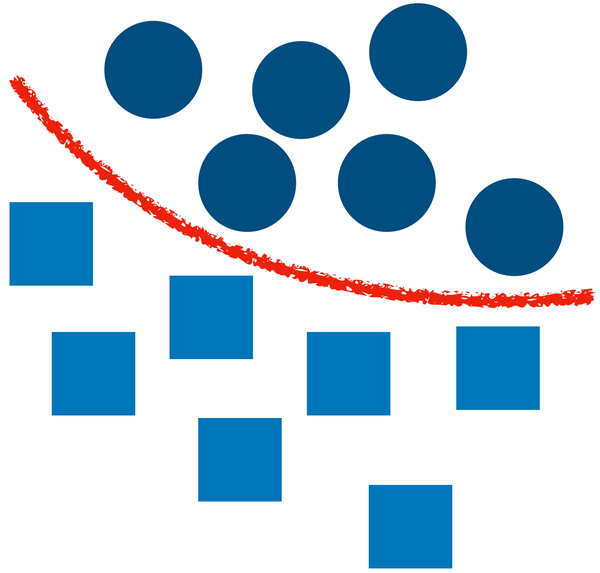
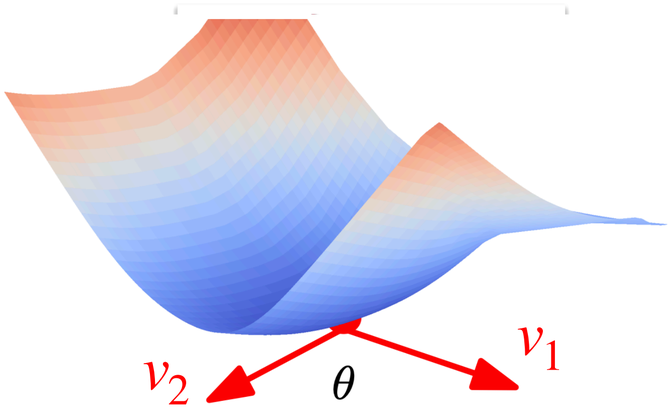
Unveiling the Hessian's Connection to the Decision Boundary



Anna Dawid

FLATIRON INSTITUTE
Center for Computational Quantum Physics



Machine learning and statistical physics back together again, Aug 9, 2023

Amazing collaborators

Les Houches Summer School open problem

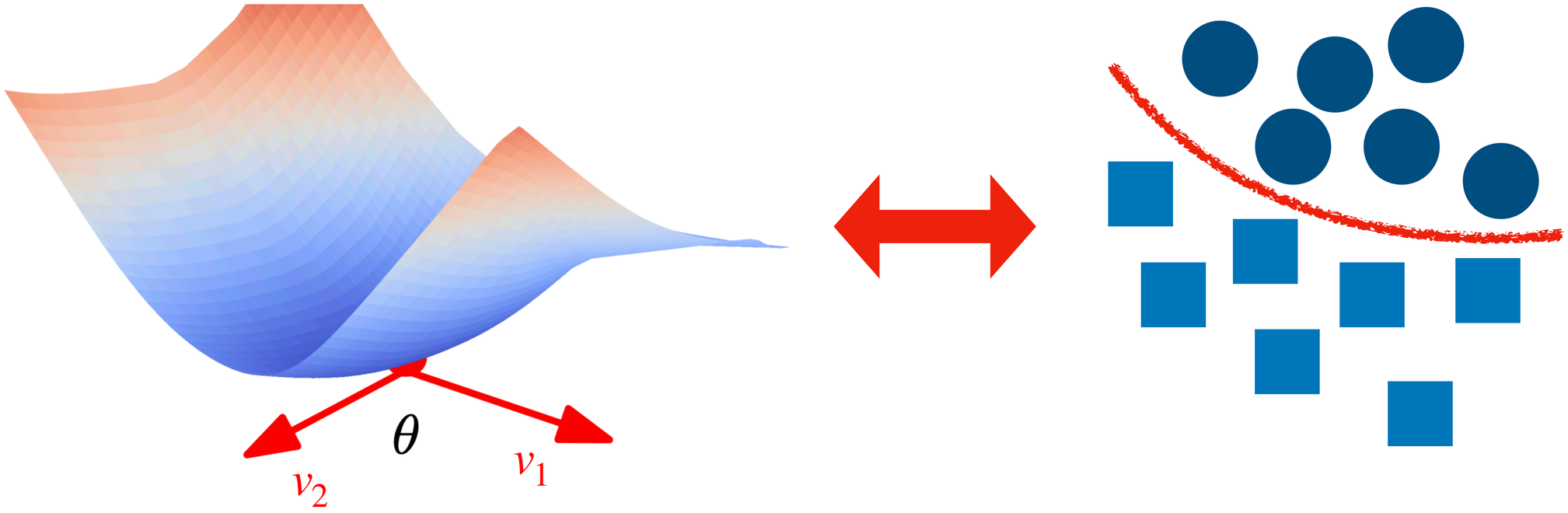
Mahalakshmi Sabanayagam
TU Munich

Freya Behrens
EPFL

Urte Adomaityte
King’s College London

Take-home message

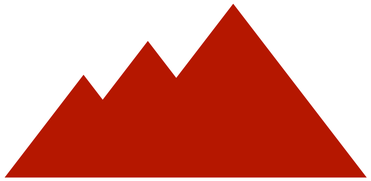
The top Hessian eigenvectors encode the decision boundary



parameter space

input space

Outline



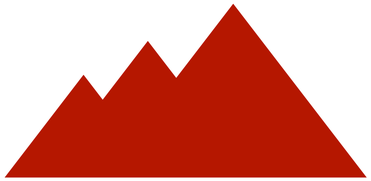
Empirical mess
in the Land of the Loss

Hessian and the decision boundary: toy example

Selected interesting consequences

- measure of the decision boundary complexity
- margin estimation
- perspectives for new generalization measure

Outline

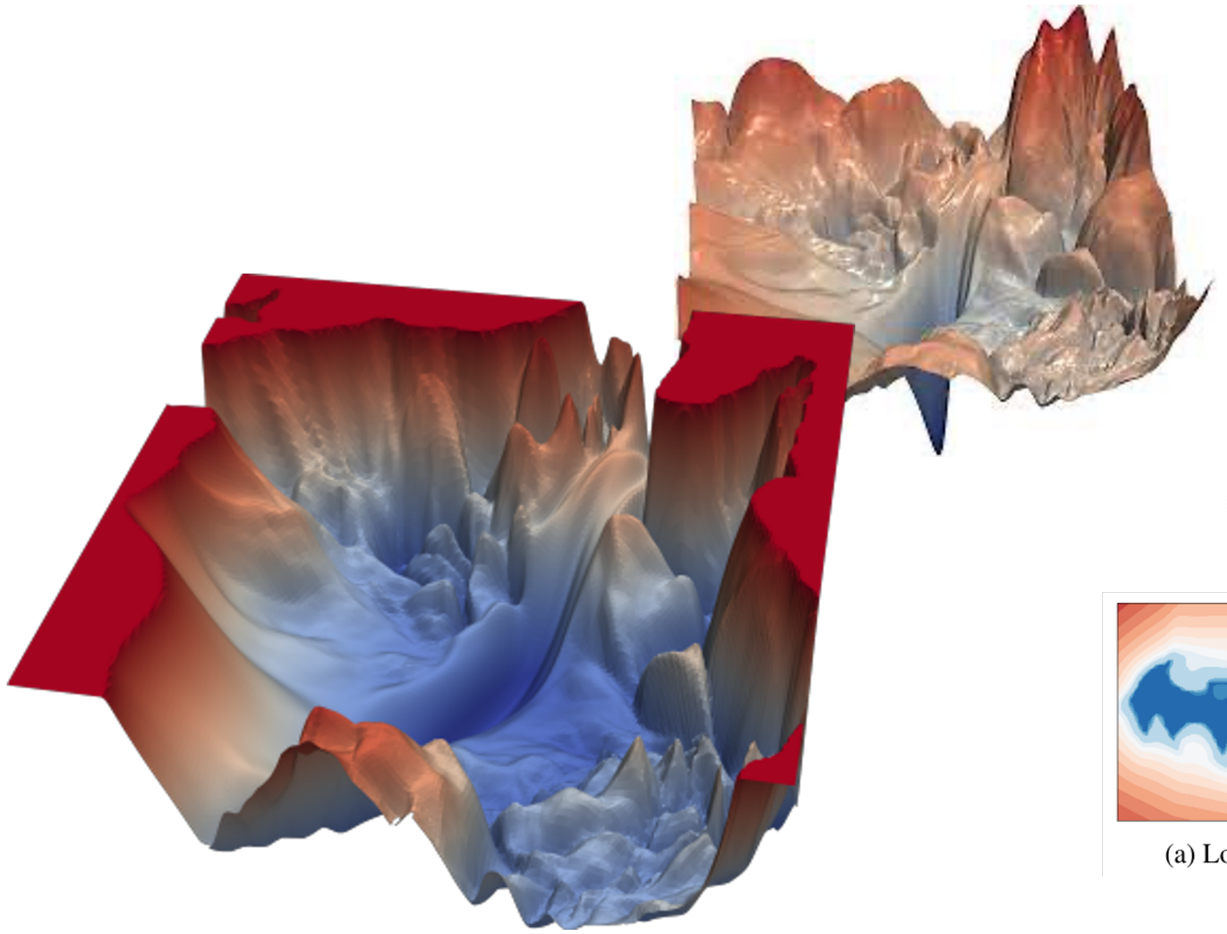


Empirical mess
in the Land of the Loss

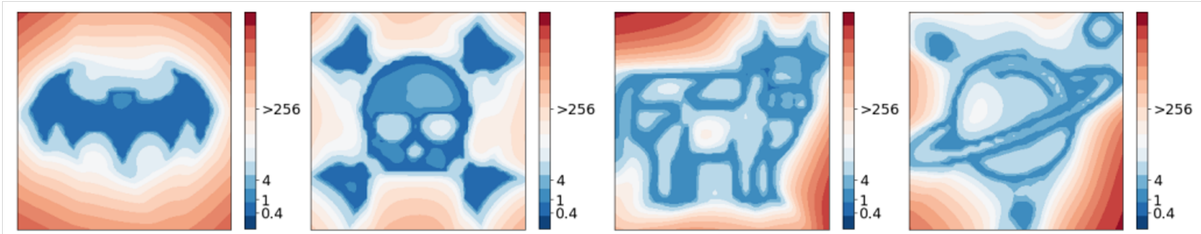
Hessian and the decision
boundary: toy example

Selected interesting
consequences

Non-convex loss landscapes of deep networks?



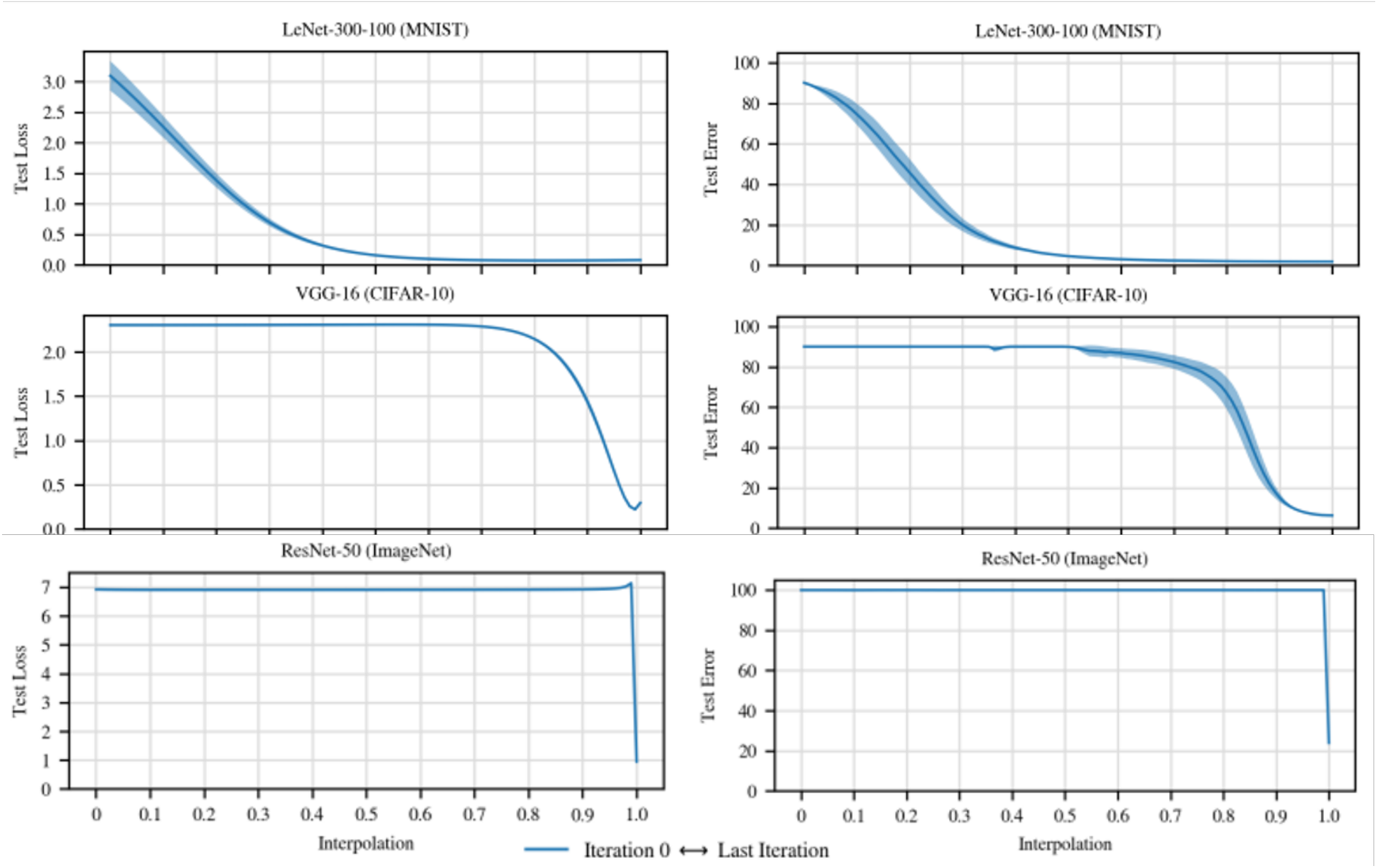
one can empirically find arbitrary 2D binary patterns
inside loss surfaces of popular neural networks!



