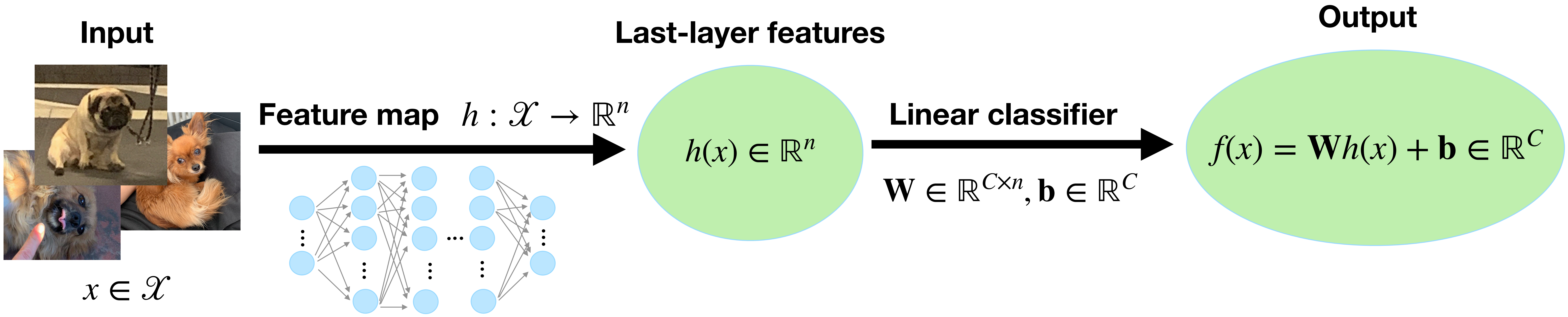
Neural (Tangent Kernel) Collapse

Mariia Seleznova[1], **Dana Weitzner**[2], **Raja Giryes**[2], **Gitta Kutyniok**[1], **Hung-Hsu Chou**[1]

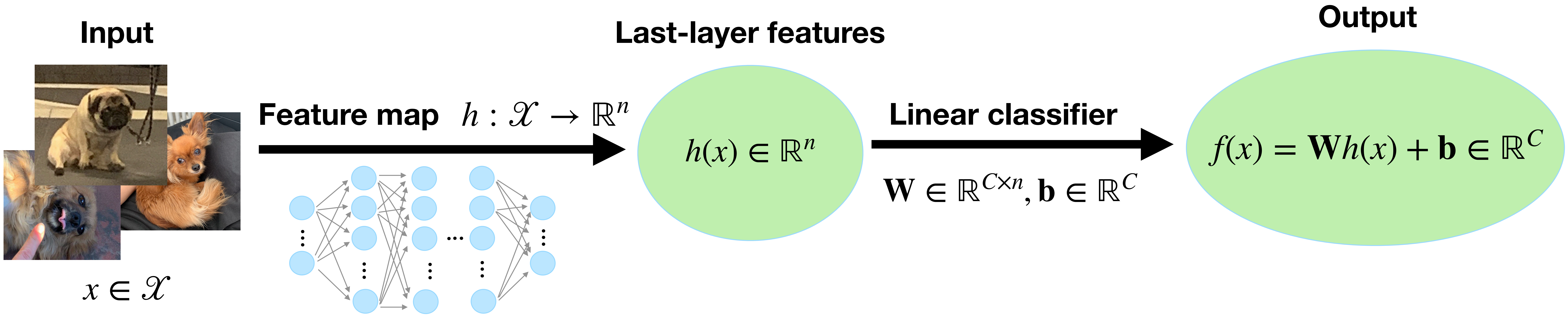
[1]*Ludwig-Maximilians-Universität München*, [2]*Tel Aviv University*

NeurIPS 2023

Setting: Deep Neural Network Classifiers

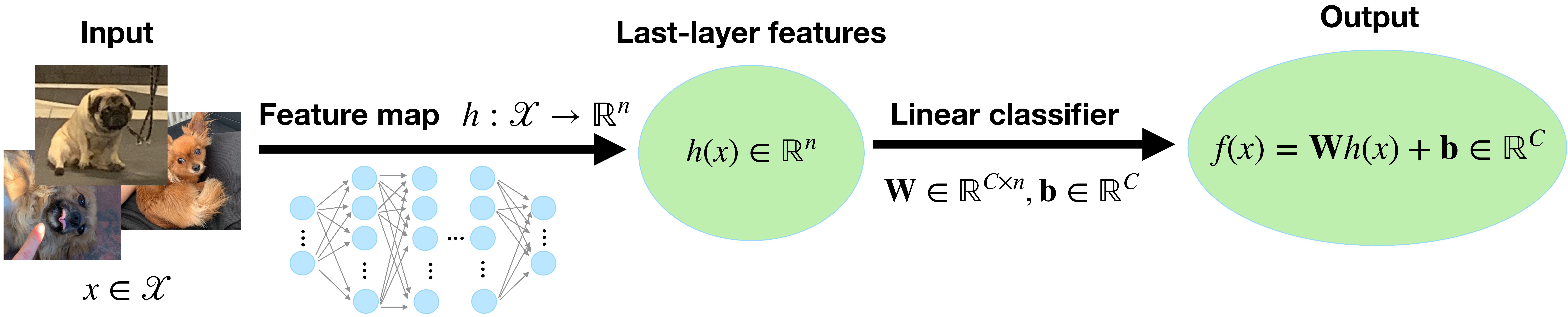


Setting: Deep Neural Network Classifiers



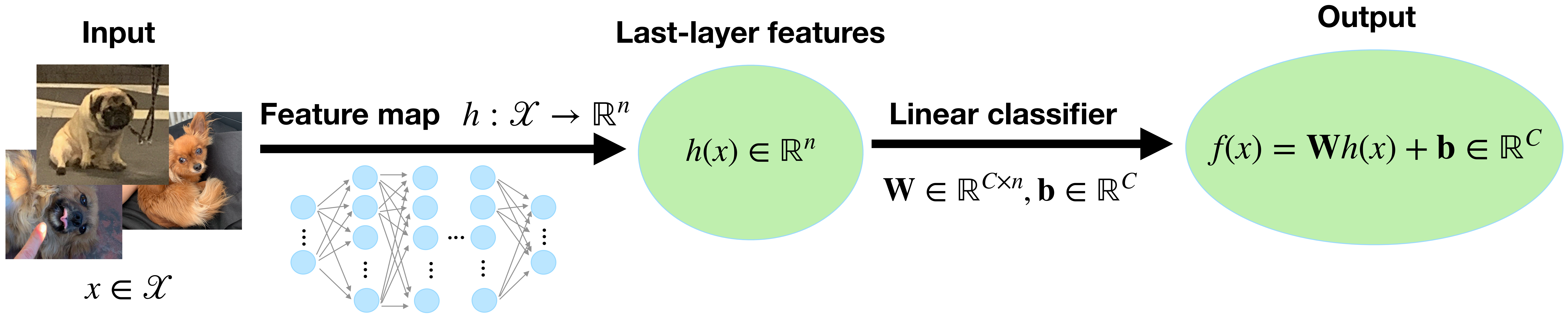
- Classification with *MSE loss*: $\mathscr{L}(\mathbf{X}, \mathbf{Y}) = \|\mathbf{W}\mathbf{H} + \mathbf{b}\mathbf{1}_N^\top - \mathbf{Y}\|_2^2$, where $\mathbf{H} = [h(x_1), \dots, h(x_N)] \in \mathbb{R}^{n \times N}$.

Setting: Deep Neural Network Classifiers



- Classification with *MSE loss*: $\mathscr{L}(\mathbf{X}, \mathbf{Y}) = \|\mathbf{W}\mathbf{H} + \mathbf{b}\mathbf{1}_N^\top - \mathbf{Y}\|_2^2$, where $\mathbf{H} = [h(x_1), \ldots, h(x_N)] \in \mathbb{R}^{n \times N}$.
- **Assumption 1:** The number of features is larger than the number of classes: $n > C$.

