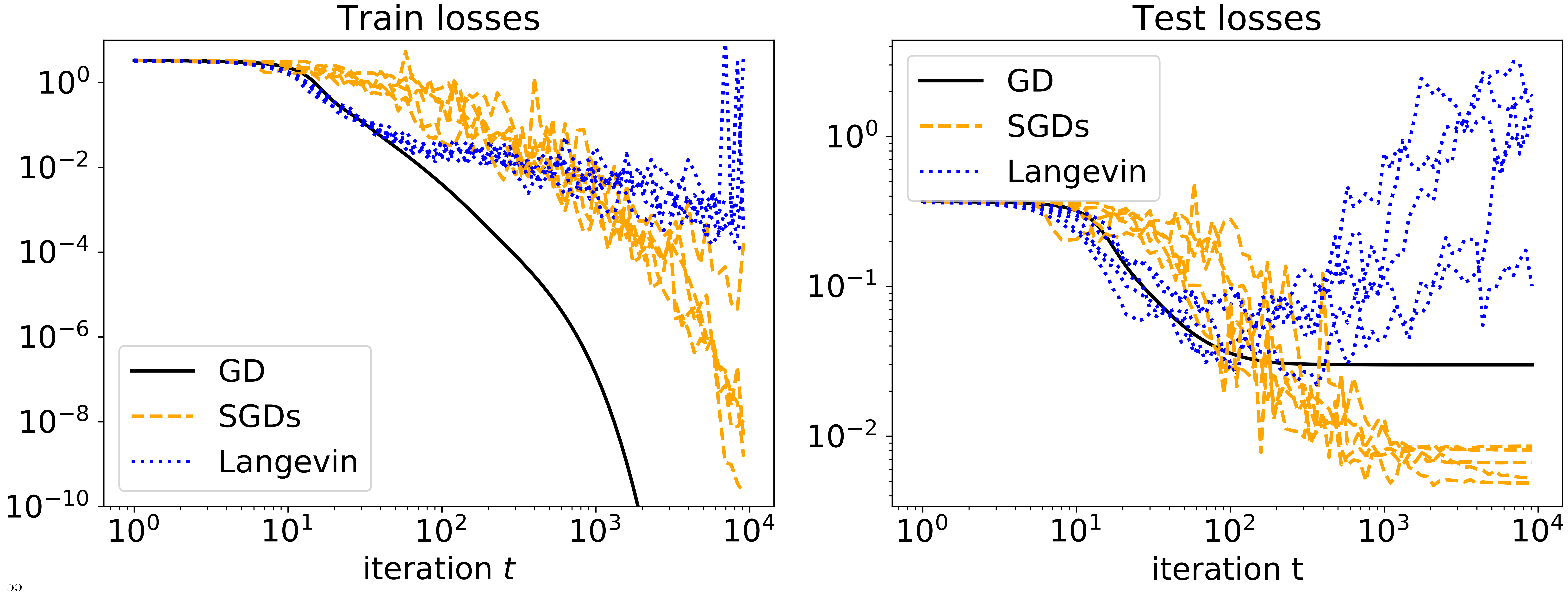
$$\mathrm{d}u_t = -\nabla_u L(w_t)\mathrm{d}t + \sqrt{2\eta^{-1}}\mathrm{d}\tilde{B}_t$$

"Wrong SDE":

$$\mathrm{d}u_t = -\nabla_u L(w_t)\mathrm{d}t + 2\sqrt{\gamma n^{-1} L(w_t)}v_t \odot \mathrm{d}\tilde{B}_t$$