(a) Loss surface on FashionMNIST dataset (b) Loss surface on CIFAR10 dataset

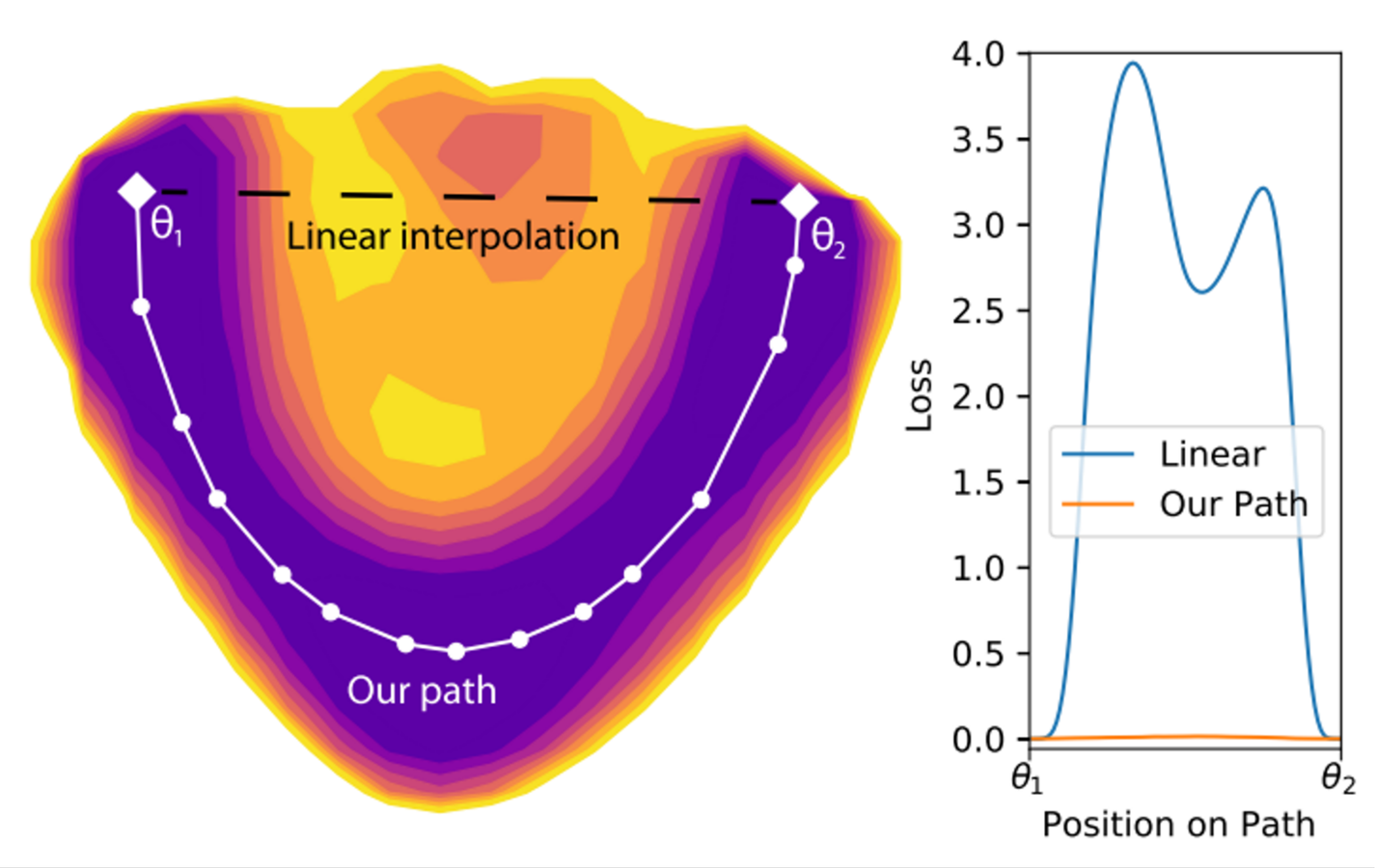
Li et al. (2018) Visualizing the Loss Landscape of Neural Nets. arXiv:1712.09913
Skorokhodov & Burtsev (2019) Loss Landscape Sightseeing with Multi-Point Optimization. arXiv:1910.03867

But… Path from a proper initialization to a minimum



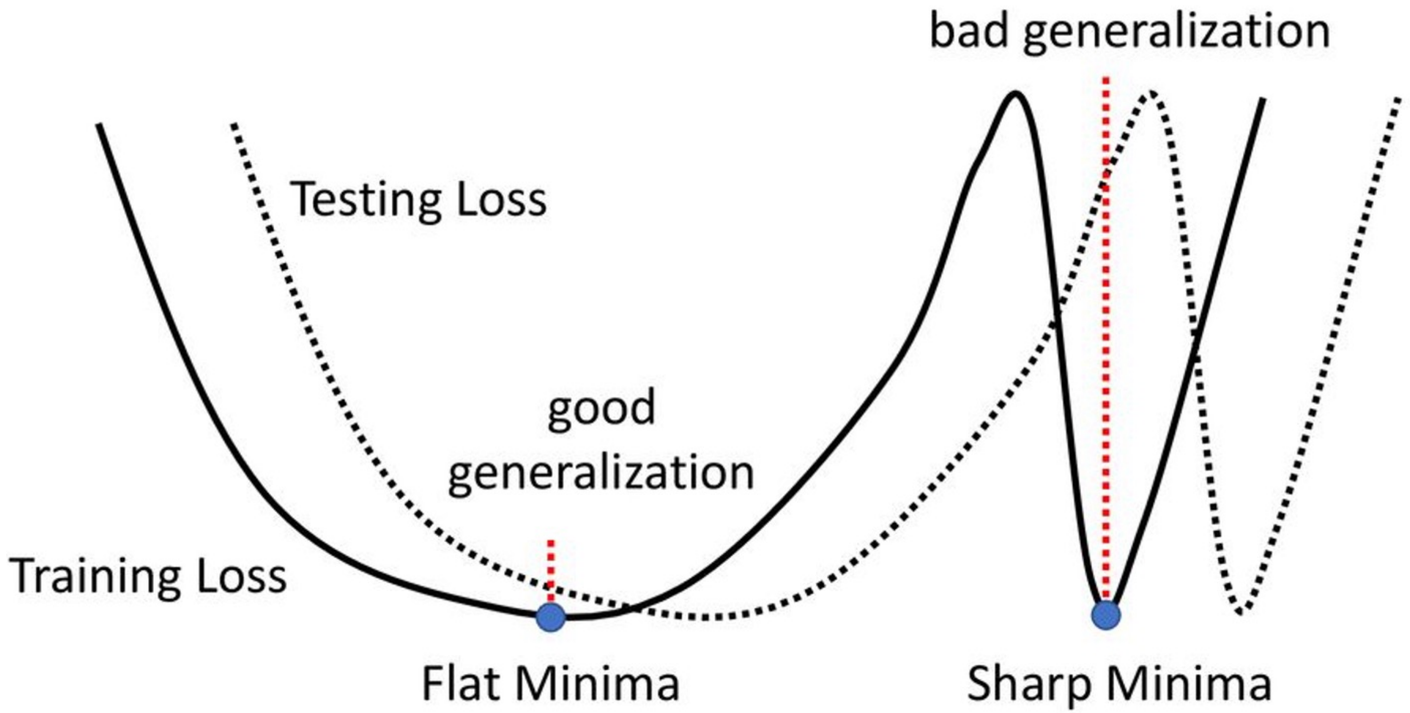
Goodfellow et al. (2015) Qualitatively Characterizing Neural Network Optimization Problems. arXiv:1412.6544
Frankle (2020) Revisiting “Qualitatively Characterizing Neural Network Optimization Problems”. arXiv:1910.03867

But… Minima are connected



Draxler et al. (2019) Essentially No Barriers in Neural Network Energy Landscape. arXiv:1803.00885

Optimizers reach flat minima that generalize better?



Keskar et al. (2017) On large-batch training for deep learning: Generalization gap and sharp minima. arXiv:1609.04836

Optimizers reach flat minima that generalize better?

Many works: **Yes!** **Flatter minima generalize better**

Jiang et al. (2019) Fantastic Generalization Measures and Where to Find Them. arxiv:1912.02178
Andriushchenko et al. (2023) A modern look at the relationship between sharpness and generalization. arXiv:2302.07011

Optimizers reach flat minima that generalize better?

Many works: **Yes!** **Flatter minima generalize better**

Also many works:

- standard (Hessian trace, spectra norm) flatness measures do **not** correlate well with generalization
- they are **not** reparametrization invariant!

$$\mathrm{ReLU}(\alpha x) = \alpha \mathrm{ReLU}(x) \qquad \text{(non-negative homogeneous)}$$

$$\mathrm{ReLU}\big(x \cdot (\alpha\theta_1)\big)\ \theta_2 = \mathrm{ReLU}(x \cdot \theta_1) \cdot (\alpha\theta_2)$$

1st-layer $\alpha\theta_1$ and 2nd-layer θ_2 | 1st-layer θ_1 and 2nd-layer $\alpha\theta_2$

arbitrarily sharp minima with effectively identical predictions!

Jiang et al. (2019) Fantastic Generalization Measures and Where to Find Them. arxiv:1912.02178
Andriushchenko et al. (2023) A modern look at the relationship between sharpness and generalization. arXiv:2302.07011

Optimizers reach flat minima that generalize better?

Many works: **Yes!** **Flatter minima generalize better**

Also many works:

- standard (Hessian trace, spectra norm) flatness measures do **not** correlate well with generalization
- they are **not** reparametrization invariant!

Also many works:
Let's **fix the flatness measure**!

$$\mathrm{ReLU}(\alpha x) = \alpha \mathrm{ReLU}(x) \quad \text{(non-negative homogeneous)}$$

$$\mathrm{ReLU}\big(x \cdot (\alpha\theta_1)\big)\ \theta_2 = \mathrm{ReLU}(x \cdot \theta_1) \cdot (\alpha\theta_2)$$

1st-layer $\alpha\theta_1$ and 2nd-layer θ_2 | 1st-layer θ_1 and 2nd-layer $\alpha\theta_2$

arbitrarily sharp minima with
effectively identical predictions!

Jiang et al. (2019) Fantastic Generalization Measures and Where to Find Them. arxiv:1912.02178
Andriushchenko et al. (2023) A modern look at the relationship between sharpness and generalization. arXiv:2302.07011

Optimizers reach flat minima that generalize better?

Many works: **Yes!** **Flatter minima generalize better**

Also many works:

- standard (Hessian trace, spectra norm) flatness measures do **not** correlate well with generalization
- they are **not** reparametrization invariant!

Also many works:
Let's **fix the flatness measure**!

$$\mathrm{ReLU}(\alpha x) = \alpha \mathrm{ReLU}(x) \quad \text{(non-negative homogeneous)}$$

$$\mathrm{ReLU}\big(x \cdot (\alpha\theta_1)\big)\ \theta_2 = \mathrm{ReLU}(x \cdot \theta_1) \cdot (\alpha\theta_2)$$

1st-layer $\alpha\theta_1$ and 2nd-layer θ_2 | 1st-layer θ_1 and 2nd-layer $\alpha\theta_2$

arbitrarily sharp minima with
effectively identical predictions!

Also many works:
No, they also don't work!
They **correlate rather with hyperparameters** than the generalization itself.

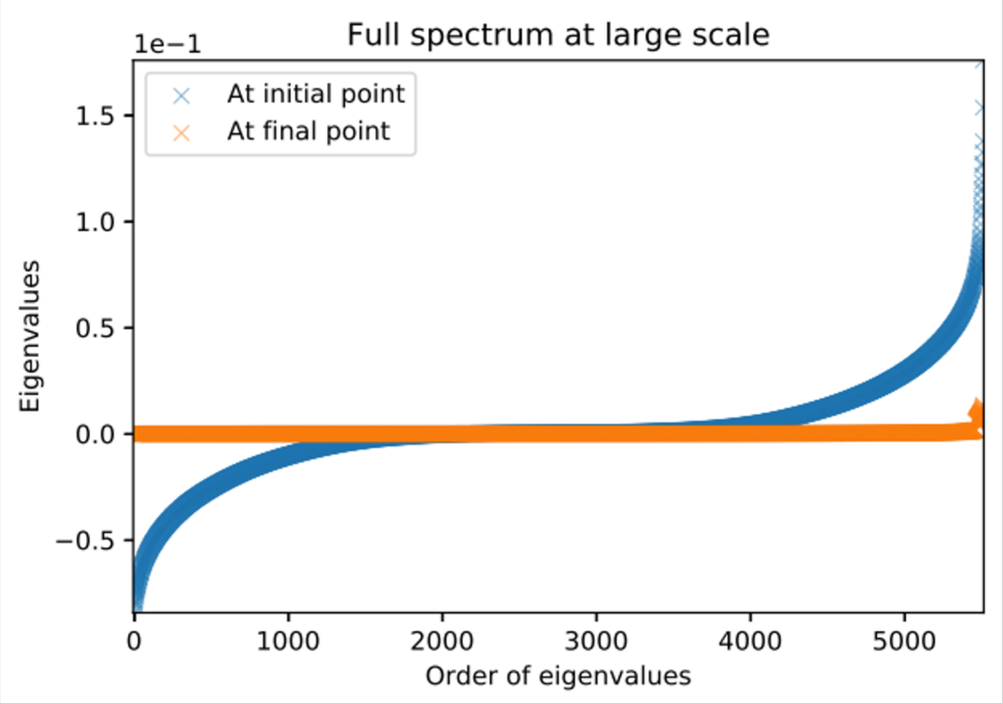
Jiang et al. (2019) Fantastic Generalization Measures and Where to Find Them. arxiv:1912.02178
Andriushchenko et al. (2023) A modern look at the relationship between sharpness and generalization. arXiv:2302.07011

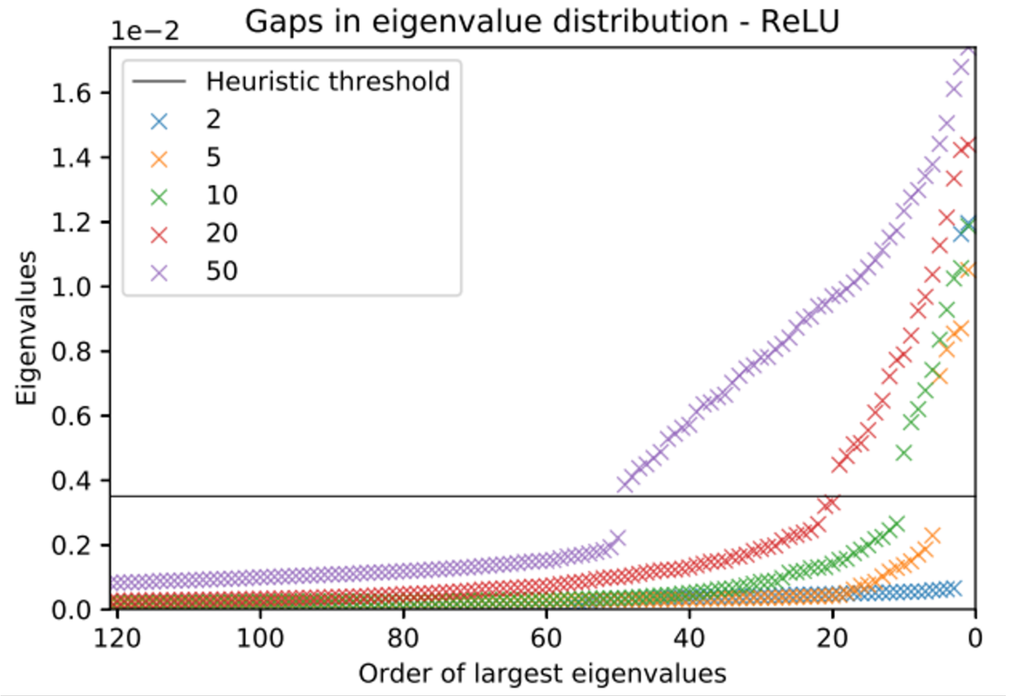
Hessian has universal properties

across deep learning setups

a.k.a. the community does agree on something!