Setting: Deep Neural Network Classifiers



- Classification with *MSE loss*: $\mathscr{L}(\mathbf{X}, \mathbf{Y}) = \|\mathbf{W}\mathbf{H} + \mathbf{b}\mathbf{1}_N^\top - \mathbf{Y}\|_2^2$, where $\mathbf{H} = [h(x_1), \ldots, h(x_N)] \in \mathbb{R}^{n \times N}$.
- **Assumption 1:** The number of features is larger than the number of classes: $n > C$.
- **Assumption 2:** The dataset is *balanced*, i.e., there are $m := N/C$ samples from each class in the dataset.

Neural Collapse (NC)

Definition: NC is a common empirical phenomenon, which occurs in the *end of training* of modern classification DNNs:

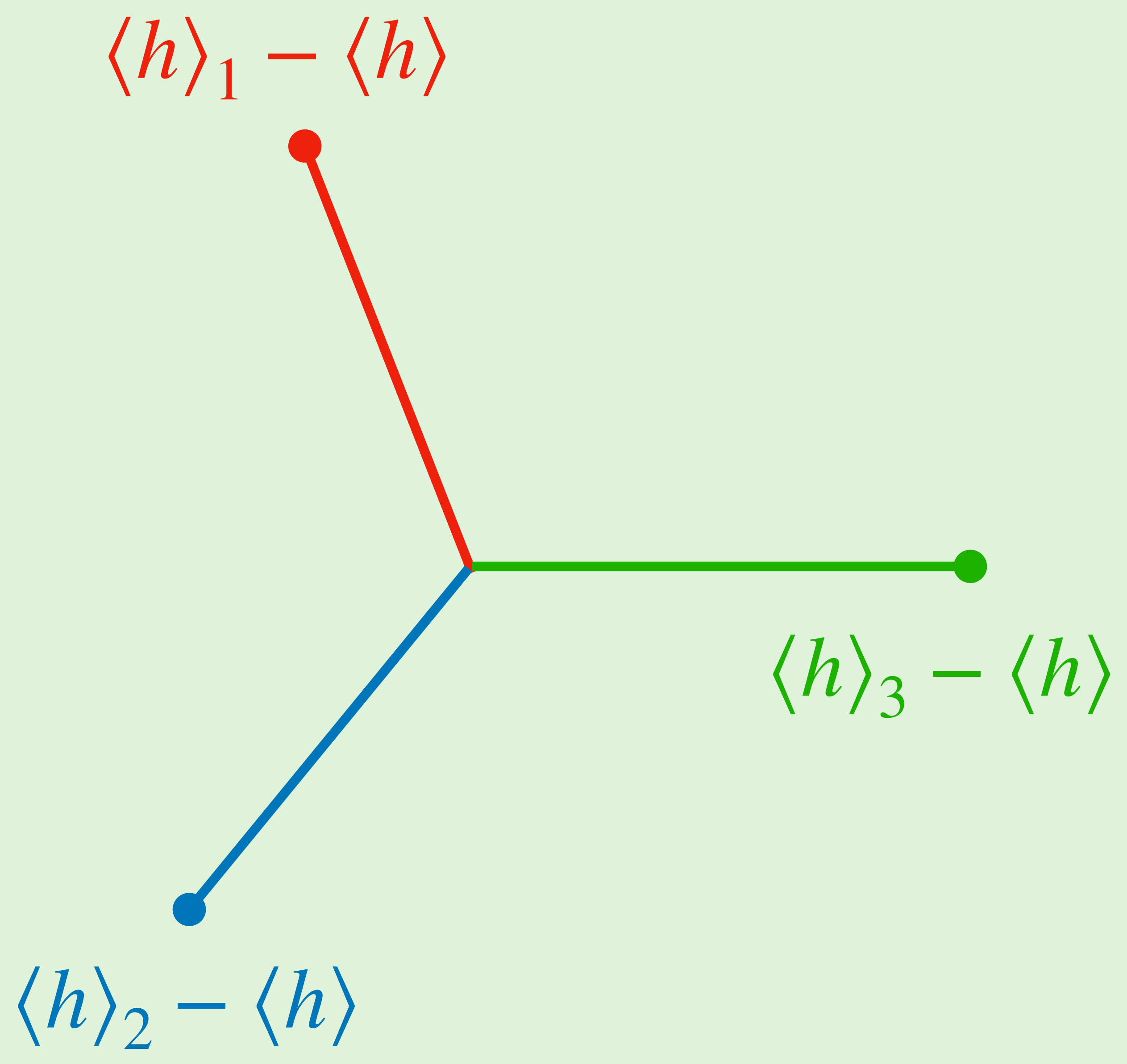
Papyan et al. **Prevalence of neural collapse during the terminal phase of deep learning training.** *Proceedings of the National Academy of Sciences, 2020.*

Neural Collapse (NC)

Definition: NC is a common empirical phenomenon, which occurs in the *end of training* of modern classification DNNs:

- **(NC1) *Variability collapse:*** features of each class collapse to their class means:

$$h(x_i^c) \to \langle h \rangle_c, \quad \forall c \in [C]$$



Papyan et al. **Prevalence of neural collapse during the terminal phase of deep learning training.** *Proceedings of the National Academy of Sciences, 2020.*

Neural Collapse (NC)

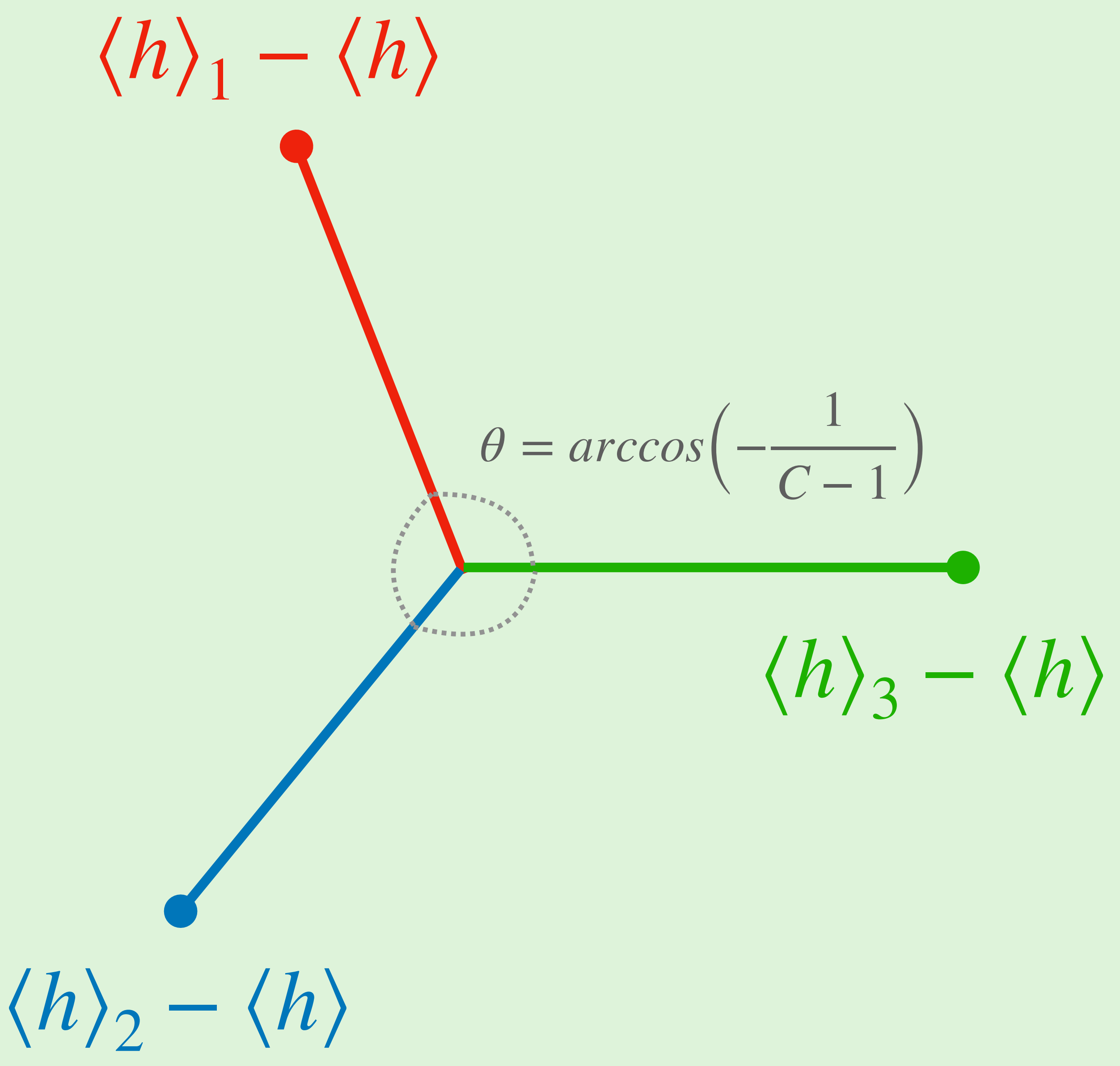
Definition: NC is a common empirical phenomenon, which occurs in the *end of training* of modern classification DNNs:

- **(NC1) *Variability collapse:*** features of each class collapse to their class means:

$$h(x_i^c) \rightarrow \langle h \rangle_c, \quad \forall c \in [C]$$

- **(NC2) *Convergence to Simplex Equiangular Tight Frame (ETF):*** centralized class means $\mathbf{M} = [\langle h \rangle_1 - \langle h \rangle, \ldots \langle h \rangle_C - \langle h \rangle]$ converge to the following configuration with *maximal separation angle*:

$$\mathbf{M}^\top \mathbf{M} \propto \frac{C}{C-1}\left(\mathbb{I}_C - \frac{1}{C}\mathbf{1}_C\mathbf{1}_C^\top\right)$$



Papyan et al. **Prevalence of neural collapse during the terminal phase of deep learning training.** *Proceedings of the National Academy of Sciences, 2020.*

Neural Collapse (NC)

Definition: NC is a common empirical phenomenon, which occurs in the *end of training* of modern classification DNNs:

- **(NC1) *Variability collapse:*** features of each class collapse to their class means:

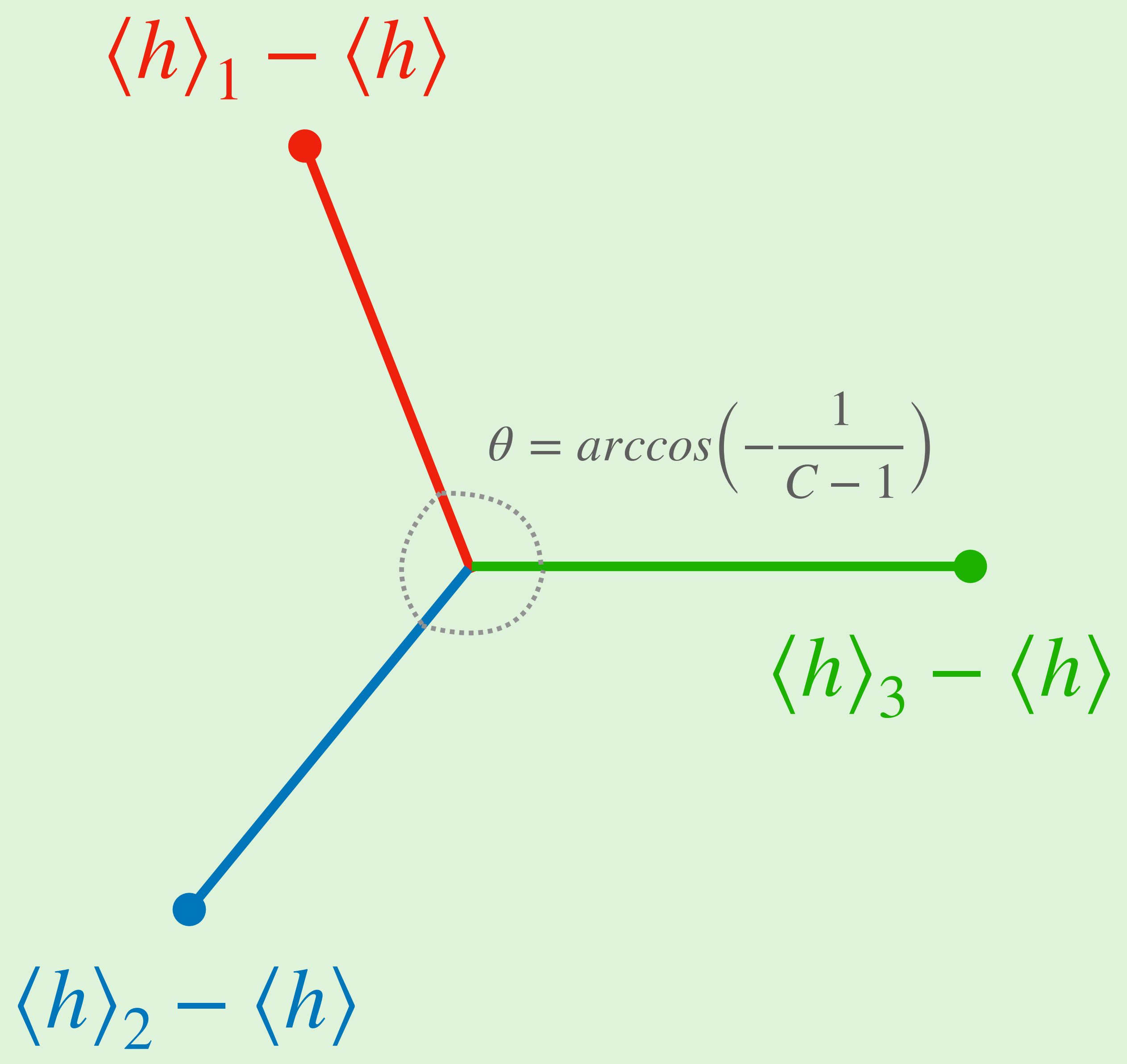
$$h(x_i^c) \to \langle h \rangle_c, \quad \forall c \in [C]$$

- **(NC2) *Convergence to Simplex Equiangular Tight Frame (ETF):*** centralized class means $\mathbf{M} = [\langle h \rangle_1 - \langle h \rangle, \ldots \langle h \rangle_C - \langle h \rangle]$ converge to the following configuration with *maximal separation angle*:

$$\mathbf{M}^\top \mathbf{M} \propto \frac{C}{C-1}\left(\mathbb{I}_C - \frac{1}{C}\mathbf{1}_C \mathbf{1}_C^\top\right)$$

- **(NC3) *Convergence to self-duality:*** the class means $\mathbf{M}$ and the final weights $\mathbf{W}^\top$ converge to each other:

$$\mathbf{M}/\|\mathbf{M}\| \to \mathbf{W}^\top/\|\mathbf{W}^\top\|$$



Papyan et al. **Prevalence of neural collapse during the terminal phase of deep learning training.** *Proceedings of the National Academy of Sciences, 2020.*

Can we explain NC theoretically?

Analyzing trained DNNs is challenging: complex non-linear training dynamics ⇝ theory relies on *simplifications*.

[1] Tirer & Bruna. **Extended unconstrained features model for exploring deep neural collapse.** *ICML, 2022.*

[2] Mixon et al. **Neural collapse with unconstrained features.** *CoRR, 2020.*

[3] Han et al. **Neural collapse under MSE loss: Proximity to and dynamics on the central path.** *ICLR, 2022.*

[4] Zhang et al. **Imitating deep learning dynamics via locally elastic stochastic differential equations.** *NeurIPS, 2021.*

[5] Poggio & Liao. **Explicit regularization and implicit bias in deep network classifiers trained with the square loss.** *CoRR, 2021.*

Can we explain NC theoretically?

Analyzing trained DNNs is challenging: complex non-linear training dynamics ⇝ theory relies on *simplifications*.

Previous works:

- *Unconstrained features model (UFM)*, where $\mathbf{H}$ is a free variable [1,2,3]

[1] Tirer & Bruna. **Extended unconstrained features model for exploring deep neural collapse.** *ICML, 2022.*

[2] Mixon et al. **Neural collapse with unconstrained features.** *CoRR, 2020.*

[3] Han et al. **Neural collapse under MSE loss: Proximity to and dynamics on the central path.** *ICLR, 2022.*

[4] Zhang et al. **Imitating deep learning dynamics via locally elastic stochastic differential equations.** *NeurIPS, 2021.*

[5] Poggio & Liao. **Explicit regularization and implicit bias in deep network classifiers trained with the square loss.** *CoRR, 2021.*

Can we explain NC theoretically?