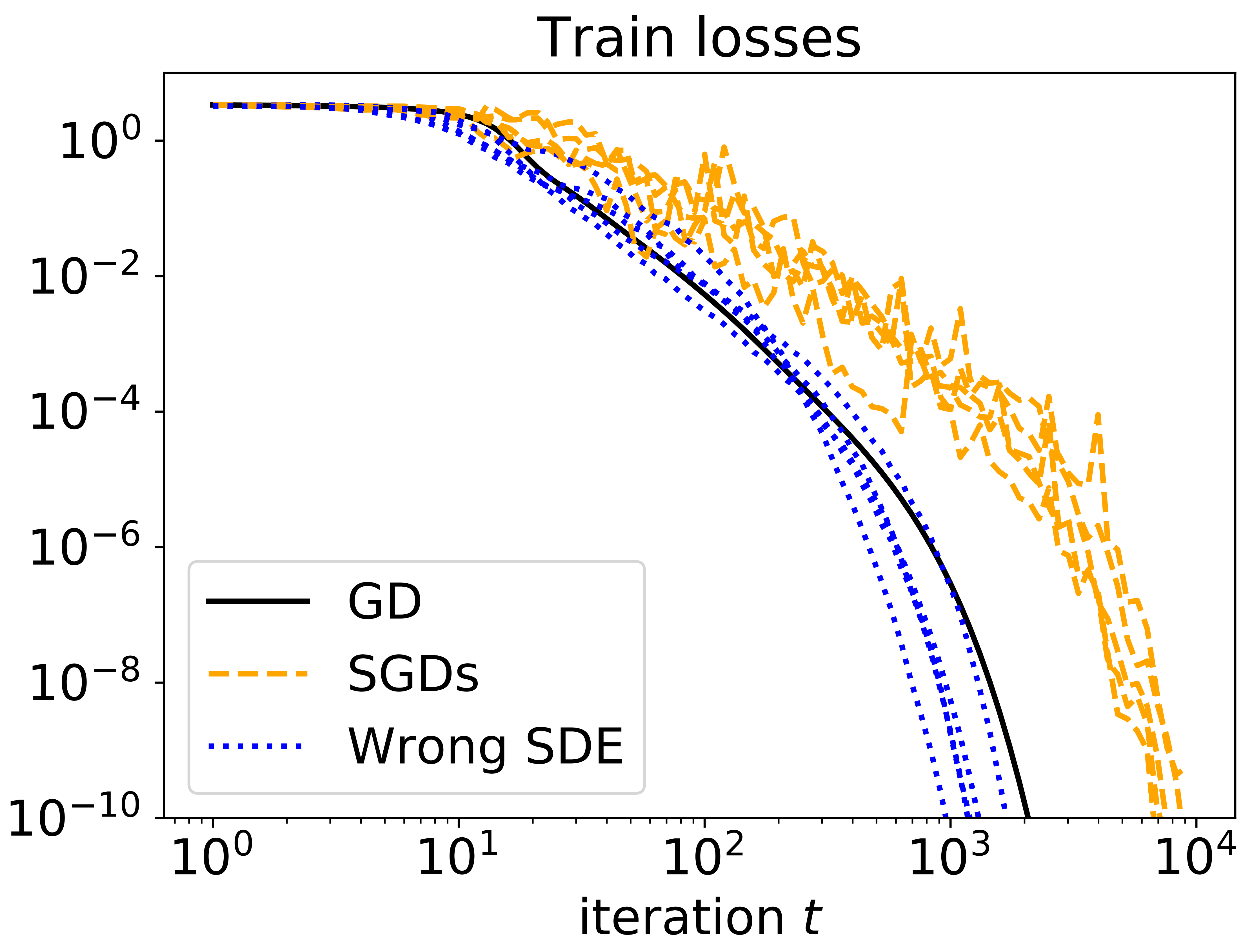
Bonus 0: other SDE modelisations

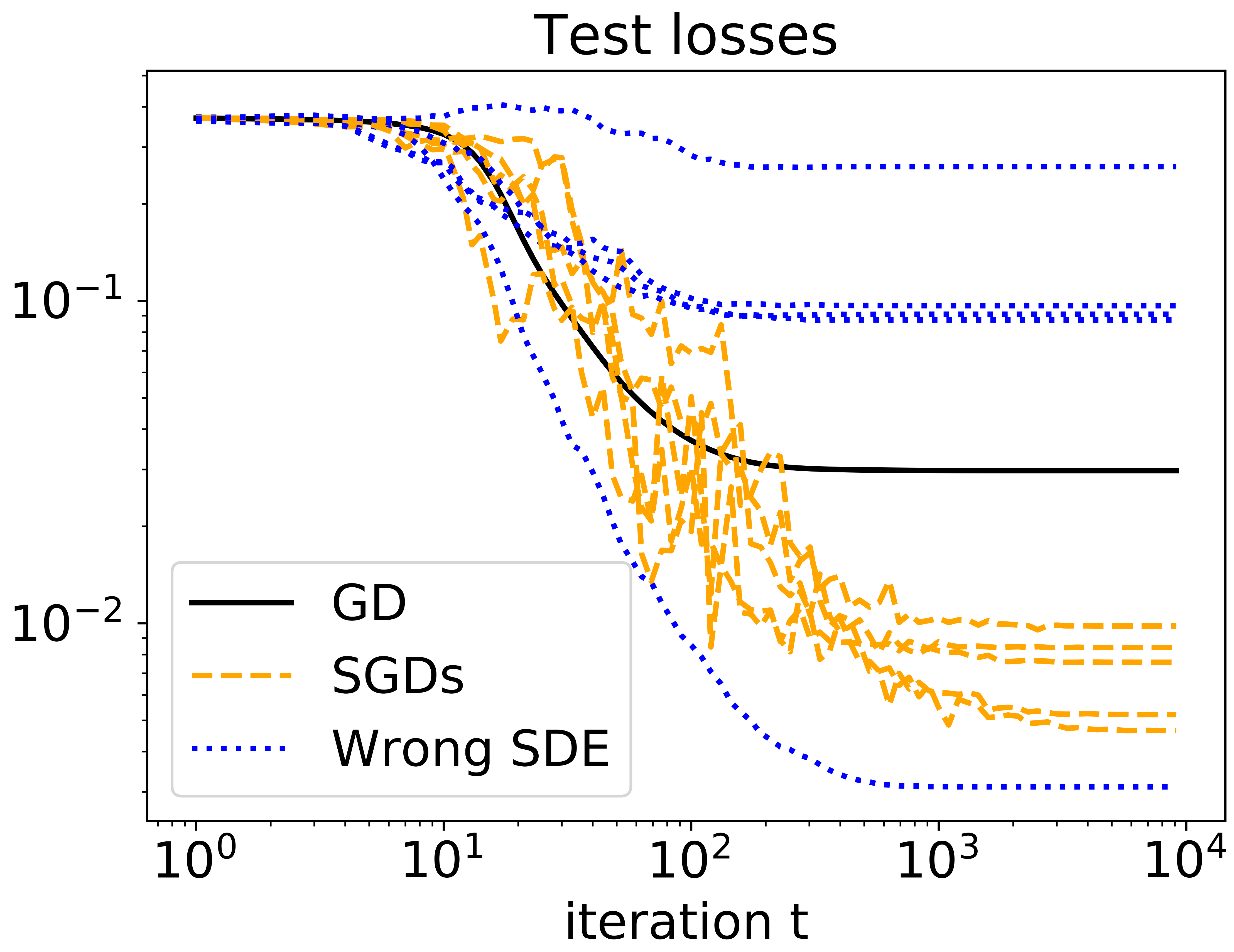
Overdamped Langevin: $$\mathrm{d}u_t = -\nabla_u L(w_t)\mathrm{d}t + \sqrt{2\eta^{-1}}\mathrm{d}\tilde{B}_t$$