$$H_{ij} = \frac{\partial^2 \mathscr{L}(\mathscr{D}, \theta)}{\partial \theta_i \partial \theta_j}\bigg|_{\theta=\theta^*}$$





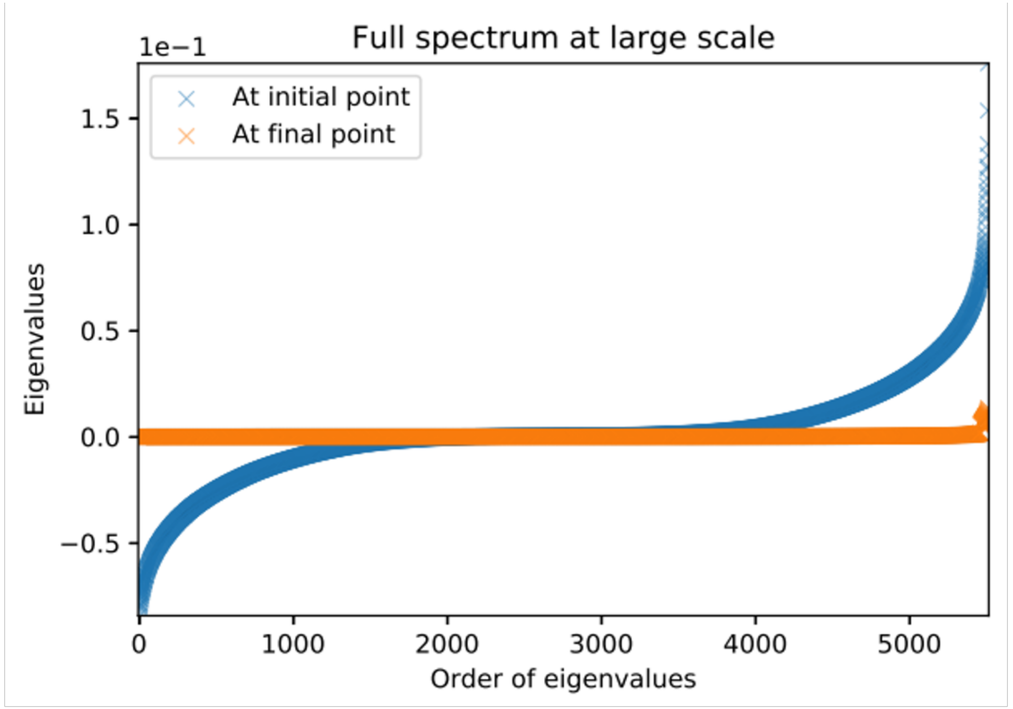
Sagun et al. (2019) Empirical Analysis of the Hessian of Over-Parametrized Neural Networks. arXiv:1706.04454

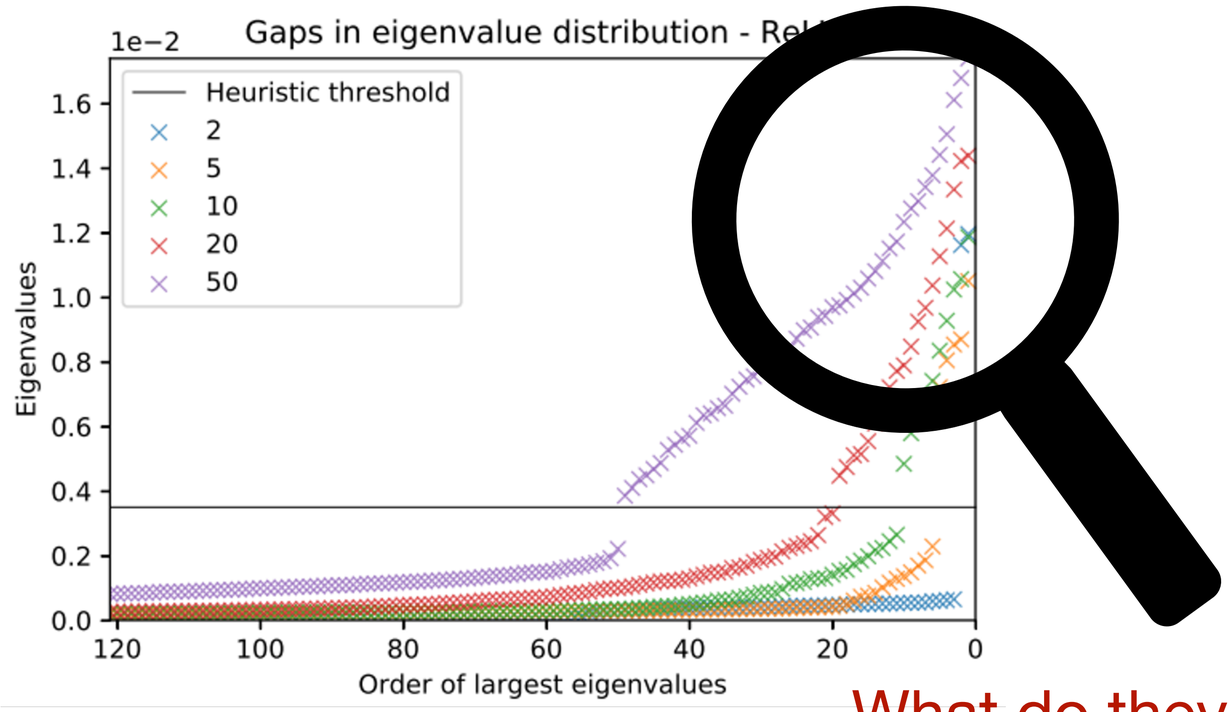
Hessian has universal properties

across deep learning setups

a.k.a. the community does agree on something!

$$H_{ij} = \frac{\partial^2 \mathscr{L}(\mathscr{D}, \theta)}{\partial \theta_i \partial \theta_j}\bigg|_{\theta=\theta^*}$$





What do they encode?

Sagun et al. (2019) Empirical Analysis of the Hessian of Over-Parametrized Neural Networks. arXiv:1706.04454

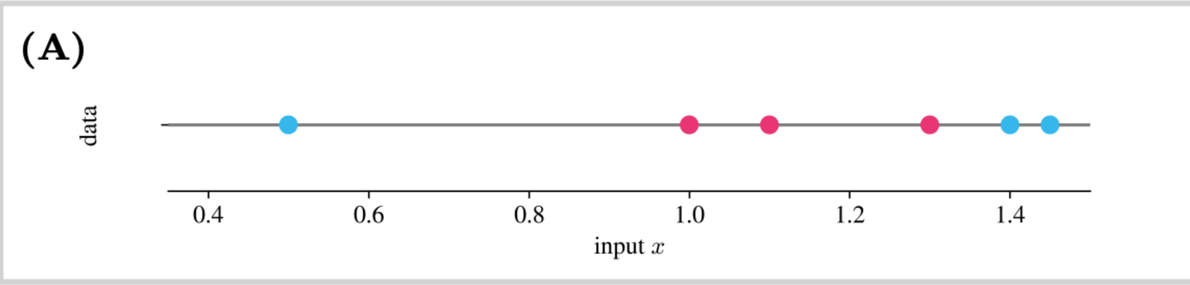
Outline

Empirical mess
in the Land of the Loss

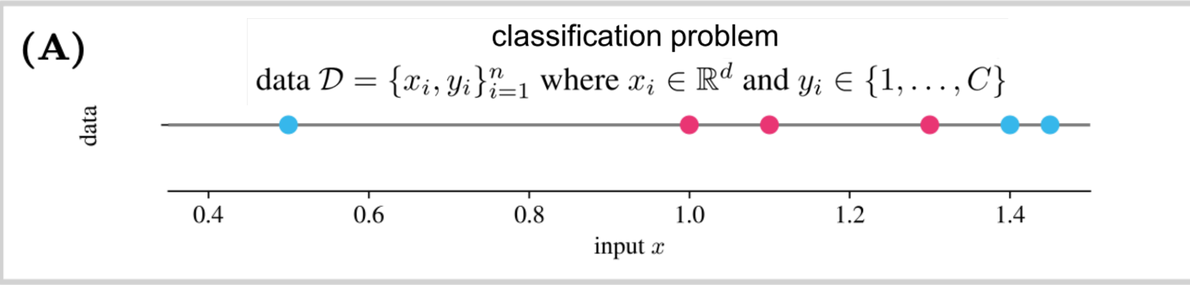
Hessian and the decision boundary: toy example

Selected interesting consequences

Toy example

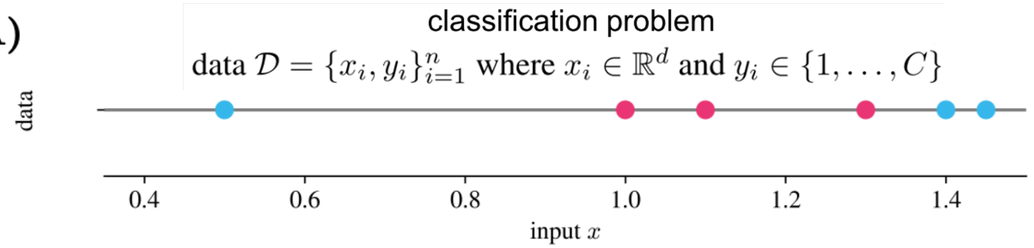


Toy example


(A)
classification problem
data $\mathcal{D} = \{x_i, y_i\}_{i=1}^n$ where $x_i \in \mathbb{R}^d$ and $y_i \in \{1, \ldots, C\}$
data
0.4
0.6
0.8
1.0
1.2
1.4
input x

Toy example

(A)



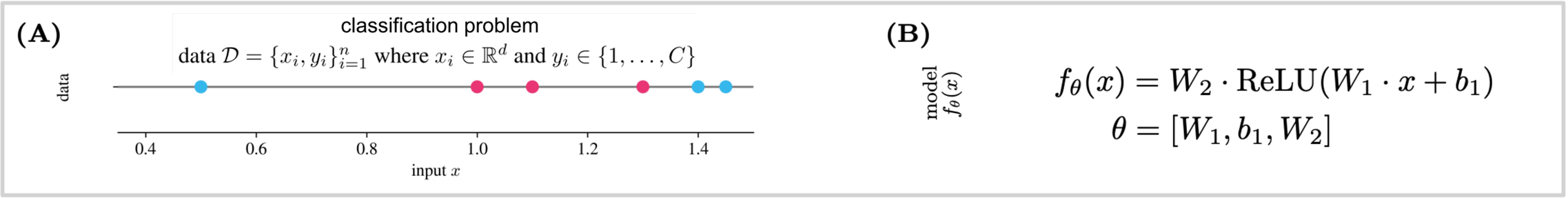
(B)

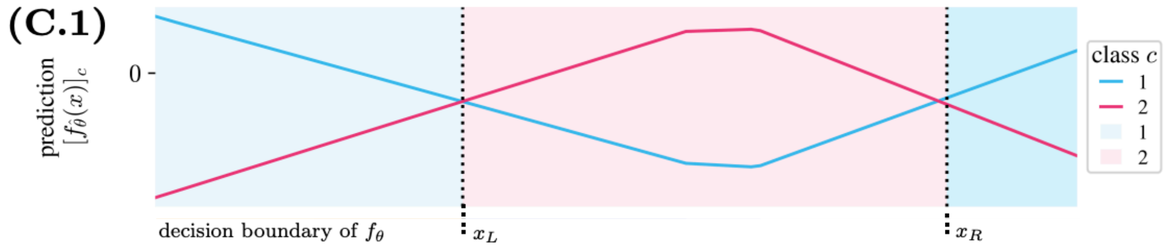
model $f_\theta(x)$

$$f_\theta(x) = W_2 \cdot \mathrm{ReLU}(W_1 \cdot x + b_1)$$

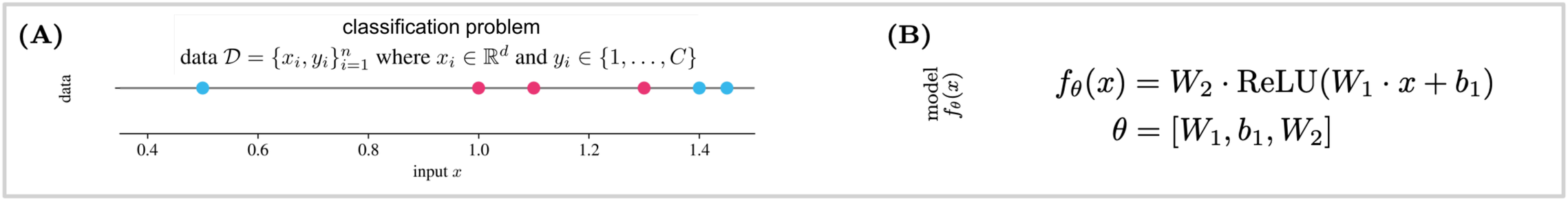
$$\theta = [W_1, b_1, W_2]$$

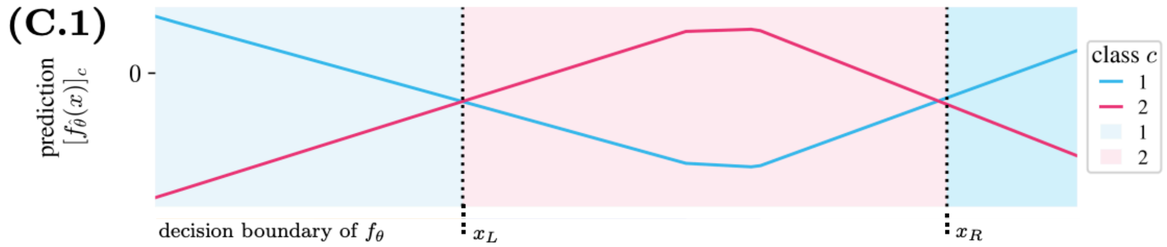
Toy example

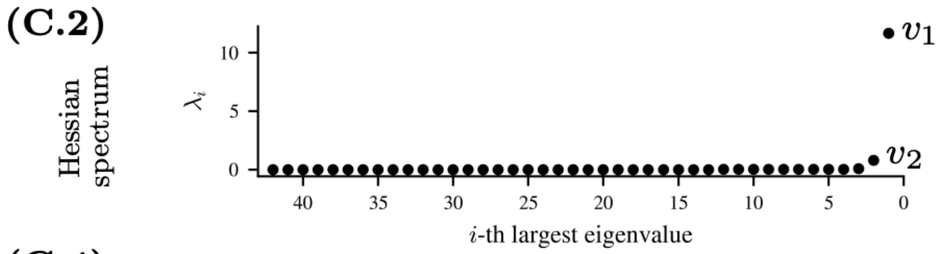

(A)
classification problem
data $\mathcal{D} = \{x_i, y_i\}_{i=1}^n$ where $x_i \in \mathbb{R}^d$ and $y_i \in \{1, \ldots, C\}$
data
0.4
0.6
0.8
1.0
1.2
1.4
input x
(B)
model $f_\theta(x)$
$f_\theta(x) = W_2 \cdot \mathrm{ReLU}(W_1 \cdot x + b_1)$
$\theta = [W_1, b_1, W_2]$