Analyzing trained DNNs is challenging: complex non-linear training dynamics ⇝ theory relies on *simplifications*.

Previous works:

- *Unconstrained features model (UFM)*, where $\mathbf{H}$ is a free variable [1,2,3]
- Imitating DNNs dynamics with stochastic differential equations [4]

[1] Tirer & Bruna. **Extended unconstrained features model for exploring deep neural collapse.** *ICML, 2022.*

[2] Mixon et al. **Neural collapse with unconstrained features.** *CoRR, 2020.*

[3] Han et al. **Neural collapse under MSE loss: Proximity to and dynamics on the central path.** *ICLR, 2022.*

[4] Zhang et al. **Imitating deep learning dynamics via locally elastic stochastic differential equations.** *NeurIPS, 2021.*

[5] Poggio & Liao. **Explicit regularization and implicit bias in deep network classifiers trained with the square loss.** *CoRR, 2021.*

Can we explain NC theoretically?

Analyzing trained DNNs is challenging: complex non-linear training dynamics ⇝ theory relies on *simplifications*.

Previous works:

- *Unconstrained features model (UFM)*, where $\mathbf{H}$ is a free variable [1,2,3]
- Imitating DNNs dynamics with stochastic differential equations [4]
- Homogeneous DNNs [5]

[1] Tirer & Bruna. **Extended unconstrained features model for exploring deep neural collapse.** *ICML, 2022.*

[2] Mixon et al. **Neural collapse with unconstrained features.** *CoRR, 2020.*

[3] Han et al. **Neural collapse under MSE loss: Proximity to and dynamics on the central path.** *ICLR, 2022.*

[4] Zhang et al. **Imitating deep learning dynamics via locally elastic stochastic differential equations.** *NeurIPS, 2021.*

[5] Poggio & Liao. **Explicit regularization and implicit bias in deep network classifiers trained with the square loss.** *CoRR, 2021.*

Can we explain NC theoretically?

Analyzing trained DNNs is challenging: complex non-linear training dynamics ⇝ theory relies on *simplifications*.

Previous works:

- *Unconstrained features model (UFM)*, where $\mathbf{H}$ is a free variable [1,2,3]
- Imitating DNNs dynamics with stochastic differential equations [4]
- Homogeneous DNNs [5]

Our contributions:

- Novel approach to analyze DNNs' dynamics: *block-structure* assumption on the *Neural Tangent Kernel (NTK)*.

[1] Tirer & Bruna. **Extended unconstrained features model for exploring deep neural collapse.** *ICML, 2022.*

[2] Mixon et al. **Neural collapse with unconstrained features.** *CoRR, 2020.*

[3] Han et al. **Neural collapse under MSE loss: Proximity to and dynamics on the central path.** *ICLR, 2022.*

[4] Zhang et al. **Imitating deep learning dynamics via locally elastic stochastic differential equations.** *NeurIPS, 2021.*

[5] Poggio & Liao. **Explicit regularization and implicit bias in deep network classifiers trained with the square loss.** *CoRR, 2021.*

Can we explain NC theoretically?

Analyzing trained DNNs is challenging: complex non-linear training dynamics ⇝ theory relies on *simplifications*.

Previous works:

- *Unconstrained features model (UFM)*, where $\mathbf{H}$ is a free variable [1,2,3]
- Imitating DNNs dynamics with stochastic differential equations [4]
- Homogeneous DNNs [5]

Our contributions:

- Novel approach to analyze DNNs' dynamics: *block-structure* assumption on the *Neural Tangent Kernel (NTK)*.

  ⇝ *Makes gradient flow dynamics of DNNs tractable!*

[1] Tirer & Bruna. **Extended unconstrained features model for exploring deep neural collapse.** *ICML, 2022.*

[2] Mixon et al. **Neural collapse with unconstrained features.** *CoRR, 2020.*

[3] Han et al. **Neural collapse under MSE loss: Proximity to and dynamics on the central path.** *ICLR, 2022.*

[4] Zhang et al. **Imitating deep learning dynamics via locally elastic stochastic differential equations.** *NeurIPS, 2021.*

[5] Poggio & Liao. **Explicit regularization and implicit bias in deep network classifiers trained with the square loss.** *CoRR, 2021.*

Can we explain NC theoretically?

Analyzing trained DNNs is challenging: complex non-linear training dynamics ⇝ theory relies on *simplifications*.

Previous works:

- *Unconstrained features model (UFM)*, where $\mathbf{H}$ is a free variable [1,2,3]
- Imitating DNNs dynamics with stochastic differential equations [4]
- Homogeneous DNNs [5]

Our contributions:

- Novel approach to analyze DNNs' dynamics: *block-structure* assumption on the *Neural Tangent Kernel (NTK)*.
  - ⇝ *Makes gradient flow dynamics of DNNs tractable!*
  - ⇝ *The NTK captures the dependence of the features on the input data (missing in UFMs).*

[1] Tirer & Bruna. **Extended unconstrained features model for exploring deep neural collapse.** *ICML, 2022.*

[2] Mixon et al. **Neural collapse with unconstrained features.** *CoRR, 2020.*

[3] Han et al. **Neural collapse under MSE loss: Proximity to and dynamics on the central path.** *ICLR, 2022.*

[4] Zhang et al. **Imitating deep learning dynamics via locally elastic stochastic differential equations.** *NeurIPS, 2021.*

[5] Poggio & Liao. **Explicit regularization and implicit bias in deep network classifiers trained with the square loss.** *CoRR, 2021.*

Can we explain NC theoretically?

Analyzing trained DNNs is challenging: complex non-linear training dynamics ⇝ theory relies on *simplifications*.

Previous works:

- *Unconstrained features model (UFM)*, where $\mathbf{H}$ is a free variable [1,2,3]
- Imitating DNNs dynamics with stochastic differential equations [4]
- Homogeneous DNNs [5]

Our contributions:

- Novel approach to analyze DNNs' dynamics: *block-structure* assumption on the *Neural Tangent Kernel (NTK)*.
  ⇝ *Makes gradient flow dynamics of DNNs tractable!*
  ⇝ *The NTK captures the dependence of the features on the input data (missing in UFMs).*
- Show that *NC occurs in fixed points of gradient flow dynamics* under additional assumptions.

[1] Tirer & Bruna. **Extended unconstrained features model for exploring deep neural collapse.** *ICML, 2022.*

[2] Mixon et al. **Neural collapse with unconstrained features.** *CoRR, 2020.*

[3] Han et al. **Neural collapse under MSE loss: Proximity to and dynamics on the central path.** *ICLR, 2022.*

[4] Zhang et al. **Imitating deep learning dynamics via locally elastic stochastic differential equations.** *NeurIPS, 2021.*

[5] Poggio & Liao. **Explicit regularization and implicit bias in deep network classifiers trained with the square loss.** *CoRR, 2021.*

Can we explain NC theoretically?

Analyzing trained DNNs is challenging: complex non-linear training dynamics ⇝ theory relies on *simplifications*.

Previous works:

- *Unconstrained features model (UFM)*, where $\mathbf{H}$ is a free variable [1,2,3]
- Imitating DNNs dynamics with stochastic differential equations [4]
- Homogeneous DNNs [5]

Our contributions:

- Novel approach to analyze DNNs' dynamics: *block-structure* assumption on the *Neural Tangent Kernel (NTK)*.
  - ⇝ *Makes gradient flow dynamics of DNNs tractable!*
  - ⇝ *The NTK captures the dependence of the features on the input data (missing in UFMs).*
- Show that *NC occurs in fixed points of gradient flow dynamics* under additional assumptions.
- Discuss *necessary conditions* for convergence to NC.