Bonus 0: other SDE modelisations

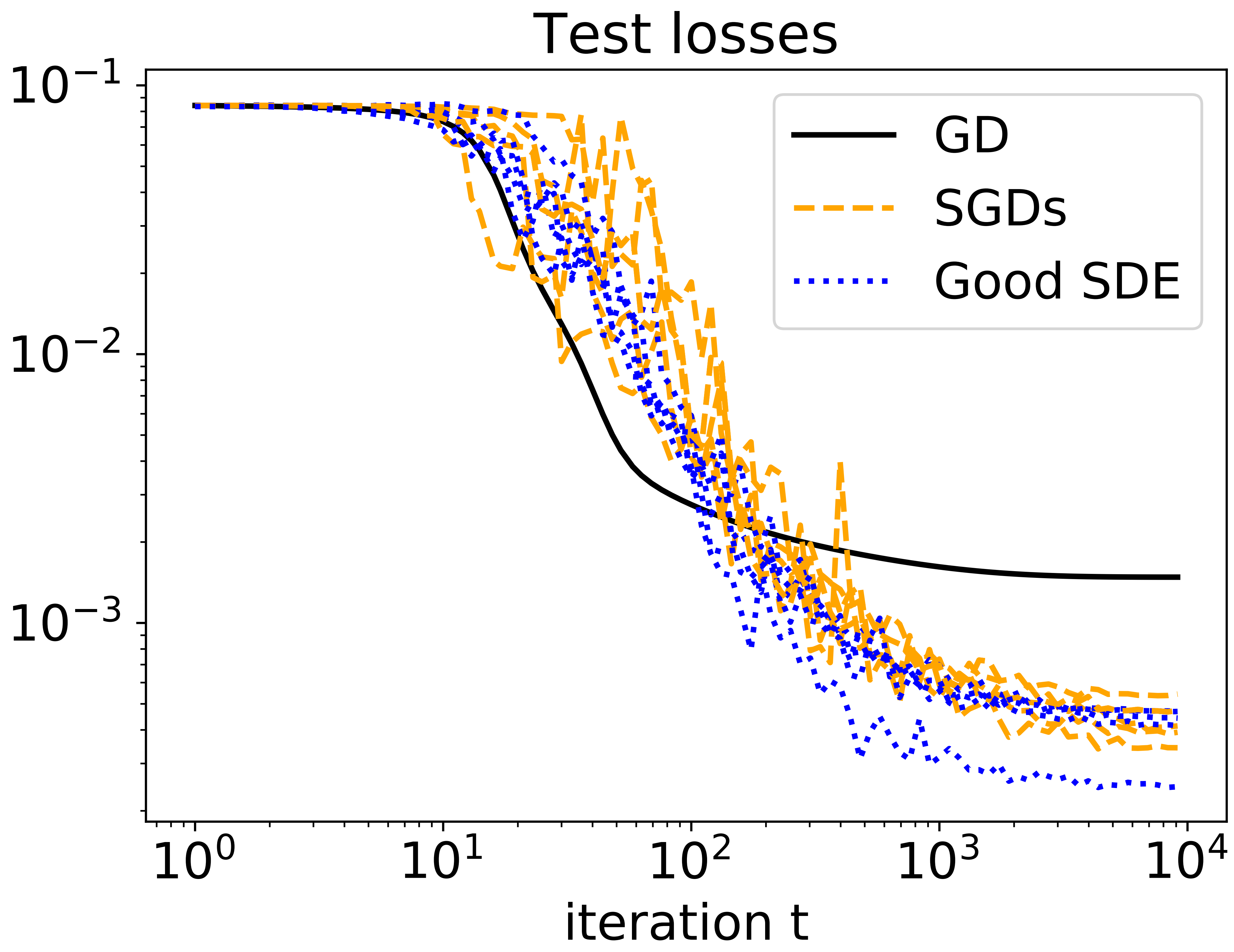
"Wrong SDE": $\mathrm{d}u_t = -\nabla_u L(w_t)\mathrm{d}t + 2\sqrt{\gamma n^{-1} L(w_t)}v_t \odot \mathrm{d}\tilde{B}_t$