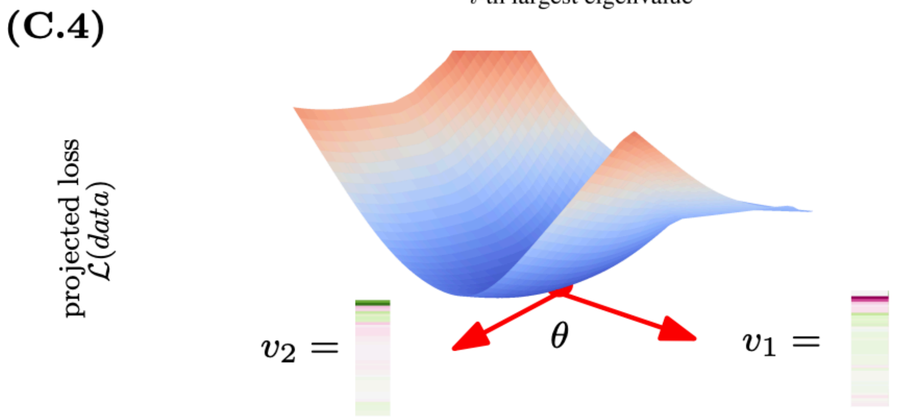

(C.1)
prediction $[f_\theta(x)]_c$
0
class c
1
2
1
2
decision boundary of f_θ
x_L
x_R

Toy example

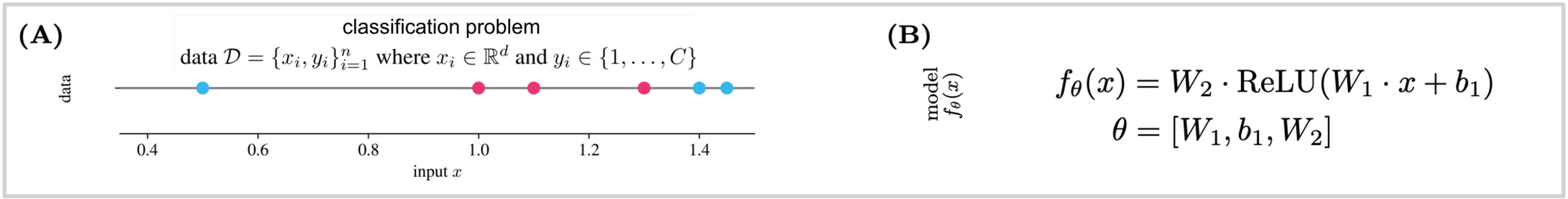

(A)
classification problem
data $\mathcal{D} = \{x_i, y_i\}_{i=1}^n$ where $x_i \in \mathbb{R}^d$ and $y_i \in \{1, \ldots, C\}$
data
0.4
0.6
0.8
1.0
1.2
1.4
input x
(B)
model $f_\theta(x)$
$f_\theta(x) = W_2 \cdot \mathrm{ReLU}(W_1 \cdot x + b_1)$
$\theta = [W_1, b_1, W_2]$

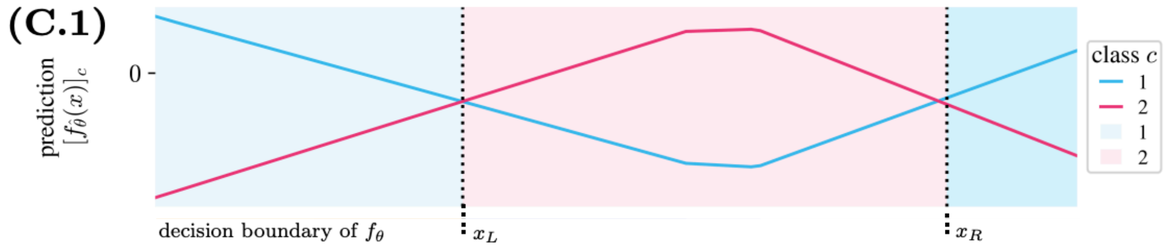

(C.1)
prediction $[f_\theta(x)]_c$
0
class c
1
2
1
2
decision boundary of f_θ
x_L
x_R

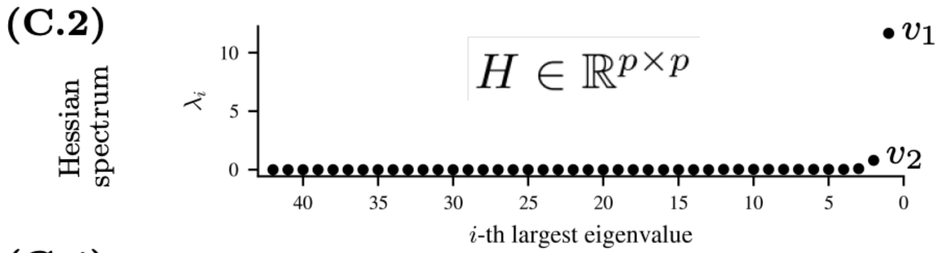

(C.2)
Hessian spectrum
λ_i
10
5
0
v_1
v_2
40
35
30
25
20
15
10
5
0
i-th largest eigenvalue

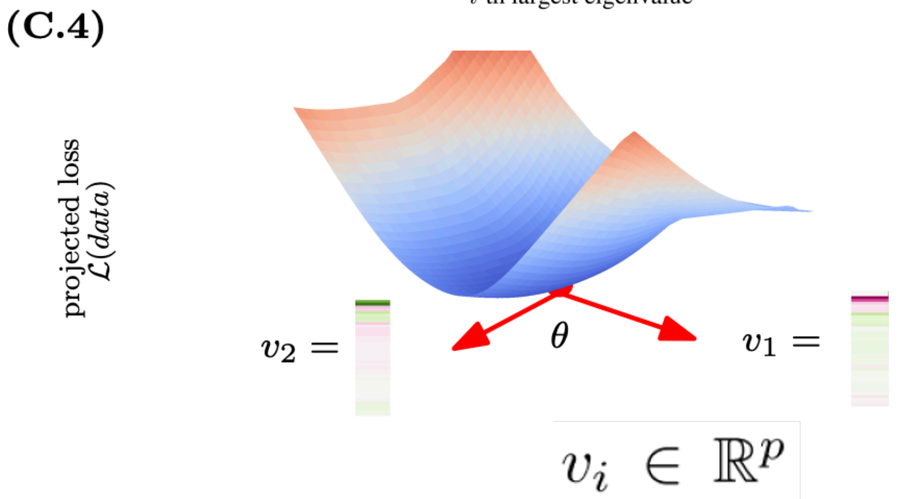

(C.4)
projected loss $\mathcal{L}(data)$
$v_2 =$
θ
$v_1 =$

Toy example

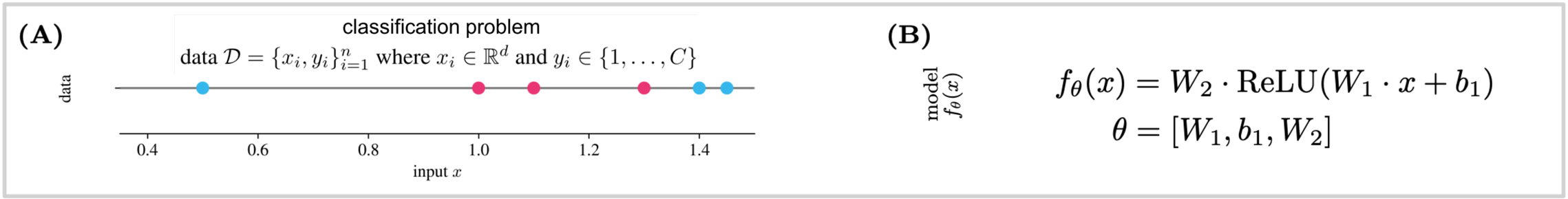

(A)
classification problem
data $\mathcal{D} = \{x_i, y_i\}_{i=1}^n$ where $x_i \in \mathbb{R}^d$ and $y_i \in \{1, \ldots, C\}$
data
0.4
0.6
0.8
1.0
1.2
1.4
input x
(B)
model $f_\theta(x)$
$f_\theta(x) = W_2 \cdot \mathrm{ReLU}(W_1 \cdot x + b_1)$
$\theta = [W_1, b_1, W_2]$

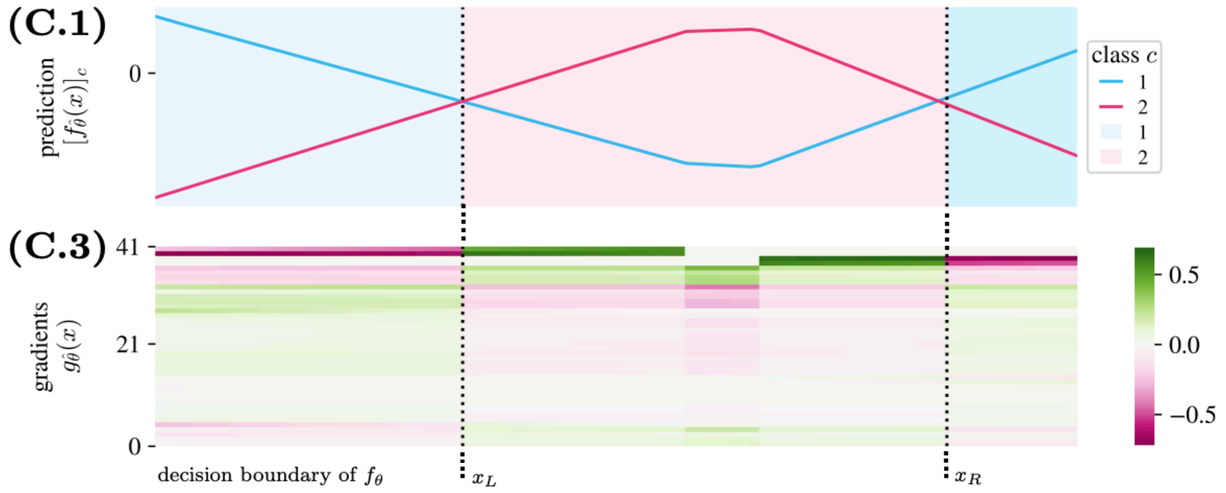

(C.1)
prediction $[f_\theta(x)]_c$
0
class c
1
2
1
2
decision boundary of f_θ
x_L
x_R

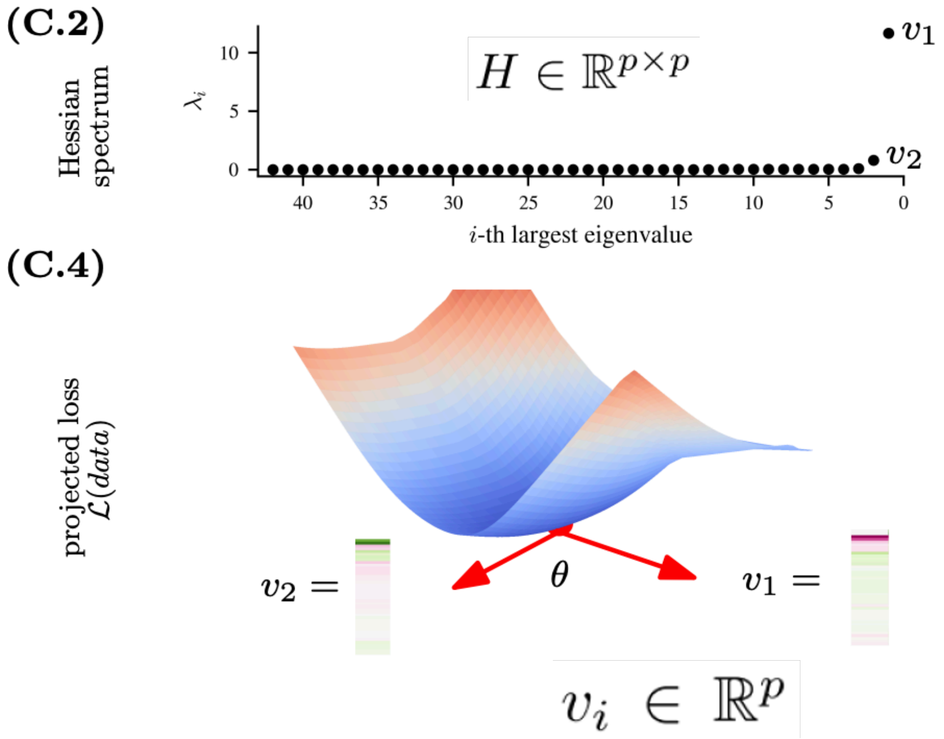

(C.2)
Hessian spectrum
$H \in \mathbb{R}^{p \times p}$
v_1
v_2
λ_i
10
5
0
40
35
30
25
20
15
10
5
0
i-th largest eigenvalue


(C.4)
projected loss $\mathcal{L}(data)$
$v_2 =$
θ
$v_1 =$
$v_i \in \mathbb{R}^p$

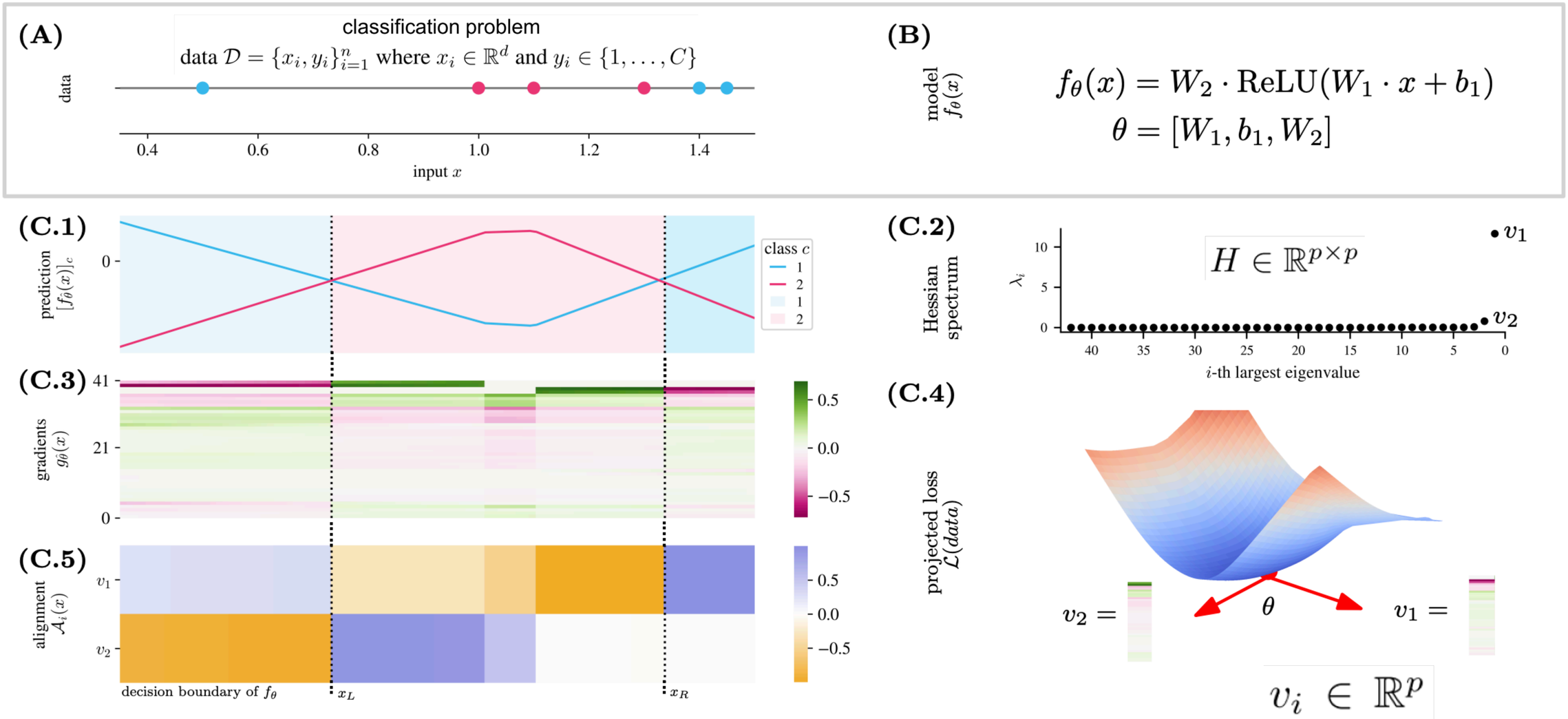
Toy example


(A)
classification problem
data $\mathcal{D} = \{x_i, y_i\}_{i=1}^n$ where $x_i \in \mathbb{R}^d$ and $y_i \in \{1, \ldots, C\}$
data
0.4
0.6
0.8
1.0
1.2
1.4
input x
(B)
model $f_\theta(x)$
$f_\theta(x) = W_2 \cdot \mathrm{ReLU}(W_1 \cdot x + b_1)$
$\theta = [W_1, b_1, W_2]$