[1] Tirer & Bruna. **Extended unconstrained features model for exploring deep neural collapse.** *ICML, 2022.*

[2] Mixon et al. **Neural collapse with unconstrained features.** *CoRR, 2020.*

[3] Han et al. **Neural collapse under MSE loss: Proximity to and dynamics on the central path.** *ICLR, 2022.*

[4] Zhang et al. **Imitating deep learning dynamics via locally elastic stochastic differential equations.** *NeurIPS, 2021.*

[5] Poggio & Liao. **Explicit regularization and implicit bias in deep network classifiers trained with the square loss.** *CoRR, 2021.*

Neural Tangent Kernel (NTK)

Definition: **Neural tangent kernel** $\Theta : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^{C \times C}$ of a DNN with output function $f : \mathcal{X} \to \mathbb{R}^C$ and trainable parameters $\mathbf{w}$ is given by

$$\Theta_{k,s}(x_i, x_j) := \left\langle \nabla_{\mathbf{w}} f_k(x_i), \nabla_{\mathbf{w}} f_s(x_j) \right\rangle, \quad x_i, x_j \in \mathcal{X}, \quad k, s \in [C]$$

Definition: **Last-layer features kernel** $\Theta^h : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^{n \times n}$ of a DNN with last-layer features $h : \mathcal{X} \to \mathbb{R}^n$ and trainable parameters $\mathbf{w}$ is given by

$$\Theta^h_{k,s}(x_i, x_j) := \left\langle \nabla_{\mathbf{w}} h_k(x_i), \nabla_{\mathbf{w}} h_s(x_j) \right\rangle, \quad x_i, x_j \in \mathcal{X}, \quad k, s \in [n]$$

[1] Baratin et al. **Implicit regularization via neural feature alignment.** *AISTATS*, 2021.

[2] Shan & Bordelon. **A theory of neural tangent kernel alignment and its influence on training.** CoRR, 2020.

Neural Tangent Kernel (NTK)

Definition: **Neural tangent kernel** $\Theta : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^{C \times C}$ of a DNN with output function $f : \mathcal{X} \to \mathbb{R}^C$ and trainable parameters $\mathbf{w}$ is given by

$$\Theta_{k,s}(x_i, x_j) := \left\langle \nabla_{\mathbf{w}} f_k(x_i), \nabla_{\mathbf{w}} f_s(x_j) \right\rangle, \quad x_i, x_j \in \mathcal{X}, \quad k, s \in [C]$$

Definition: **Last-layer features kernel** $\Theta^h : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^{n \times n}$ of a DNN with last-layer features $h : \mathcal{X} \to \mathbb{R}^n$ and trainable parameters $\mathbf{w}$ is given by

$$\Theta^h_{k,s}(x_i, x_j) := \left\langle \nabla_{\mathbf{w}} h_k(x_i), \nabla_{\mathbf{w}} h_s(x_j) \right\rangle, \quad x_i, x_j \in \mathcal{X}, \quad k, s \in [n]$$

- Intuitively, the NTK captures *correlations between input samples* during training.

[1] Baratin et al. **Implicit regularization via neural feature alignment.** *AISTATS*, 2021.
[2] Shan & Bordelon. **A theory of neural tangent kernel alignment and its influence on training.** CoRR, 2020.

Neural Tangent Kernel (NTK)

Definition: **Neural tangent kernel** $\Theta : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^{C \times C}$ of a DNN with output function $f : \mathcal{X} \to \mathbb{R}^C$ and trainable parameters $\mathbf{w}$ is given by

$$\Theta_{k,s}(x_i, x_j) := \left\langle \nabla_{\mathbf{w}} f_k(x_i), \nabla_{\mathbf{w}} f_s(x_j) \right\rangle, \quad x_i, x_j \in \mathcal{X}, \quad k, s \in [C]$$

Definition: **Last-layer features kernel** $\Theta^h : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^{n \times n}$ of a DNN with last-layer features $h : \mathcal{X} \to \mathbb{R}^n$ and trainable parameters $\mathbf{w}$ is given by

$$\Theta^h_{k,s}(x_i, x_j) := \left\langle \nabla_{\mathbf{w}} h_k(x_i), \nabla_{\mathbf{w}} h_s(x_j) \right\rangle, \quad x_i, x_j \in \mathcal{X}, \quad k, s \in [n]$$

- Intuitively, the NTK captures *correlations between input samples* during training.
- **NTK alignment:** during training, «correlations» $\Theta(x_i, x_j)$ become larger for samples (x_i, x_j) from the same class [1,2]

[1] Baratin et al. **Implicit regularization via neural feature alignment.** *AISTATS*, 2021.
[2] Shan & Bordelon. **A theory of neural tangent kernel alignment and its influence on training.** CoRR, 2020.

Neural Tangent Kernel (NTK)

Definition: Neural tangent kernel $\Theta : \mathscr{X} \times \mathscr{X} \to \mathbb{R}^{C \times C}$ of a DNN with output function $f : \mathscr{X} \to \mathbb{R}^C$ and trainable parameters $\mathbf{w}$ is given by

$$\Theta_{k,s}(x_i, x_j) := \left\langle \nabla_{\mathbf{w}} f_k(x_i), \nabla_{\mathbf{w}} f_s(x_j) \right\rangle, \quad x_i, x_j \in \mathscr{X}, \quad k, s \in [C]$$

Definition: Last-layer features kernel $\Theta^h : \mathscr{X} \times \mathscr{X} \to \mathbb{R}^{n \times n}$ of a DNN with last-layer features $h : \mathscr{X} \to \mathbb{R}^n$ and trainable parameters $\mathbf{w}$ is given by

$$\Theta^h_{k,s}(x_i, x_j) := \left\langle \nabla_{\mathbf{w}} h_k(x_i), \nabla_{\mathbf{w}} h_s(x_j) \right\rangle, \quad x_i, x_j \in \mathscr{X}, \quad k, s \in [n]$$

- Intuitively, the NTK captures *correlations between input samples* during training.
- **NTK alignment:** during training, «correlations» $\Theta(x_i, x_j)$ become larger for samples (x_i, x_j) from the same class [1,2]

⇝ The NTK develops an (approximate) block structure during training of DNN classifiers!

[1] Baratin et al. **Implicit regularization via neural feature alignment.** *AISTATS*, 2021.

[2] Shan & Bordelon. **A theory of neural tangent kernel alignment and its influence on training.** CoRR, 2020.

Block-Structure of the NTK

Definition. We say a kernel $\Theta : \mathscr{X} \times \mathscr{X} \to \mathbb{R}^{K \times K}$ has a block structure with values $(\lambda_1, \lambda_2, \lambda_3)$ s.t. $\lambda_1 > \lambda_2 > \lambda_3 > 0$ if

$$\Theta_{k,k}(x, x) = \lambda_1, \quad \Theta_{k,k}(x_i^c, x_j^c) = \lambda_2, \quad \Theta_{k,k}(x_i^c, x_j^{c'}) = \lambda_3, \quad k = [1,K],$$

and $\Theta_{k,s}(x, \tilde{x}) = 0$ for $k \neq s$.

Assumption (NTK block structure). The NTK $\Theta : \mathscr{X} \times \mathscr{X} \to \mathbb{R}^{C \times C}$ has a block structure with values $(\gamma_d, \gamma_c, \gamma_n)$, and the last-layer features kernel $\Theta^h : \mathscr{X} \times \mathscr{X} \to \mathbb{R}^{n \times n}$ has a block structure with values $(\kappa_d, \kappa_c, \kappa_n)$.