Bonus 0: other SDE modelisations

"Our SDE": $\mathrm{d}u_t = -\nabla_u L(w_t)\mathrm{d}t + 2\sqrt{\gamma n^{-1} L(w_t)}v_t \odot [X^\top \mathrm{d}B_t]$



Bonus 1: adding label noise

Perturb label at time t $\tilde{y}_{i_t} = y_{i_t} + \Delta_t$ where $\Delta_t \sim \mathrm{unif}\{2\delta_t, -2\delta_t\}$

$(\delta_t)_{t\in\mathbb{N}} \in \mathbb{R}_+^{\mathbb{N}}$

"Slowed down loss": $\tilde{L}(w_t) = L(w_t) + \delta_t^2$

Modified: $$\tilde{\alpha}_\infty = \alpha \odot \exp\left(-2\gamma \mathrm{diag}\left(\frac{X^\top X}{n}\right) \int_0^{+\infty} \tilde{L}(\beta_s)\mathrm{d}s \right)$$

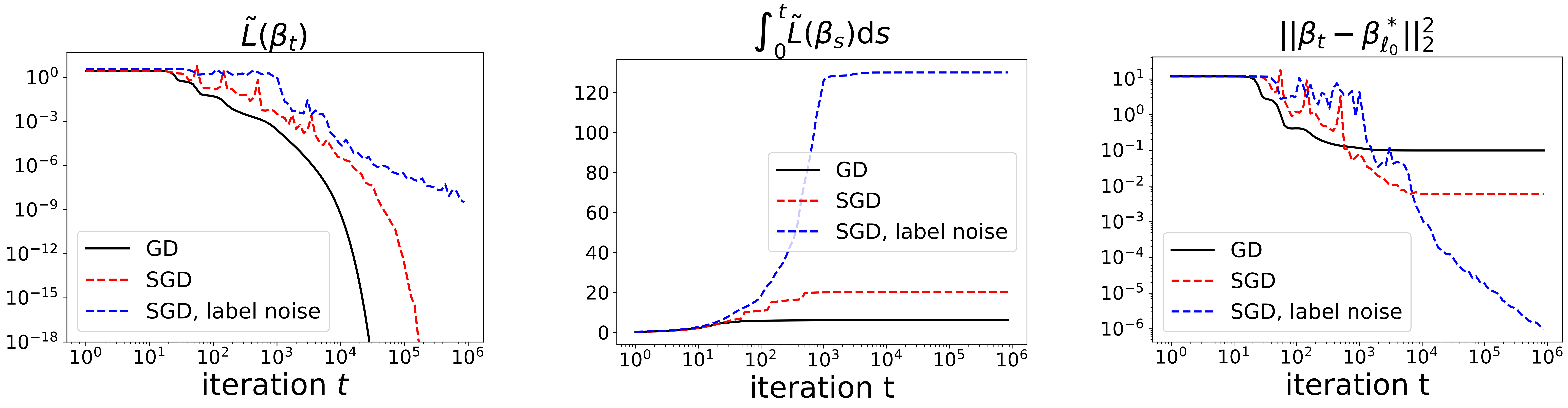
Bonus 1: adding label noise

Perturb label at time t: $\tilde{y}_{i_t} = y_{i_t} + \Delta_t$ where $\Delta_t \sim \mathrm{unif}\{2\delta_t, -2\delta_t\}$ $(\delta_t)_{t\in\mathbb{N}} \in \mathbb{R}_+^{\mathbb{N}}$