(C.1)
prediction $[f_\theta(x)]_c$
0
class c
1
2
1
2
(C.3)
gradients $g_\theta(x)$
41
21
0
0.5
0.0
−0.5
decision boundary of f_θ
x_L
x_R


(C.2)
Hessian spectrum
$H \in \mathbb{R}^{p \times p}$
v_1
v_2
λ_i
10
5
0
40
35
30
25
20
15
10
5
0
i-th largest eigenvalue
(C.4)
projected loss $\mathcal{L}(data)$
$v_2 =$
θ
$v_1 =$
$v_i \in \mathbb{R}^p$

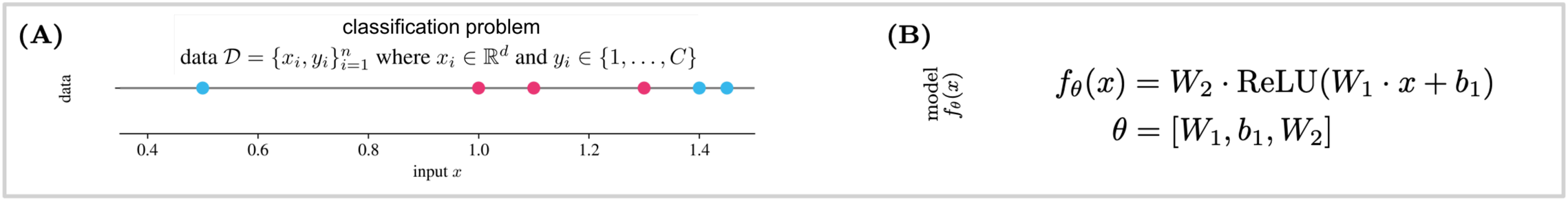
Toy example

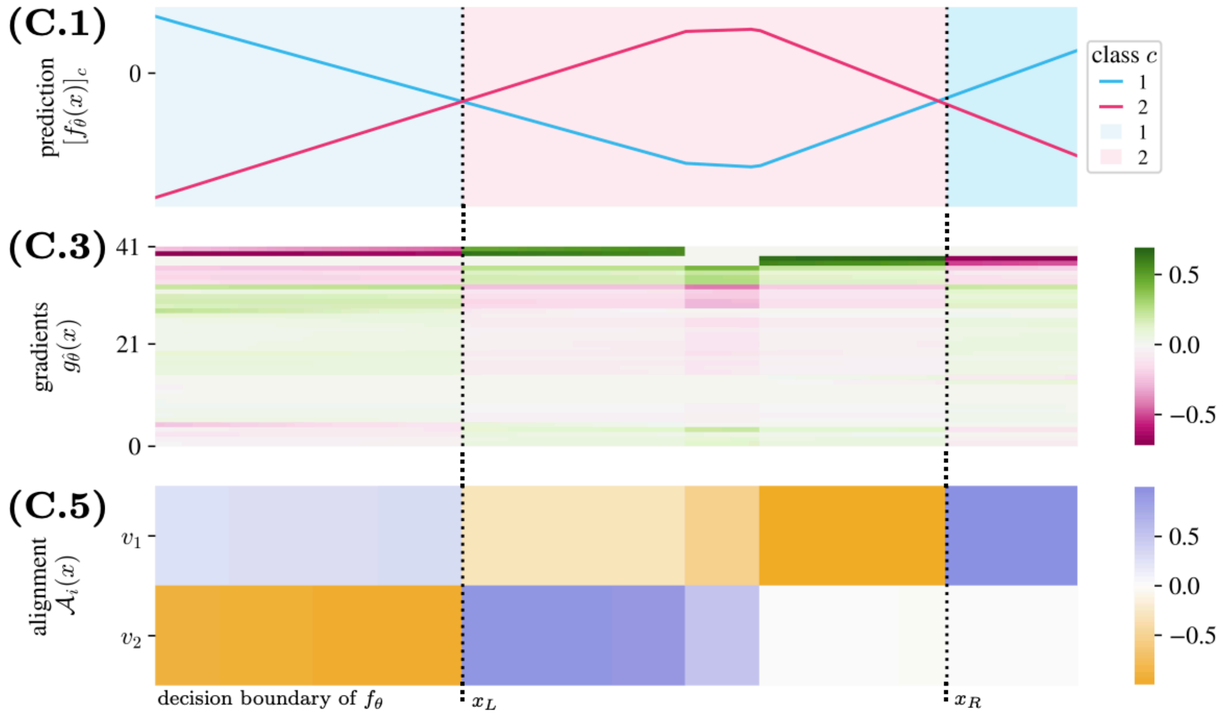

(A)
classification problem
data $\mathcal{D} = \{x_i, y_i\}_{i=1}^n$ where $x_i \in \mathbb{R}^d$ and $y_i \in \{1, \ldots, C\}$
data
0.4
0.6
0.8
1.0
1.2
1.4
input x
(B)
model $f_\theta(x)$
$f_\theta(x) = W_2 \cdot \mathrm{ReLU}(W_1 \cdot x + b_1)$
$\theta = [W_1, b_1, W_2]$
(C.1)
prediction $[f_\theta(x)]_c$
0
class c
1
2
1
2
(C.2)
Hessian spectrum
$H \in \mathbb{R}^{p \times p}$
v_1
v_2
λ_i
10
5
0
40
35
30
25
20
15
10
5
0
i-th largest eigenvalue
(C.3)
gradients $g_\theta(x)$
41
21
0
0.5
0.0
−0.5
(C.4)
projected loss $\mathcal{L}(data)$
$v_2 =$
θ
$v_1 =$
$v_i \in \mathbb{R}^p$
(C.5)
alignment $\mathcal{A}_i(x)$
v_1
v_2
0.5
0.0
−0.5
decision boundary of f_θ
x_L
x_R

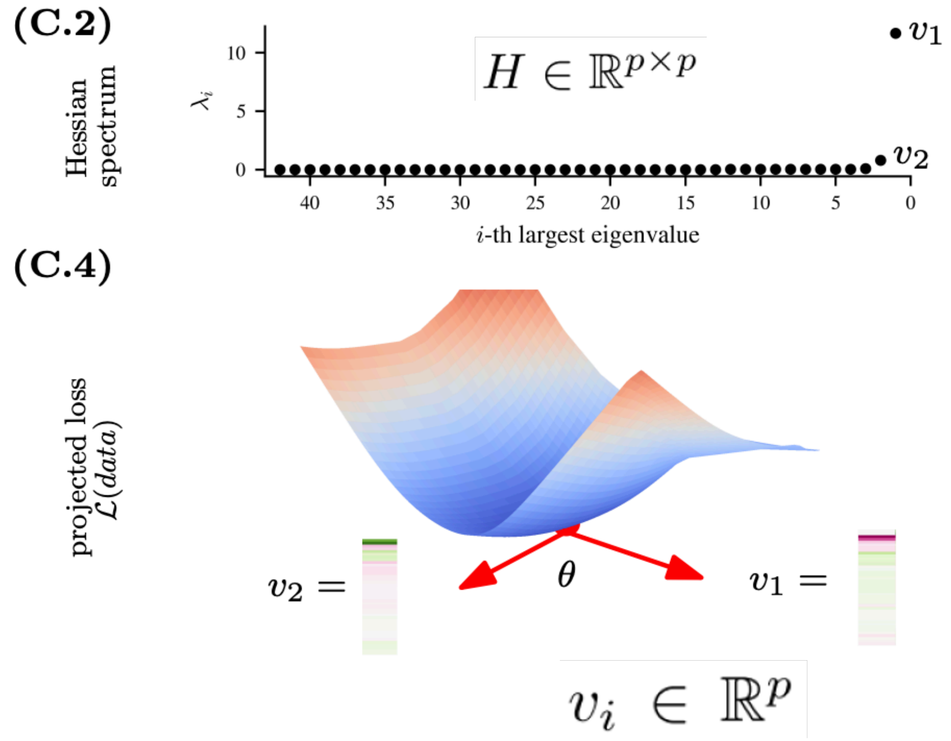
Toy example

Alignment between the *i*-th Hessian eigenvector and the reinforcing gradient:

$$\mathcal{A}_i(x) = \frac{\langle g_\theta(x), v_i \rangle}{\|g_\theta(x)\| \|v_i\|}$$



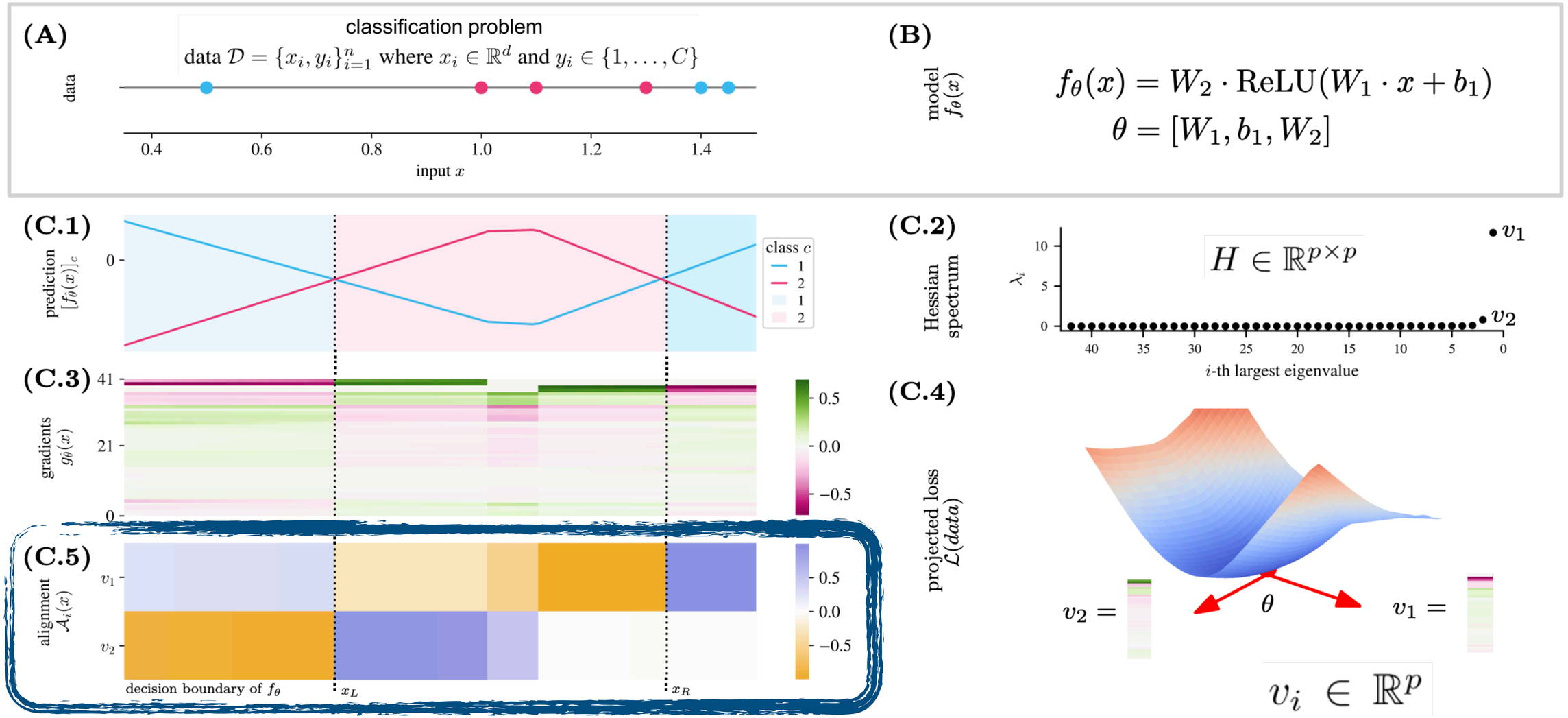




Toy example

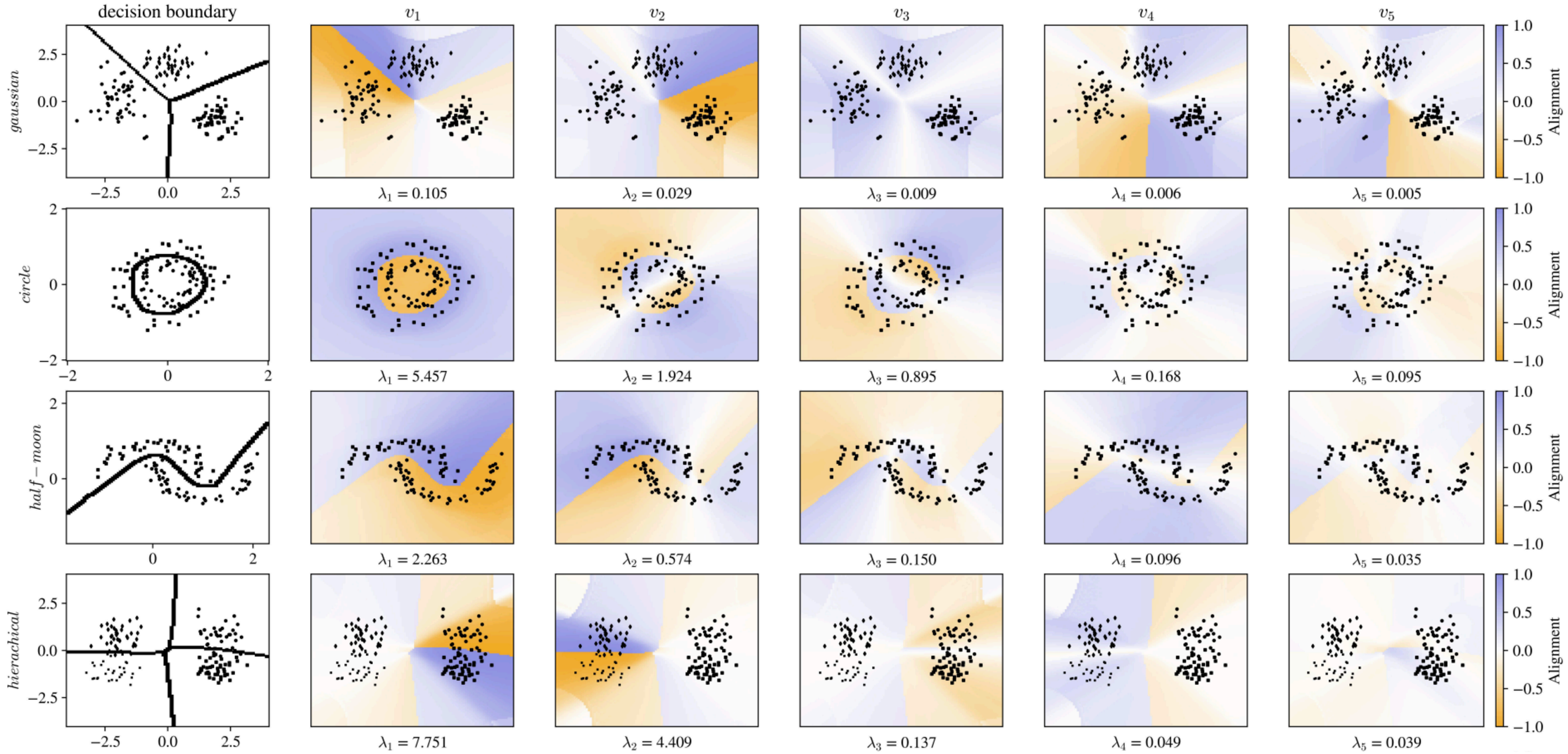
Alignment between the *i*-th Hessian eigenvector and the reinforcing gradient:

$$\mathcal{A}_i(x) = \frac{\langle g_\theta(x), v_i \rangle}{\|g_\theta(x)\| \|v_i\|}$$