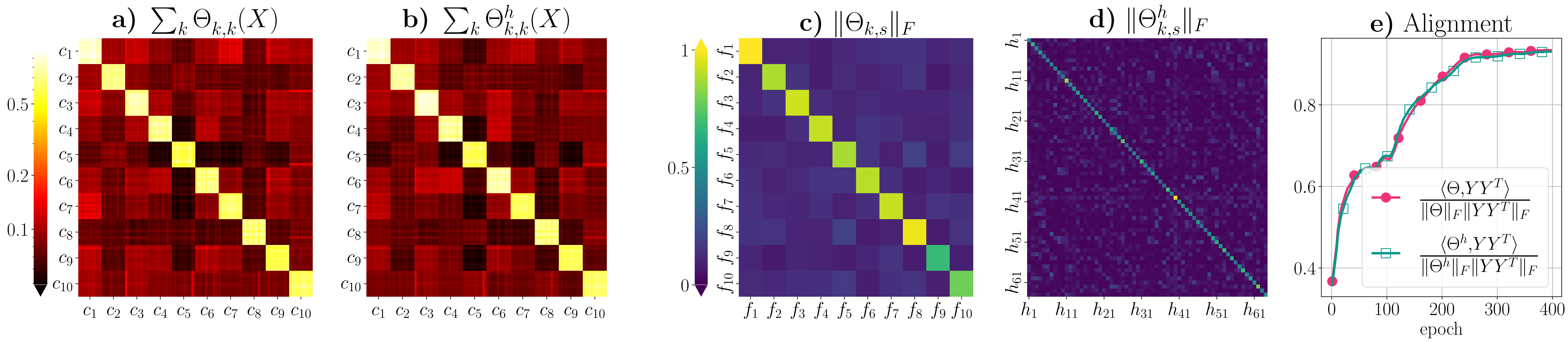


Figure: The NTK block structure of ResNet20 trained on MNIST.

Block-Structure of the NTK

Definition. We say a kernel $\Theta : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^{K \times K}$ has a block structure with values $(\lambda_1, \lambda_2, \lambda_3)$ s.t. $\lambda_1 > \lambda_2 > \lambda_3 > 0$ if

$$\Theta_{k,k}(x, x) = \lambda_1, \quad \Theta_{k,k}(x_i^c, x_j^c) = \lambda_2, \quad \Theta_{k,k}(x_i^c, x_j^{c'}) = \lambda_3, \quad k = [1,K],$$

and $\Theta_{k,s}(x, \tilde{x}) = 0$ for $k \neq s$.

Inputs from the same class

Assumption (NTK block structure)**.** The NTK $\Theta : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^{C \times C}$ has a block structure with values $(\gamma_d, \gamma_c, \gamma_n)$, and the last-layer features kernel $\Theta^h : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^{n \times n}$ has a block structure with values $(\kappa_d, \kappa_c, \kappa_n)$.

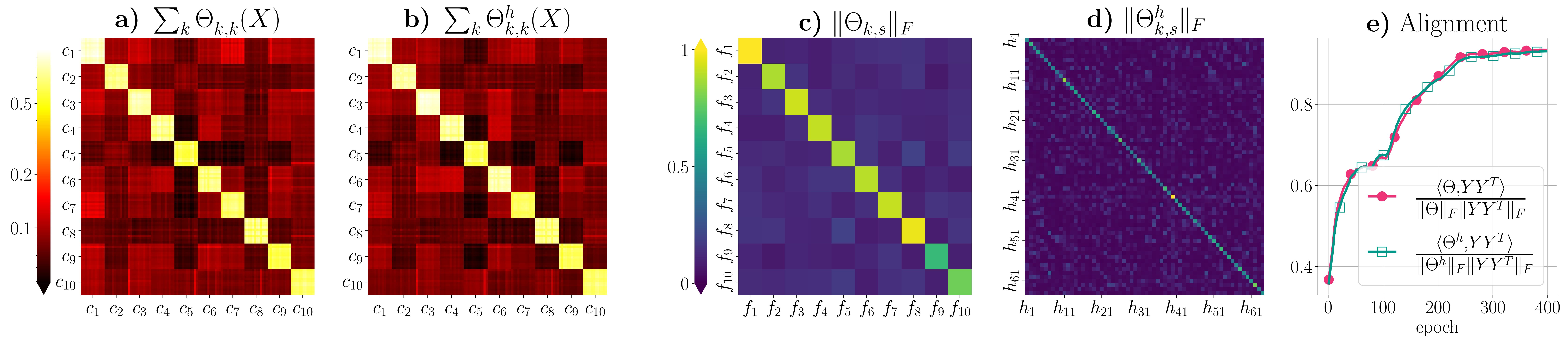


Figure: The NTK block structure of ResNet20 trained on MNIST.

Block-Structure of the NTK

Definition. We say a kernel $\Theta : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^{K \times K}$ has a block structure with values $(\lambda_1, \lambda_2, \lambda_3)$ s.t. $\lambda_1 > \lambda_2 > \lambda_3 > 0$ if

$$\Theta_{k,k}(x, x) = \lambda_1, \quad \Theta_{k,k}(x_i^c, x_j^c) = \lambda_2, \quad \Theta_{k,k}(x_i^c, x_j^{c'}) = \lambda_3, \quad k = [1,K],$$

and $\Theta_{k,s}(x, \tilde{x}) = 0$ for $k \neq s$.

Inputs from the same class — Inputs from different classes

Assumption (NTK block structure). The NTK $\Theta : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^{C \times C}$ has a block structure with values $(\gamma_d, \gamma_c, \gamma_n)$, and the last-layer features kernel $\Theta^h : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^{n \times n}$ has a block structure with values $(\kappa_d, \kappa_c, \kappa_n)$.

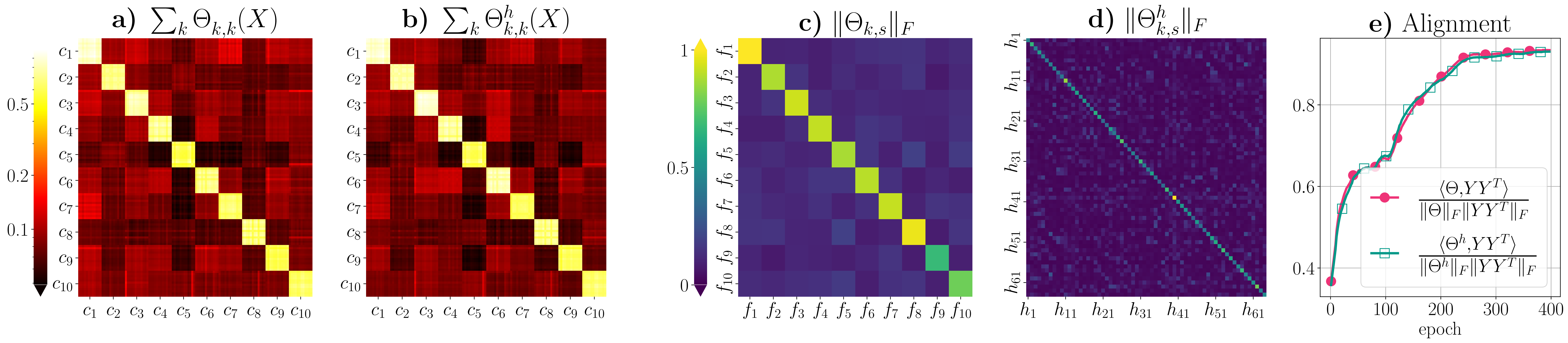


Figure: The NTK block structure of ResNet20 trained on MNIST.

Gradient Flow with Block-Structured NTK

Theorem. Suppose the NTK block structure assumption holds. Then the *gradient flow dynamics* of a DNN is given by

$$\begin{cases} \dot{\mathbf{H}} = & -\mathbf{W}^\top[(\kappa_d - \kappa_c)\mathbf{R} + (\kappa_c - \kappa_n)m\mathbf{R}_{class} + \kappa_n N \mathbf{R}_{global}] \\ \dot{\mathbf{W}} = & -\mathbf{R}\mathbf{H}^\top \\ \dot{\mathbf{b}} = & -\mathbf{R}_{global}\mathbf{1}_N, \end{cases}$$

where we defined the following residual components:

$$\mathbf{R} = f(\mathbf{X}) - \mathbf{Y}, \qquad \mathbf{R}_{class} = \underbrace{[\langle \mathbf{r} \rangle_1, \ldots, \langle \mathbf{r} \rangle_C]}_{:=\mathbf{R}_1} \otimes \mathbf{1}_m^\top, \qquad \mathbf{R}_{global} = \langle \mathbf{r} \rangle \otimes \mathbf{1}_N^\top.$$