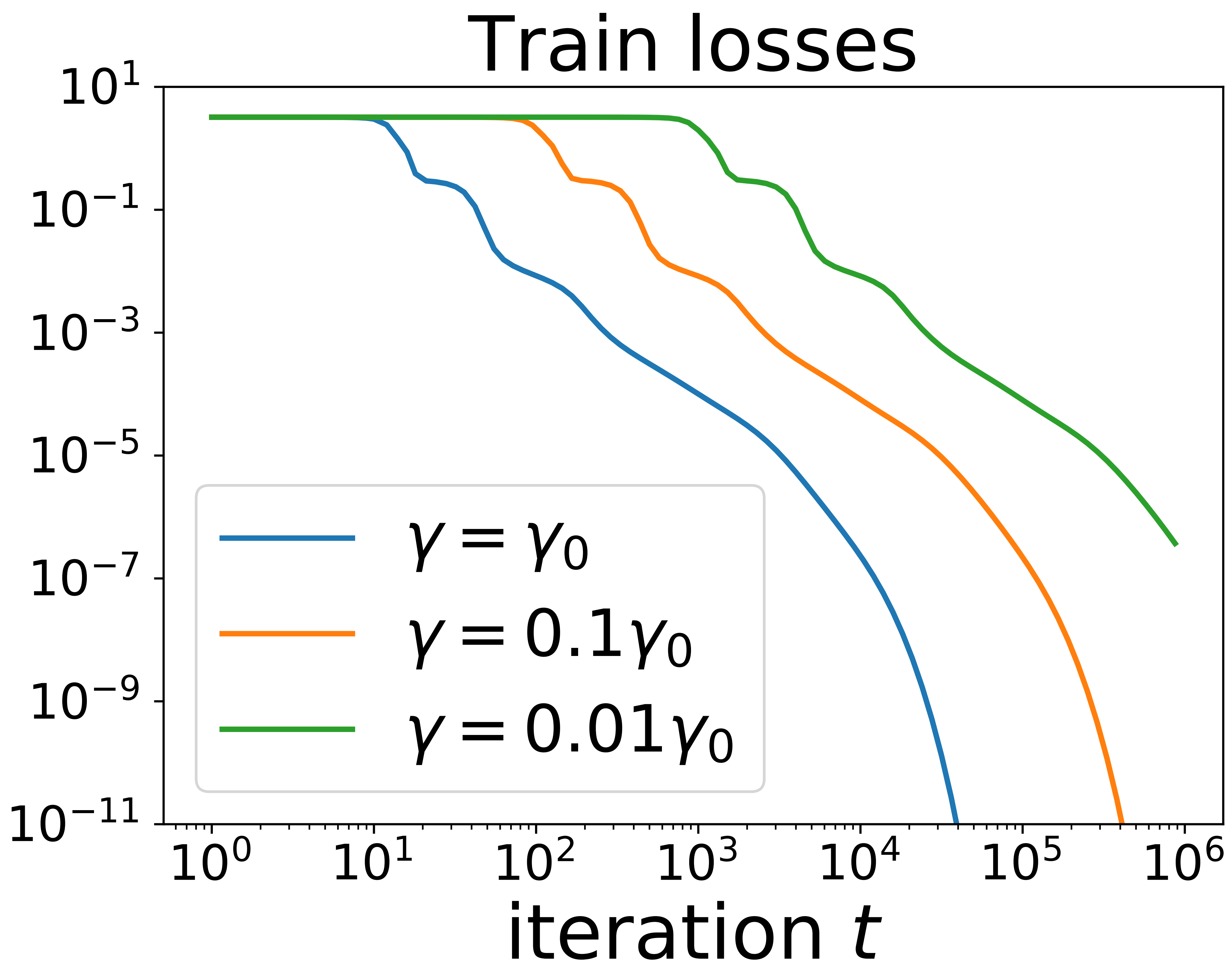
Experimental setup:

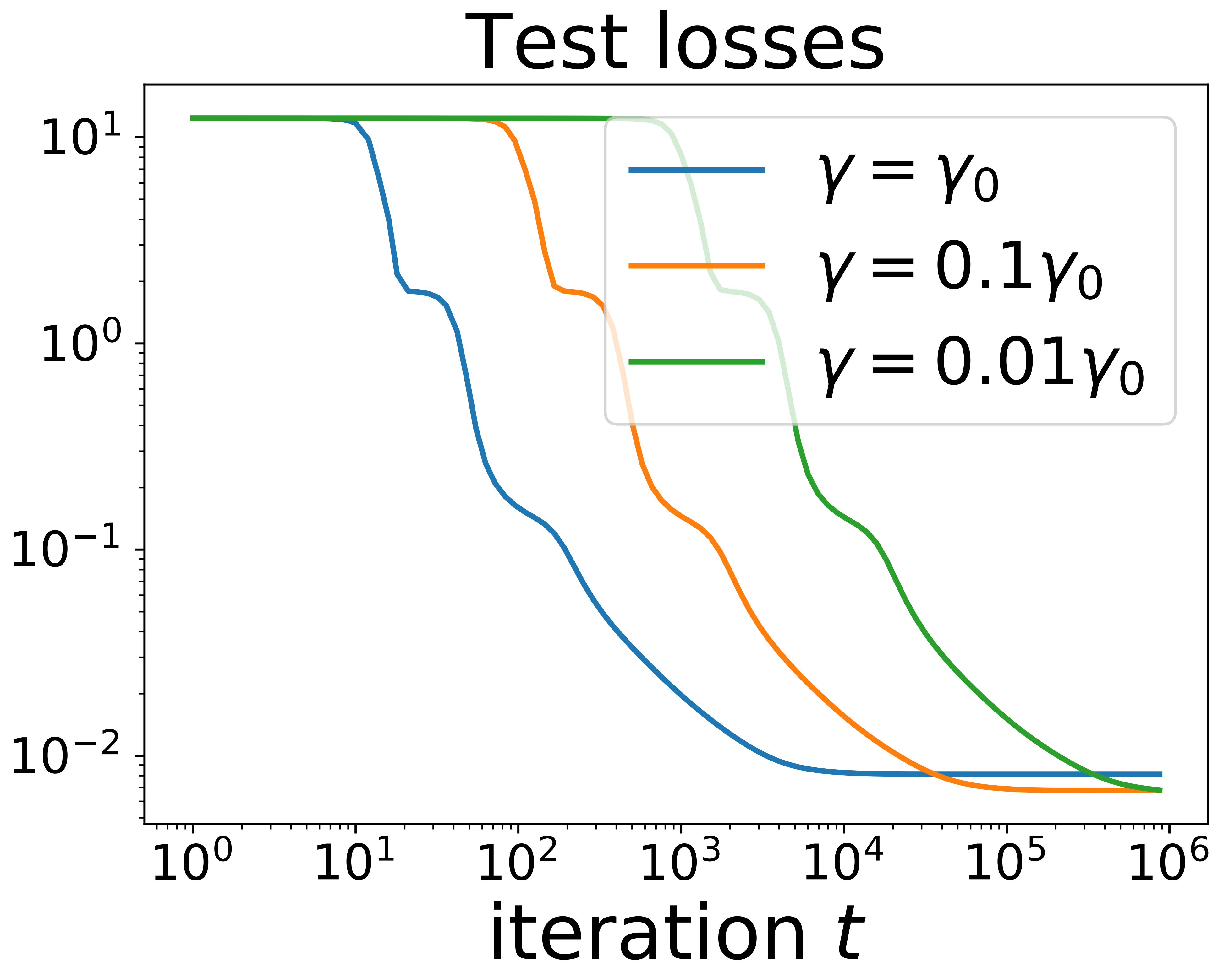
$n = 40$ $d = 100$ $\|\beta^*_{\ell_0}\|_0 = 5$ $t \leq 10^3$: $\delta_t = 1$

$x_i \sim \mathcal{N}(0, I)$ $y_i = \langle x_i, \beta^*_{\ell_0} \rangle$ $t > 10^3$: $\delta_t = 0$