2D datasets

The topmost eigenvectors have a close-to-one absolute alignment with gradients of loss of samples on the decision boundary of the network



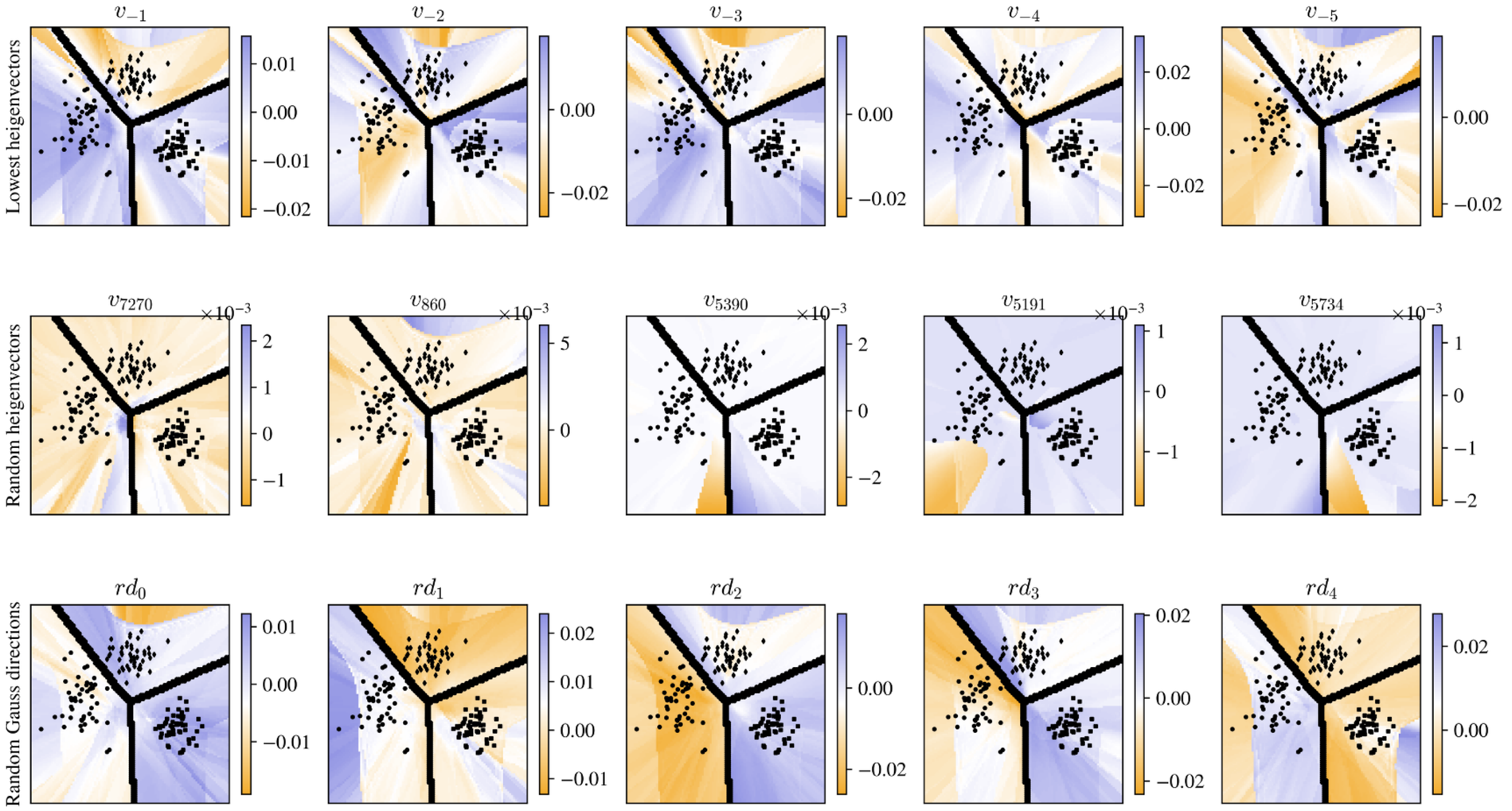
2D datasets

Not accidental! Compare with the alignment with…

... Lowest eigenvectors

... Random eigenvectors

... Random directions



Outline

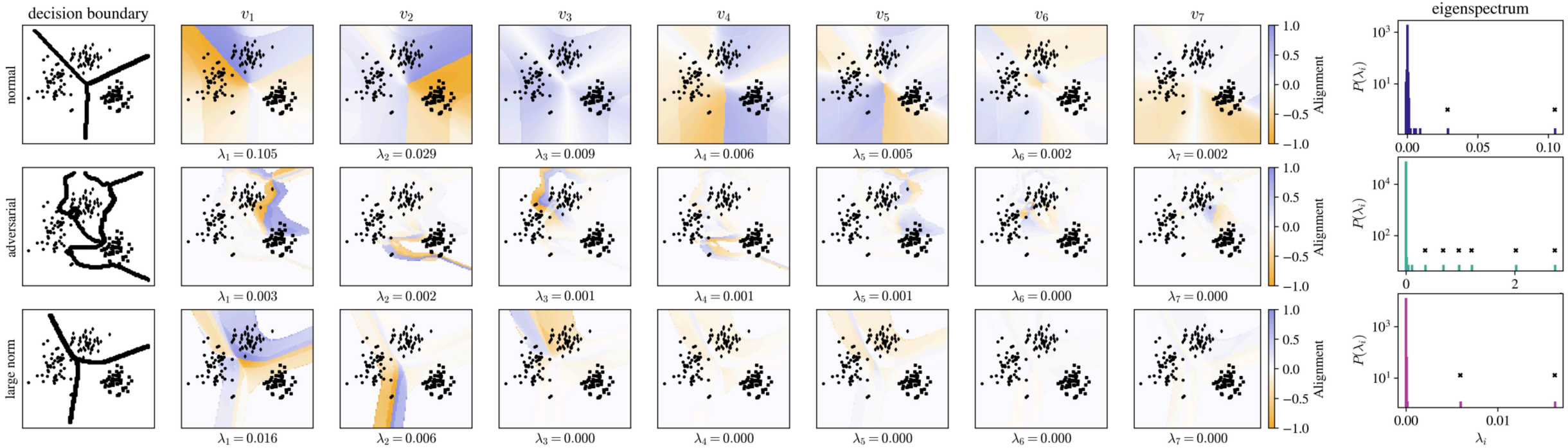
Empirical mess
in the Land of the Loss

Hessian and the decision boundary: toy example

Selected interesting consequences

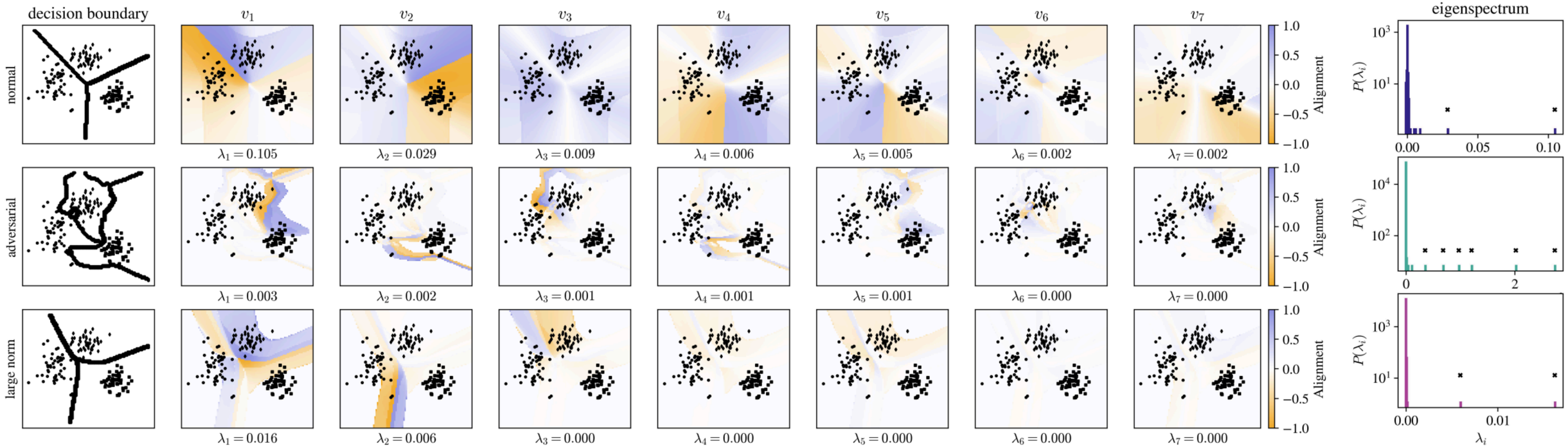
- measure of the decision boundary complexity
- perspectives for new generalization measure
- margin estimation

Decision boundary complexity



Decision boundary complexity

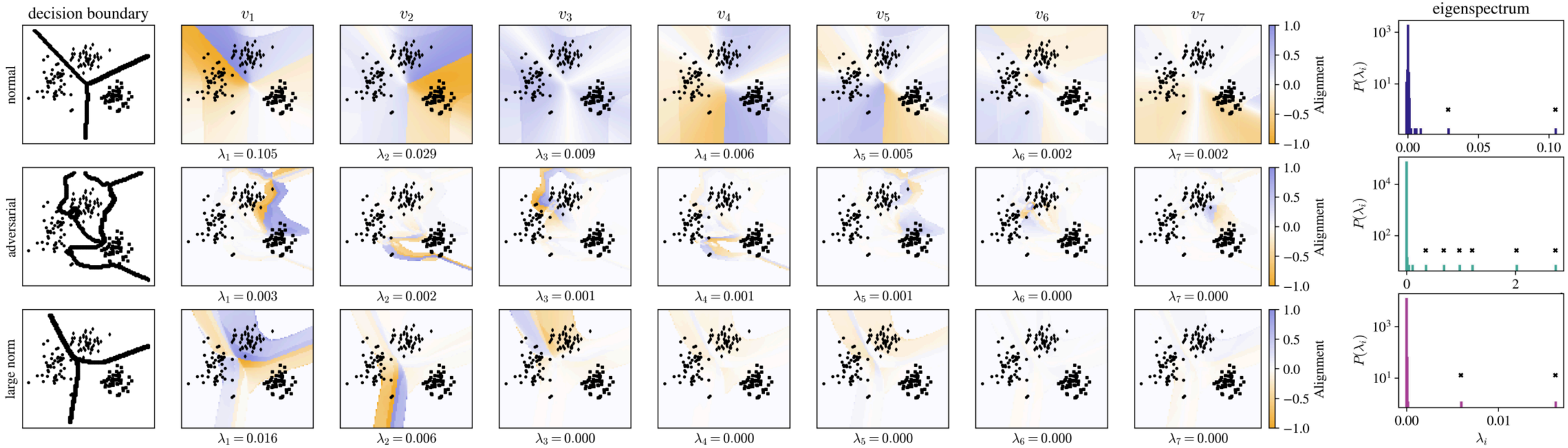
More eigenvectors are needed to describe the decision boundary in the adversarial initialization and large norm case than in the normal training.



Decision boundary complexity

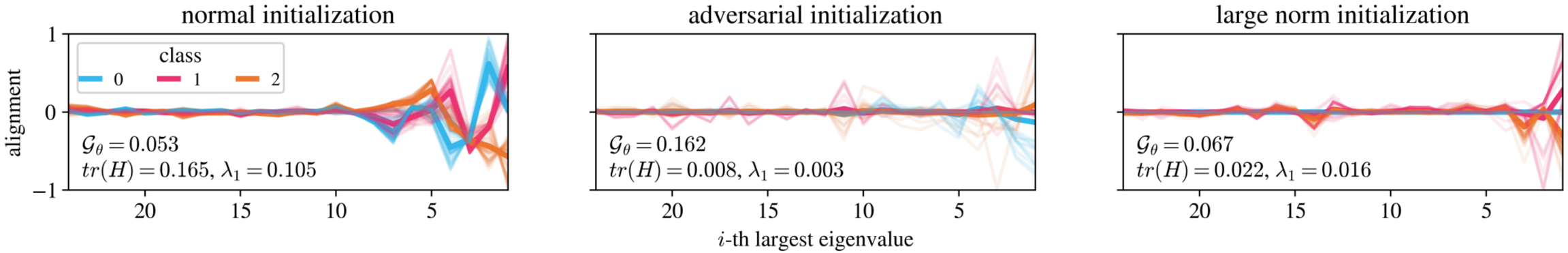
More eigenvectors are needed to describe the decision boundary in the adversarial initialization and large norm case than in the normal training.

Number of outliers in the spectrum depends on the decision boundary complexity (not only the number of classes)



Decision boundary complexity measure!