Gradient Flow with Block-Structured NTK

Theorem. Suppose the NTK block structure assumption holds. Then the *gradient flow dynamics* of a DNN is given by

$$\begin{cases} \dot{\mathbf{H}} = & -\mathbf{W}^\top[(\kappa_d - \kappa_c)\mathbf{R} + (\kappa_c - \kappa_n)m\mathbf{R}_{class} + \kappa_n N \mathbf{R}_{global}] \\ \dot{\mathbf{W}} = & -\mathbf{R}\mathbf{H}^\top \\ \dot{\mathbf{b}} = & -\mathbf{R}_{global}\mathbf{1}_N, \end{cases}$$

where we defined the following residual components:

Class-means of the residuals

$$\mathbf{R} = f(\mathbf{X}) - \mathbf{Y}, \qquad \mathbf{R}_{class} = \underbrace{[\langle \mathbf{r} \rangle_1, \ldots, \langle \mathbf{r} \rangle_C]}_{:=\mathbf{R}_1} \otimes \mathbf{1}_m^\top, \qquad \mathbf{R}_{global} = \langle \mathbf{r} \rangle \otimes \mathbf{1}_N^\top.$$

Gradient Flow with Block-Structured NTK

Theorem. Suppose the NTK block structure assumption holds. Then the *gradient flow dynamics* of a DNN is given by

$$\begin{cases} \dot{\mathbf{H}} = & -\mathbf{W}^\top[(\kappa_d - \kappa_c)\mathbf{R} + (\kappa_c - \kappa_n)m\mathbf{R}_{class} + \kappa_n N \mathbf{R}_{global}] \\ \dot{\mathbf{W}} = & -\mathbf{R}\mathbf{H}^\top \\ \dot{\mathbf{b}} = & -\mathbf{R}_{global}\mathbf{1}_N, \end{cases}$$

where we defined the following residual components:

Class-means of the residuals

Global mean of the residuals

$$\mathbf{R} = f(\mathbf{X}) - \mathbf{Y}, \qquad \mathbf{R}_{class} = \underbrace{[\langle \mathbf{r} \rangle_1, \ldots, \langle \mathbf{r} \rangle_C]}_{:=\mathbf{R}_1} \otimes \mathbf{1}_m^\top, \qquad \mathbf{R}_{global} = \langle \mathbf{r} \rangle \otimes \mathbf{1}_N^\top.$$

Gradient Flow with Block-Structured NTK

Theorem. Suppose the NTK block structure assumption holds. Then the *gradient flow dynamics* of a DNN is given by

$$\begin{cases} \dot{\mathbf{H}} = & -\mathbf{W}^\top[(\kappa_d - \kappa_c)\mathbf{R} + (\kappa_c - \kappa_n)m\mathbf{R}_{class} + \kappa_n N \mathbf{R}_{global}] \\ \dot{\mathbf{W}} = & -\mathbf{R}\mathbf{H}^\top \\ \dot{\mathbf{b}} = & -\mathbf{R}_{global}\mathbf{1}_N, \end{cases}$$

where we defined the following residual components:

Class-means of the residuals

Global mean of the residuals

$$\mathbf{R} = f(\mathbf{X}) - \mathbf{Y}, \qquad \mathbf{R}_{class} = \underbrace{[\langle \mathbf{r} \rangle_1, \ldots, \langle \mathbf{r} \rangle_C]}_{:=\mathbf{R}_1} \otimes \mathbf{1}_m^\top, \qquad \mathbf{R}_{global} = \langle \mathbf{r} \rangle \otimes \mathbf{1}_N^\top.$$

Theorem. The following quantity is *invariant in time:*

$$\mathbf{E} := \frac{1}{m}\mathbf{W}^\top\mathbf{W} - \frac{1}{\mu_{class}}\mathbf{H}_1\mathbf{H}_1^\top - \frac{1}{\mu_{single}}\mathbf{H}_2\mathbf{H}_2^\top + \frac{\alpha}{\mu_{class}}\langle h \rangle \langle h \rangle^\top,$$

where $[\mathbf{H}_1, \mathbf{H}_2] := \mathbf{H}\mathbf{Q}/\sqrt{m}$ for a certain orthogonal matrix $\mathbf{Q}$, and $\alpha, \mu_{class}, \mu_{single}$ are some positive constants.

Gradient Flow with Block-Structured NTK

Theorem. Suppose the NTK block structure assumption holds. Then the *gradient flow dynamics* of a DNN is given by

$$\begin{cases} \dot{\mathbf{H}} = & -\mathbf{W}^\top[(\kappa_d - \kappa_c)\mathbf{R} + (\kappa_c - \kappa_n)m\mathbf{R}_{class} + \kappa_n N \mathbf{R}_{global}] \\ \dot{\mathbf{W}} = & -\mathbf{R}\mathbf{H}^\top \\ \dot{\mathbf{b}} = & -\mathbf{R}_{global}\mathbf{1}_N, \end{cases}$$

where we defined the following residual components:

Class-means of the residuals

Global mean of the residuals

$$\mathbf{R} = f(\mathbf{X}) - \mathbf{Y}, \qquad \mathbf{R}_{class} = \underbrace{[\langle \mathbf{r} \rangle_1, \ldots, \langle \mathbf{r} \rangle_C]}_{:=\mathbf{R}_1} \otimes \mathbf{1}_m^\top, \qquad \mathbf{R}_{global} = \langle \mathbf{r} \rangle \otimes \mathbf{1}_N^\top.$$

Theorem. The following quantity is *invariant in time:*

Class-means

$$\mathbf{E} := \frac{1}{m}\mathbf{W}^\top\mathbf{W} - \frac{1}{\mu_{class}}\mathbf{H}_1\mathbf{H}_1^\top - \frac{1}{\mu_{single}}\mathbf{H}_2\mathbf{H}_2^\top + \frac{\alpha}{\mu_{class}}\langle h \rangle \langle h \rangle^\top,$$

where $[\mathbf{H}_1, \mathbf{H}_2] := \mathbf{H}\mathbf{Q}/\sqrt{m}$ for a certain orthogonal matrix $\mathbf{Q}$, and $\alpha, \mu_{class}, \mu_{single}$ are some positive constants.

Gradient Flow with Block-Structured NTK

Theorem. Suppose the NTK block structure assumption holds. Then the *gradient flow dynamics* of a DNN is given by

$$\begin{cases} \dot{\mathbf{H}} = & -\mathbf{W}^\top[(\kappa_d - \kappa_c)\mathbf{R} + (\kappa_c - \kappa_n)m\mathbf{R}_{class} + \kappa_n N \mathbf{R}_{global}] \\ \dot{\mathbf{W}} = & -\mathbf{R}\mathbf{H}^\top \\ \dot{\mathbf{b}} = & -\mathbf{R}_{global}\mathbf{1}_N, \end{cases}$$

where we defined the following residual components:

Class-means of the residuals

Global mean of the residuals

$$\mathbf{R} = f(\mathbf{X}) - \mathbf{Y}, \qquad \mathbf{R}_{class} = \underbrace{[\langle \mathbf{r} \rangle_1, \ldots, \langle \mathbf{r} \rangle_C]}_{:=\mathbf{R}_1} \otimes \mathbf{1}_m^\top, \qquad \mathbf{R}_{global} = \langle \mathbf{r} \rangle \otimes \mathbf{1}_N^\top.$$

Theorem. The following quantity is *invariant in time:*

Class-means

«Variability» within classes

$$\mathbf{E} := \frac{1}{m}\mathbf{W}^\top\mathbf{W} - \frac{1}{\mu_{class}}\mathbf{H}_1\mathbf{H}_1^\top - \frac{1}{\mu_{single}}\mathbf{H}_2\mathbf{H}_2^\top + \frac{\alpha}{\mu_{class}}\langle h \rangle \langle h \rangle^\top,$$

where $[\mathbf{H}_1, \mathbf{H}_2] := \mathbf{H}\mathbf{Q}/\sqrt{m}$ for a certain orthogonal matrix $\mathbf{Q}$, and $\alpha, \mu_{class}, \mu_{single}$ are some positive constants.