Bonus 2: step-size for GD has very little impact on the implicit bias

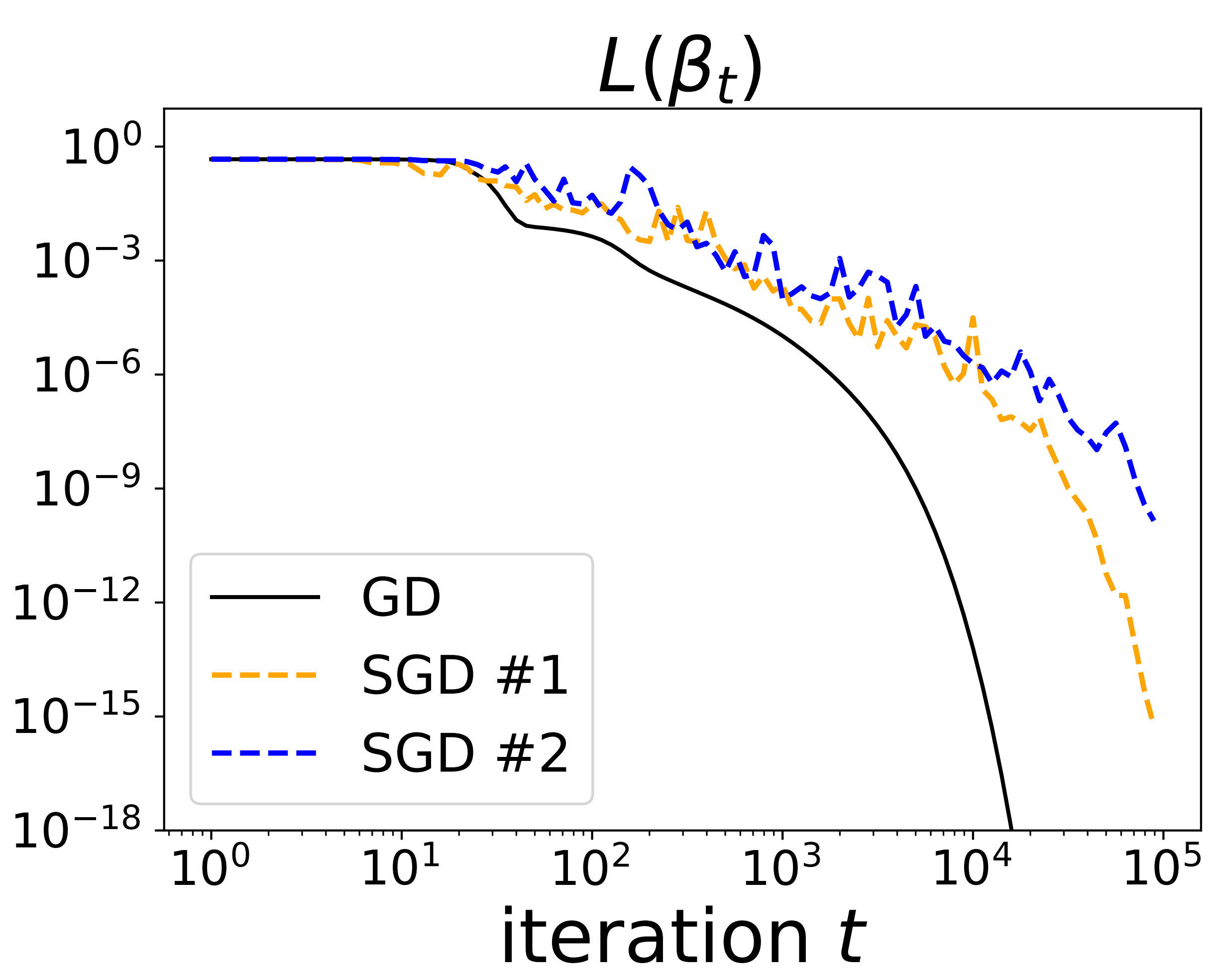


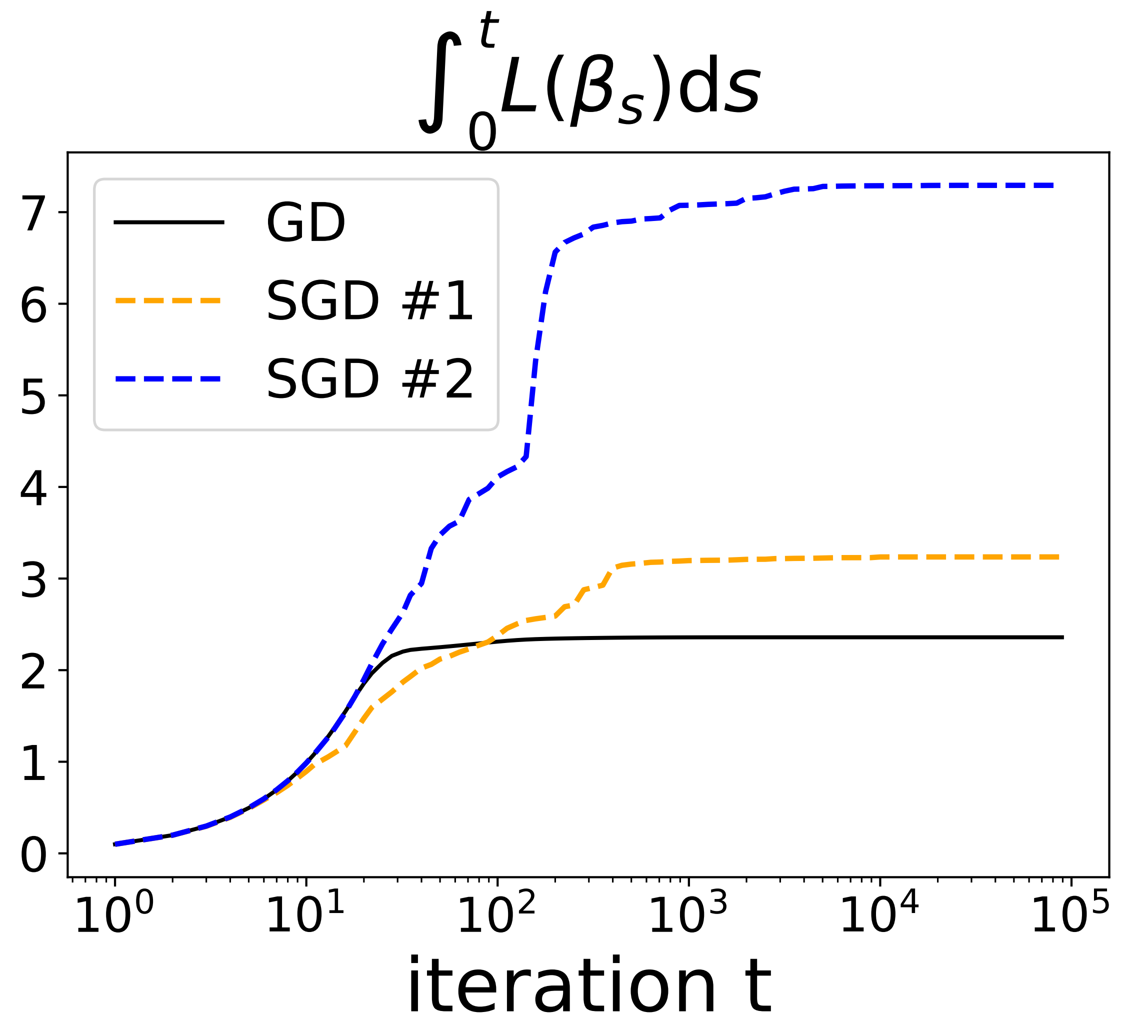


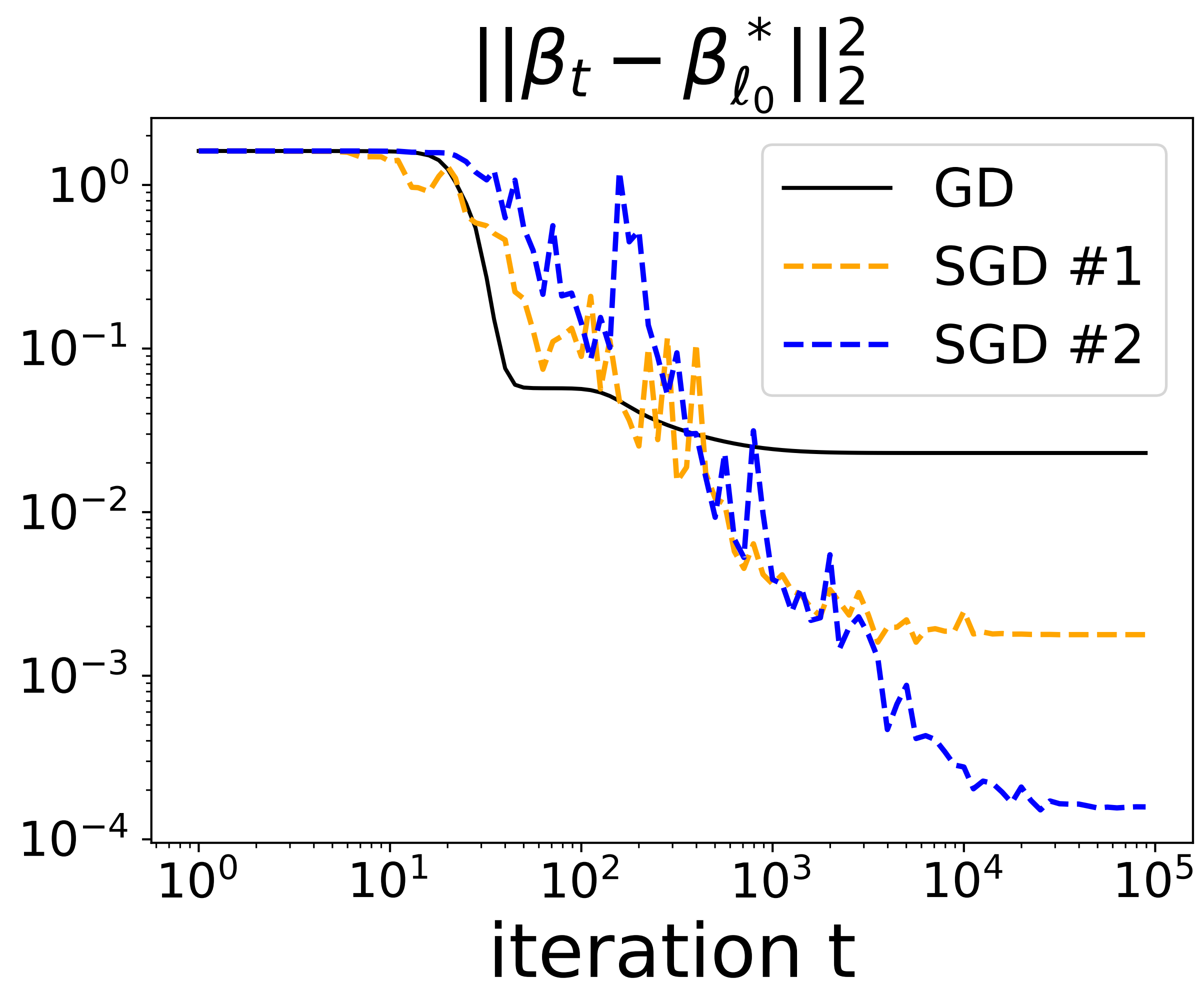
Bonus 3: the slower the training loss, the better the bias.

Setting: $n = 40$ $d = 100$ $\|\beta^*_{\ell_0}\|_0 = 5$

$(\alpha = 0.1)$ $x_i \sim \mathcal{N}(0, I)$ $y_i = \langle x_i, \beta^*_{\ell_0} \rangle$







stochastic !

$$\underbrace{\alpha_\infty}_{\text{“effective” initialisation}} = \alpha \odot \exp\Big(-2\gamma \mathrm{diag}\Big(\frac{X^\top X}{n}\Big) \int_0^{+\infty} \underbrace{L(\beta_s)\mathrm{d}s}_{\text{training loss}} \Big) < \overbrace{\alpha}^{\text{initialisation scale}}$$