Alignment of gradients of loss of individual training samples with top Hessian eigenvectors behaves differently for the simple and complex decision boundary



We quantified this different behavior in an attempt to create a generalization measure

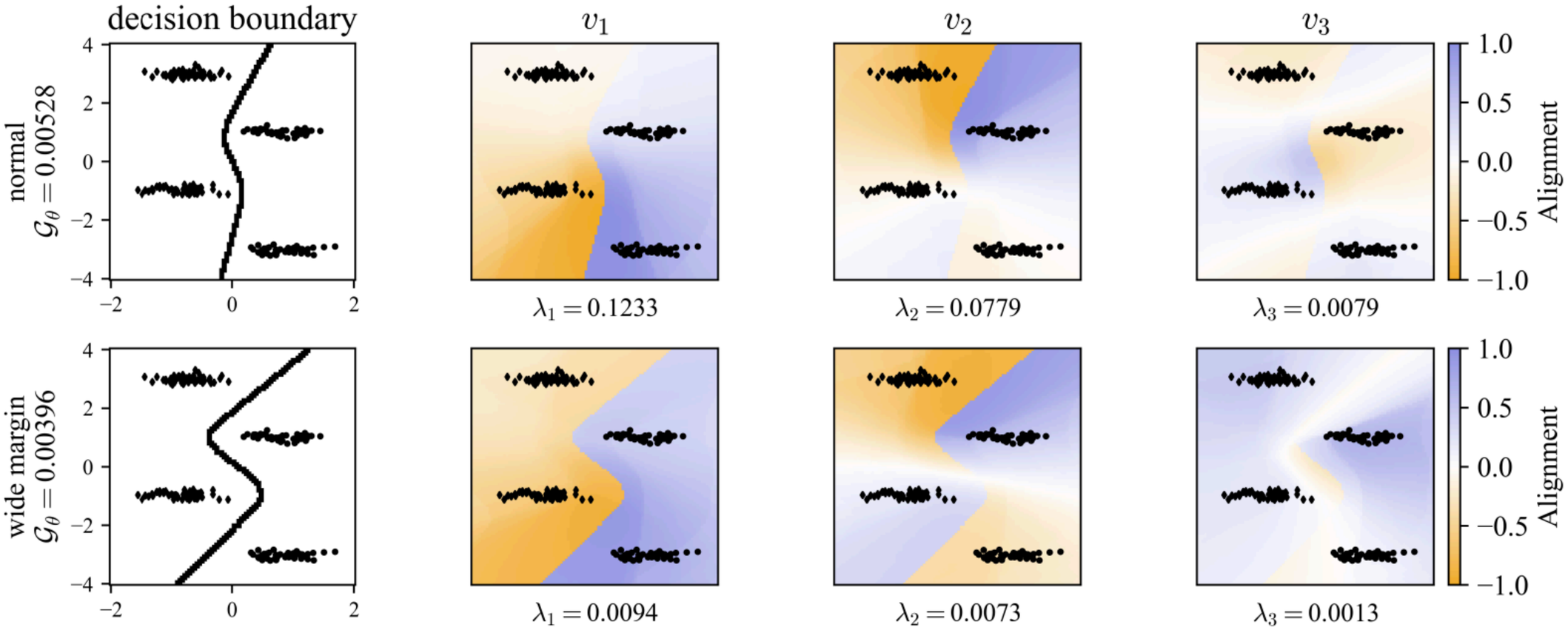
$$m_i = \frac{1}{n}\sum_{s=1}^{n} |\mathcal{A}_i(x_s)| \quad ; \quad \mathcal{G}_\theta = \frac{1}{p}\sum_{i=1}^{p} \mathbb{1}[m_i > \epsilon]$$

✓ Invariant to reparametrization and promising results for 2D datasets, Iris, and few-digit MNIST

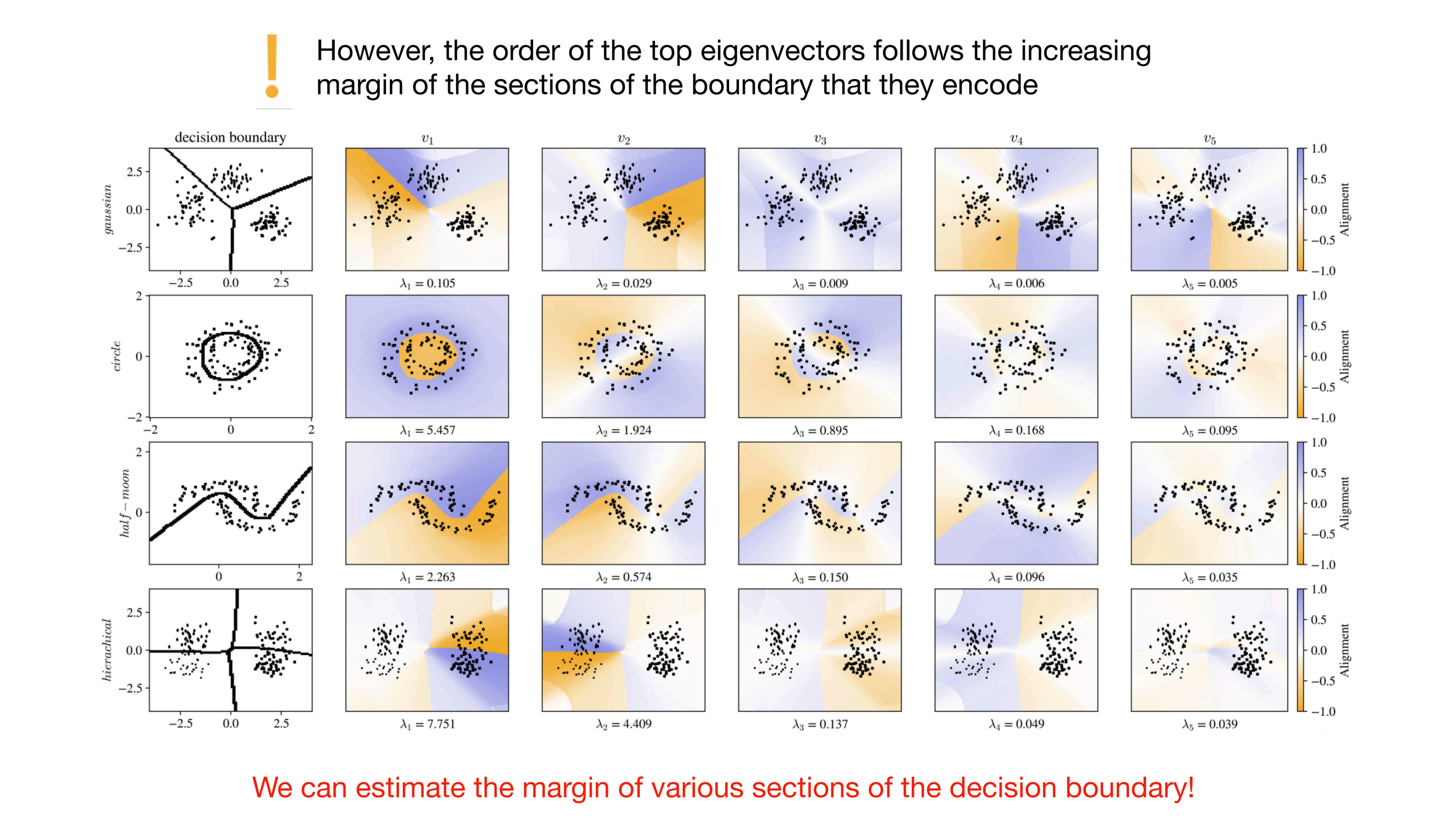
✗ Expensive to compute and…

Simplicity bias though…

Simplicity of the decision boundary is not everything!
There is also a margin!

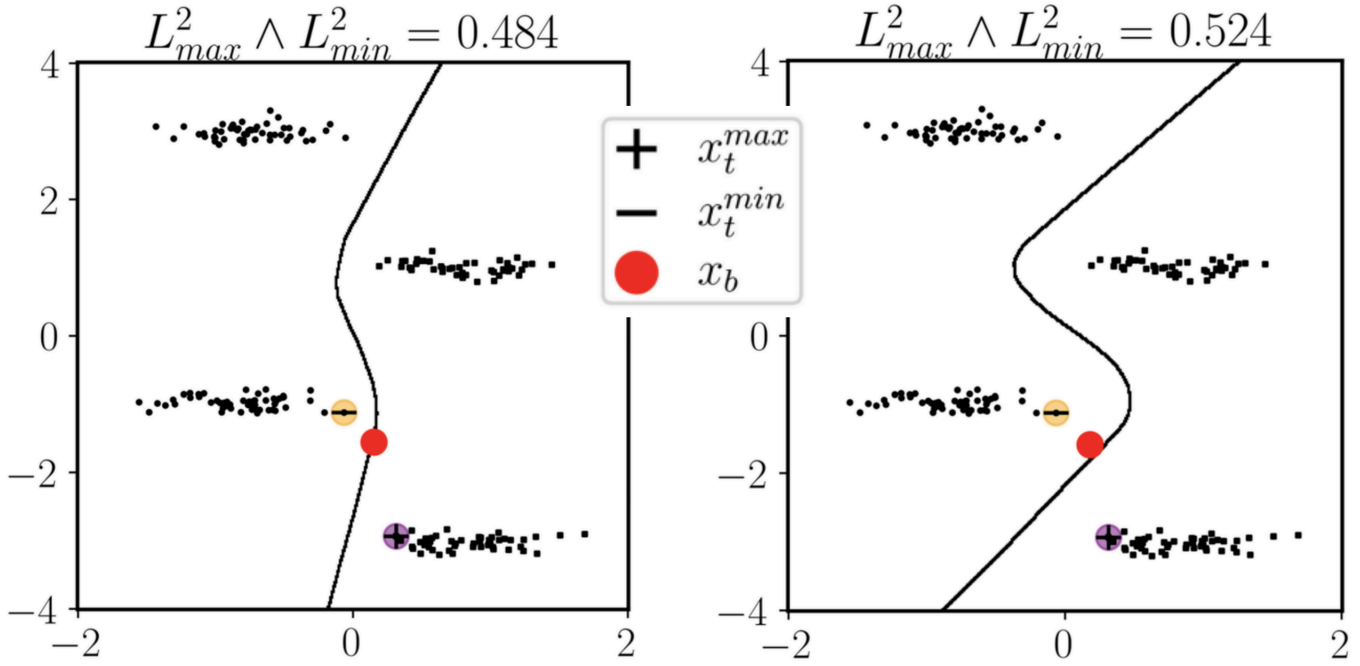


Complexity measure cannot detect the poorer generalization caused by the simplicity bias!


However, the order of the top eigenvectors follows the increasing margin of the sections of the boundary that they encode
decision boundary
v_1
v_2
v_3
v_4
v_5
gaussian
circle
half − moon
hierachical
$\lambda_1 = 0.105$
$\lambda_2 = 0.029$
$\lambda_3 = 0.009$
$\lambda_4 = 0.006$
$\lambda_5 = 0.005$
$\lambda_1 = 5.457$
$\lambda_2 = 1.924$
$\lambda_3 = 0.895$
$\lambda_4 = 0.168$
$\lambda_5 = 0.095$
$\lambda_1 = 2.263$
$\lambda_2 = 0.574$
$\lambda_3 = 0.150$
$\lambda_4 = 0.096$
$\lambda_5 = 0.035$
$\lambda_1 = 7.751$
$\lambda_2 = 4.409$
$\lambda_3 = 0.137$
$\lambda_4 = 0.049$
$\lambda_5 = 0.039$
Alignment
We can estimate the margin of various sections of the decision boundary!

Margin estimation

The top Hessian eigenvector allows to estimate the margin of the decision boundary!



(a) normal initialization

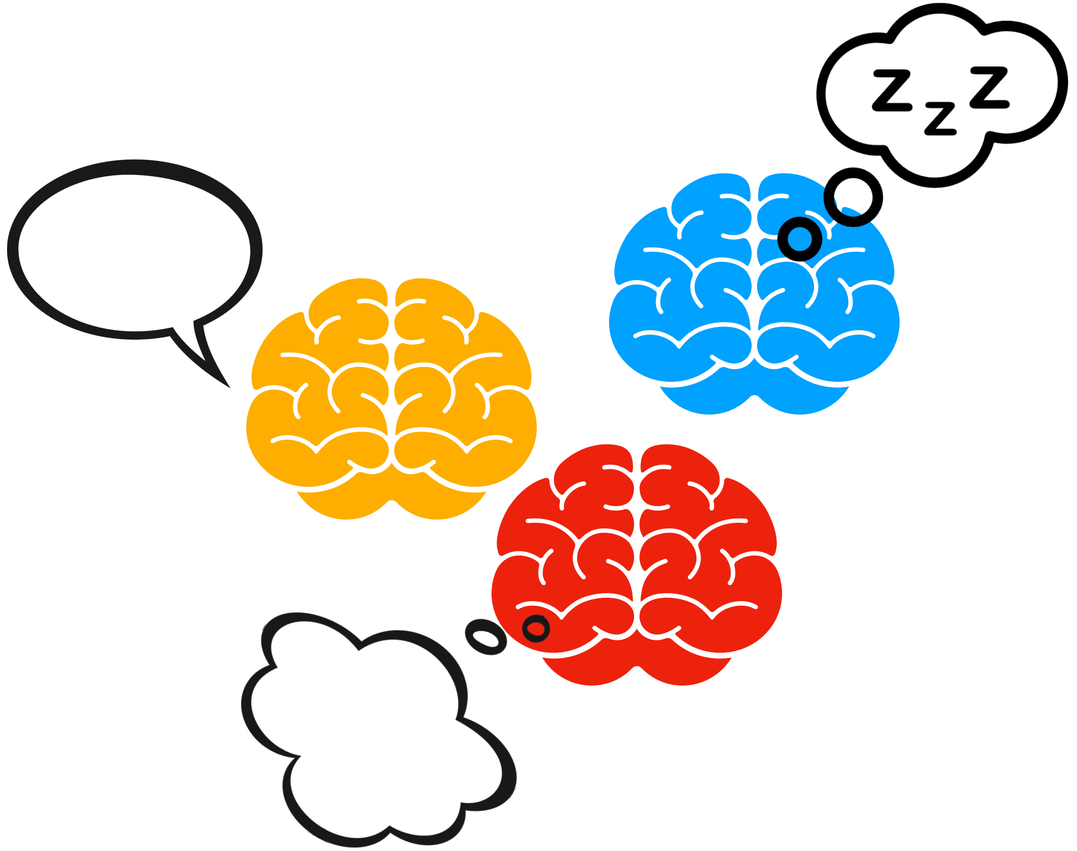
(b) wide-margin initialization

Training samples x_t that is the closest to the boundary in its narrowest part: they are **selected** as the one with the largest alignment with the top Hessian eigenvector v_1

A sample on the boundary x_b in its narrowest part: it is **optimized** to have the largest alignment with the top Hessian eigenvector v_1

Together, the decision boundary complexity measure and the margin estimation technique indicate a better generalizing model!