Neural Collapse with Block-Structured NTK

Theorem. Assume that the NTK block structure assumption holds. Assume further that the last-layer features are *centralized*, i.e, $\langle h \rangle = \bar{0}$, and the gradient flow dynamics invariant is zero, i.e., $\mathbf{E} = \mathbb{O}$. Then the DNN's dynamic exhibits neural collapse as defined in **(NC1)-(NC3)**.

Neural Collapse with Block-Structured NTK

Theorem. Assume that the NTK block structure assumption holds. Assume further that the last-layer features are *centralized*, i.e, $\langle h \rangle = \bar{0}$, and the gradient flow dynamics invariant is zero, i.e., $\mathbf{E} = \mathbb{O}$. Then the DNN's dynamic exhibits neural collapse as defined in **(NC1)-(NC3)**.

- Zero invariant assumption $\mathbf{E} = \mathbb{O}$ has similar effects to joint *regularization* of $\mathbf{W}$ and $\mathbf{H}$.

Neural Collapse with Block-Structured NTK

Theorem. Assume that the NTK block structure assumption holds. Assume further that the last-layer features are *centralized*, i.e, $\langle h \rangle = \bar{0}$, and the gradient flow dynamics invariant is zero, i.e., $\mathbf{E} = \mathbb{O}$. Then the DNN's dynamic exhibits neural collapse as defined in **(NC1)-(NC3)**.

- Zero invariant assumption $\mathbf{E} = \mathbb{O}$ has similar effects to joint *regularization* of $\mathbf{W}$ and $\mathbf{H}$.
- Centralized features $\langle h \rangle \approx \bar{0}$ are (approximately) achieved by *batch-normalization*.

Neural Collapse with Block-Structured NTK

Theorem. Assume that the NTK block structure assumption holds. Assume further that the last-layer features are *centralized*, i.e, $\langle h \rangle = \bar{0}$, and the gradient flow dynamics invariant is zero, i.e., $\mathbf{E} = \mathbb{O}$. Then the DNN's dynamic exhibits neural collapse as defined in **(NC1)-(NC3)**.

- Zero invariant assumption $\mathbf{E} = \mathbb{O}$ has similar effects to joint *regularization* of $\mathbf{W}$ and $\mathbf{H}$.
- Centralized features $\langle h \rangle \approx \bar{0}$ are (approximately) achieved by *batch-normalization*.

$\rightsquigarrow$ *Regularization and batch-normalization are important for NC!*

Neural Collapse with Block-Structured NTK

Theorem. Assume that the NTK block structure assumption holds. Assume further that the last-layer features are *centralized*, i.e, $\langle h \rangle = \bar{0}$, and the gradient flow dynamics invariant is zero, i.e., $\mathbf{E} = \mathbb{O}$. Then the DNN's dynamic exhibits neural collapse as defined in **(NC1)-(NC3)**.

- Zero invariant assumption $\mathbf{E} = \mathbb{O}$ has similar effects to joint *regularization* of $\mathbf{W}$ and $\mathbf{H}$.
- Centralized features $\langle h \rangle \approx \bar{0}$ are (approximately) achieved by *batch-normalization*.

⇝ *Regularization and batch-normalization are important for NC!*

- If the additional assumptions do not hold, there are non-trivial *fixed points not satisfying NC* within our model.

Neural Collapse with Block-Structured NTK

Theorem. Assume that the NTK block structure assumption holds. Assume further that the last-layer features are *centralized*, i.e, $\langle h \rangle = \bar{0}$, and the gradient flow dynamics invariant is zero, i.e., $\mathbf{E} = \mathbb{O}$. Then the DNN's dynamic exhibits neural collapse as defined in **(NC1)-(NC3)**.

- Zero invariant assumption $\mathbf{E} = \mathbb{O}$ has similar effects to joint *regularization* of $\mathbf{W}$ and $\mathbf{H}$.
- Centralized features $\langle h \rangle \approx \bar{0}$ are (approximately) achieved by *batch-normalization*.

⇝ *Regularization and batch-normalization are important for NC!*

- If the additional assumptions do not hold, there are non-trivial *fixed points not satisfying NC* within our model.
- Condition $\mathbf{E} \propto \mathbf{W}^\top \mathbf{W} - c\langle h \rangle \langle h \rangle^\top$ is *necessary for NC* (zero invariant with $\langle h \rangle = \bar{0}$ is a special case).

Experiments

Architectures:
- ResNet20,
- VGG11/16,
- DenseNet40.

Datasets:
- MNIST,
- FashionMNIST,
- CIFAR10.

⇝ 9 models in total

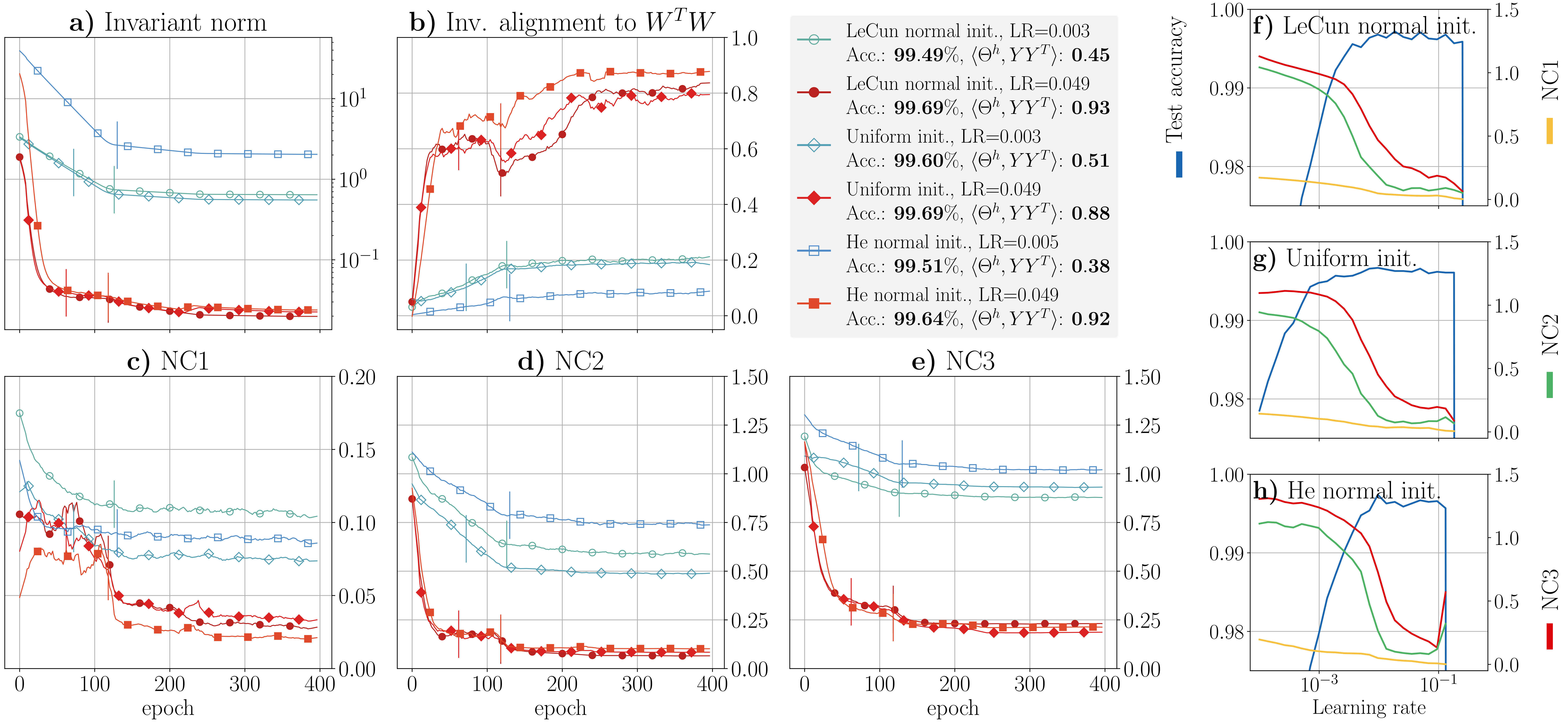


Figure: ResNet20 trained on MNIST. Red lines: DNNs that exhibit NC, blue lines: DNNs that do not exhibit NC.

Thanks for your attention!