Open questions



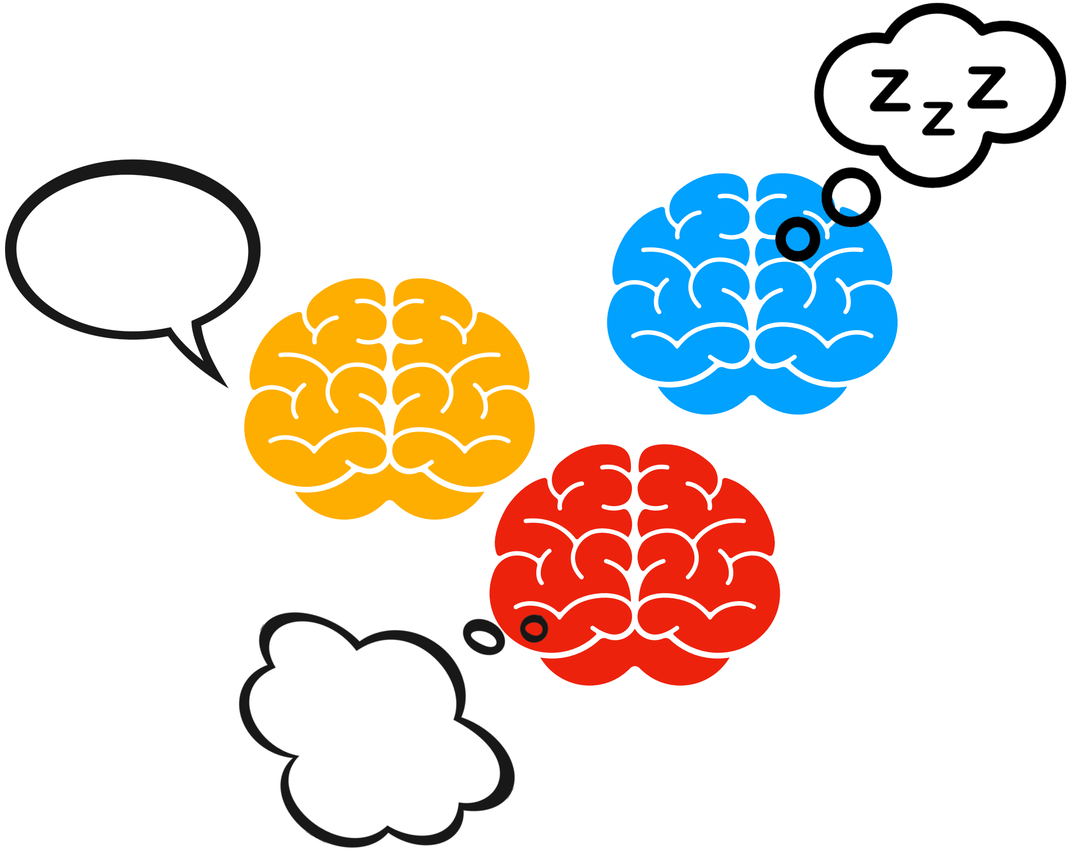
1. **Making the generalization measure useful**

2. **Theoretical connection**

3. **Higher dimensions**

- Does separation between eigenvectors/boundaries hold?
- Do we still need more eigenvectors for crooked high-dim boundaries?
- How to even attack it?

Cool new perspectives



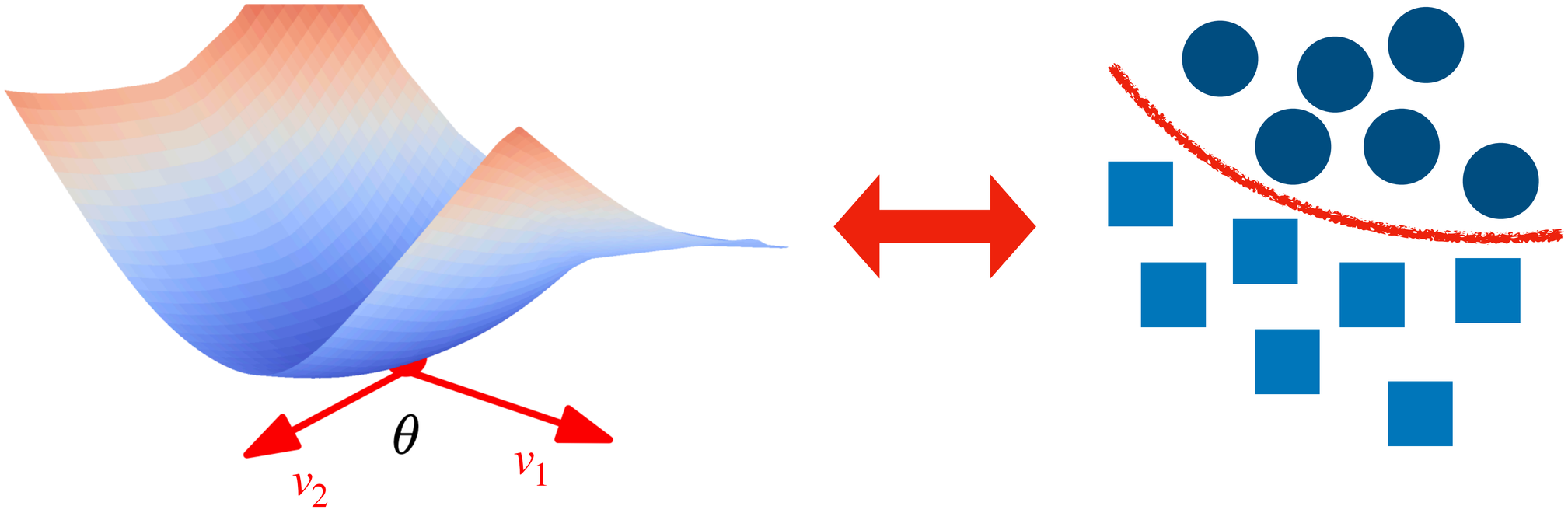
1. Can it tell us anything about
   the model uncertainty?

2. Margin estimation and robustness

3. Do decision boundaries
   of the adversarially trained models look the same
   as the regularly trained models?

Take-home message

The top Hessian eigenvectors encode the decision boundary



parameter space

input space

Amazing collaborators

Mahalakshmi Sabanayagam
TU Munich

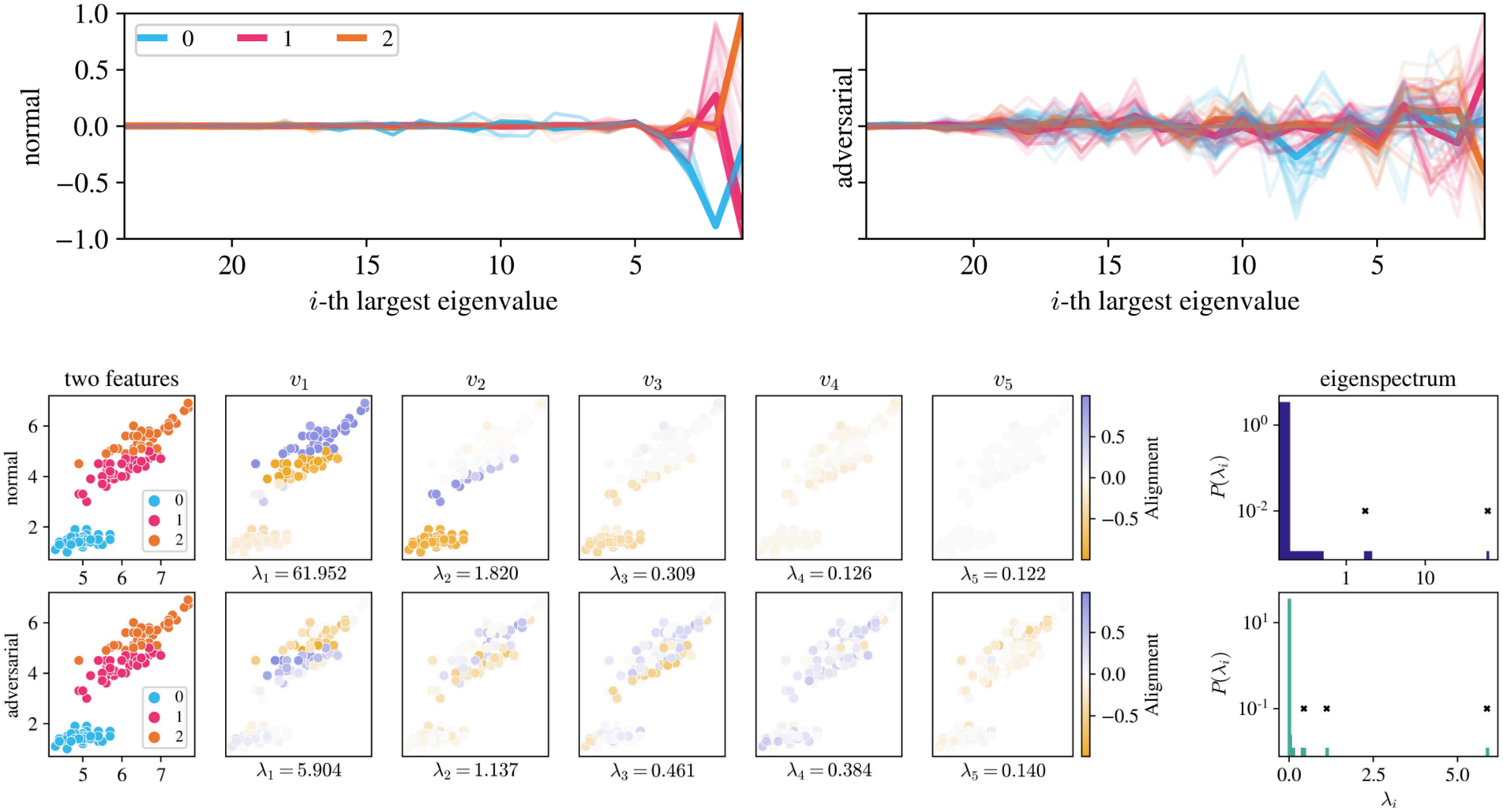
Freya Behrens
EPFL

Urte Adomaityte
King’s College London

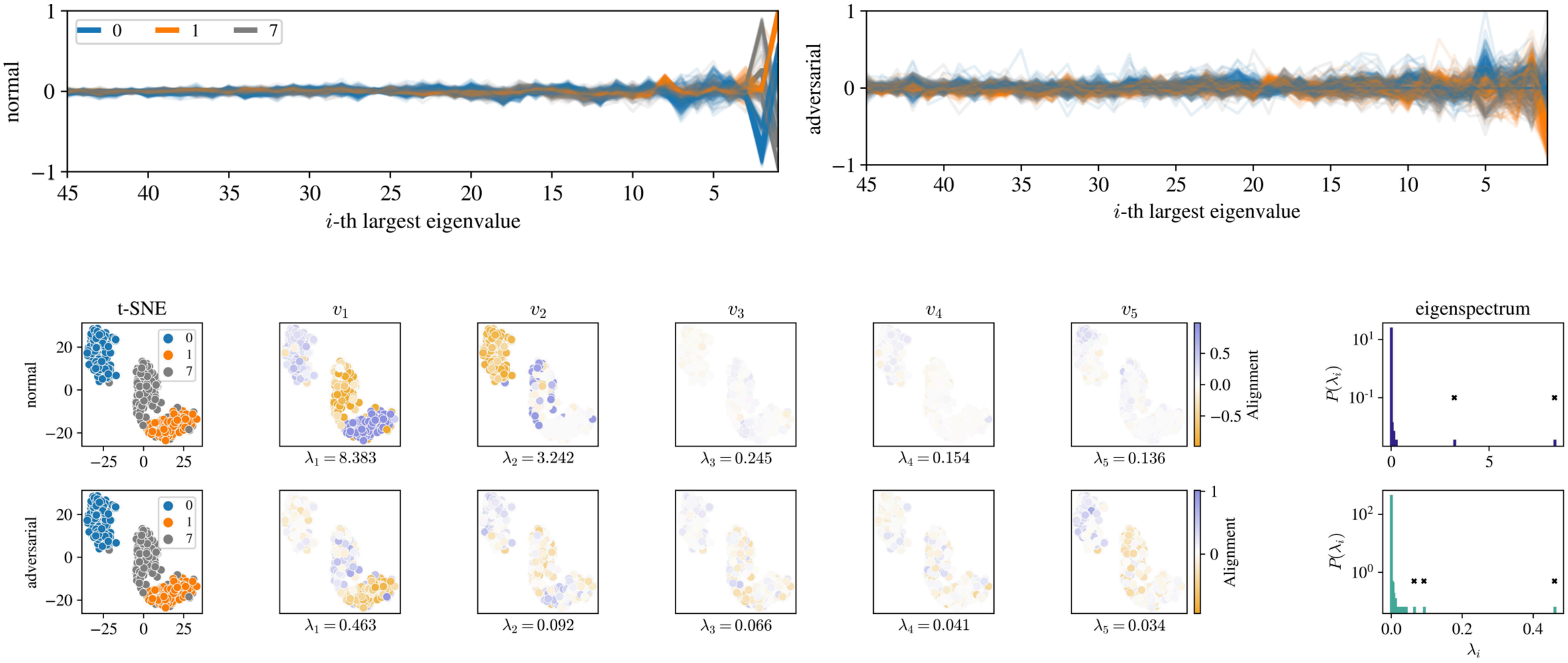
Thank you for your attention!

adawid@flatironinstitute.org

High(er)-dimensional results: Iris



High-dimensional results: MNIST-017



Mathematical analysis

We consider that the converged minimum $\theta := \theta^*$ is an exact minimum, meaning that the loss and its gradient at θ^* is zero, i.e., $\mathcal{L}(\theta^*; \mathcal{D}) = 0$ and $\nabla_\theta \mathcal{L}(\theta^*; \mathcal{D}) = \mathbf{0}$. Using this information, we expand the loss using the second-order Taylor's approximation at $\theta := \theta^*$.

$$\mathcal{L}(\theta^* + \Delta\theta; \mathcal{D}) = \mathcal{L}(\theta^*; \mathcal{D}) + \nabla_\theta \mathcal{L}(\theta^*; \mathcal{D})^T \Delta\theta + \frac{1}{2} \Delta\theta^T \nabla_\theta^2 \mathcal{L}(\theta^*; \mathcal{D}) \Delta\theta$$

$$\mathcal{L}(\theta^* + \Delta\theta; \mathcal{D}) = \frac{1}{2} \Delta\theta^T H_{\theta^*} \Delta\theta$$

H_{θ^*} is the Hessian of the training loss function evaluated at the minimum. We denote its eigenvectors and corresponding eigenvalues as v_i and λ_i. Now, let's consider $\Delta\theta := \frac{g_\theta(x)}{||g_\theta(x)||}$ to be a reinforcing gradient of some input x in the dataset $\mathcal{D} := \{\mathcal{X}, \mathcal{Y}\}$, and an overparametrized classifier (e.g., neural network) with p parameters denoted by $f(\theta, \mathcal{X})$.

$$\mathcal{L}\left(f\left(\theta^* + \frac{g_\theta(x)}{||g_\theta(x)||}, \mathcal{X}\right), \mathcal{Y}\right) = \frac{1}{2}\frac{g_\theta(x)^T}{||g_\theta(x)||}\sum_{i=1}^{p}\lambda_i v_i v_i^T \frac{g_\theta(x)}{||g_\theta(x)||}$$

$$= \frac{1}{2}\sum_{i=1}^{p}\lambda_i \frac{\langle g_\theta(x), v_i\rangle}{||g_\theta(x)||}\frac{\langle g_\theta(x), v_i\rangle}{||g_\theta(x)||}$$

$$\mathcal{L}\left(f\left(\theta^*, \mathcal{X}\right) + \nabla_\theta f\left(\theta^*, \mathcal{X}\right)^T \frac{g_\theta(x)}{||g_\theta(x)||}, \mathcal{Y}\right) = \frac{1}{2}\sum_{i=1}^{p}\lambda_i \mathcal{A}_i(x)^2$$

$$\mathcal{L}\left(\mathcal{Y} + \nabla_\theta f\left(\theta^*, \mathcal{X}\right)^T \frac{g_\theta(x)}{||g_\theta(x)||}, \mathcal{Y}\right) = \frac{1}{2}\sum_{i=1}^{p}\lambda_i \mathcal{A}_i(x)^2$$

maximal for:

- $\nabla_\theta f(\theta^*, \mathcal{X})$ that changes predictions of maximal number of samples (shifts the decision boundary)
- $g_\theta(x)$ aligned with this $\nabla_\theta f(\theta^*, \mathcal{X})$

maximal when

$g_\theta(x)$ and v_1 are aligned