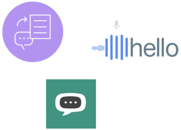
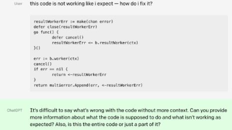
Robustness in Deep Learning: The good, the bad, the ugly

Zhenyu Zhu, **Fanghui Liu**, Grigorios G. Chrysos, Volkan Cevher

Laboratory for Information and Inference Systems (LIONS)
École Polytechnique Fédérale de Lausanne (EPFL)

Over-parameterization: more parameters than training data

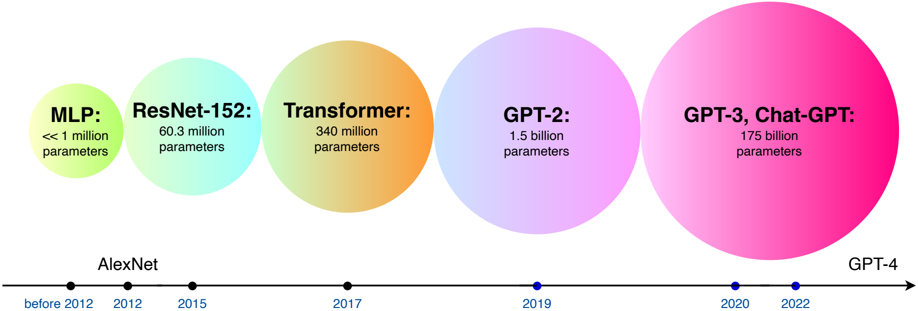



User this code is not working like i expect — how do i fix it?

```
resultWorkerErr := make(chan error)
defer close(resultWorkerErr)
go func() {
        defer cancel()
        resultWorkerErr <- b.resultWorker(ctx)
}()

err := b.worker(ctx)
cancel()
if err == nil {
        return <-resultWorkerErr
}
return multierror.Append(err, <-resultWorkerErr)
```

ChatGPT It's difficult to say what's wrong with the code without more context. Can you provide more information about what the code is supposed to do and what isn't working as expected? Also, is this the entire code or just a part of it?



Challenges in deep learning: robustness

Robust, Secure, Trustworthy Machine Learning

(a) Turtle classified as rifle [AEIK18].

(b) Stop sign classified as 45 mph sign [EEF+18].

Challenges in deep learning: robustness

Robust, Secure, Trustworthy Machine Learning

(a) Turtle classified as rifle [AEIK18].

(b) Stop sign classified as 45 mph sign [EEF+18].

Understanding robustness from function space theory!

Why function space theory is needed? (lazy training regime)

$$\mathcal{F}_{\mathrm{NN},m} = \left\{ f_m(\boldsymbol{x};\boldsymbol{\Theta}) = \sum_{i=1}^{m} a_i \max\left(\langle \boldsymbol{w}_i, \boldsymbol{x}\rangle, 0\right) : a_i \in \mathbb{R}, \boldsymbol{w}_i \in \mathbb{R}^d \right\}$$

- Gaussian initialization: $\boldsymbol{w}_i, a_i \sim \mathcal{N}(0, \texttt{var})$

Why function space theory is needed? (lazy training regime)

$$\mathcal{F}_{\mathrm{NN},m} = \left\{ f_m(\boldsymbol{x};\boldsymbol{\Theta}) = \sum_{i=1}^{m} a_i \max\left(\langle \boldsymbol{w}_i, \boldsymbol{x}\rangle, 0\right) : a_i \in \mathbb{R}, \boldsymbol{w}_i \in \mathbb{R}^d \right\}$$

- Gaussian initialization: $\boldsymbol{w}_i, a_i \sim \mathcal{N}(0, \texttt{var})$

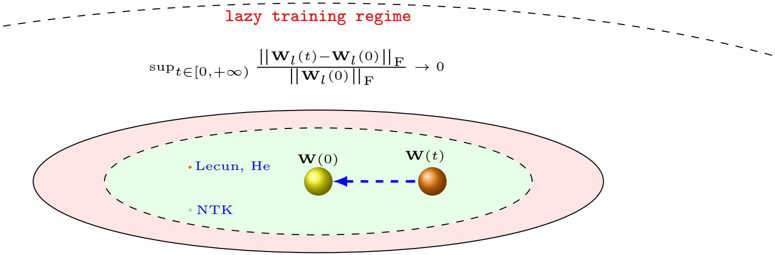


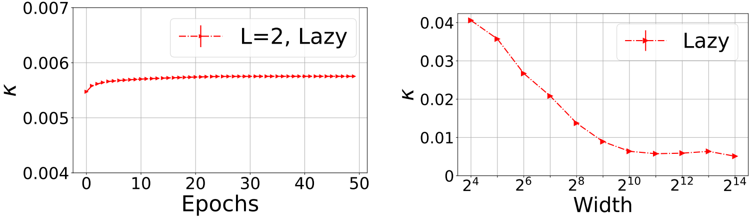
Figure: Training dynamics of two-layer ReLU NNs under different initializations [JGH18, COB19, LXMZ21].

Why function space theory is needed? (lazy training regime)

$$\mathcal{F}_{\mathrm{NN},m} = \left\{ f_m(\boldsymbol{x};\boldsymbol{\Theta}) = \sum_{i=1}^{m} a_i \max\left(\langle \boldsymbol{w}_i, \boldsymbol{x}\rangle, 0\right) : a_i \in \mathbb{R}, \boldsymbol{w}_i \in \mathbb{R}^d \right\}$$

- Gaussian initialization: $\boldsymbol{w}_i, a_i \sim \mathcal{N}(0, \mathtt{var})$

$$\text{lazy training ratio } \kappa := \frac{\sum_{l=1}^{L} \|\boldsymbol{W}_l(t) - \boldsymbol{W}_l(0)\|_{\mathrm{F}}}{\sum_{l=1}^{L} \|\boldsymbol{W}_l(0)\|_{\mathrm{F}}}$$



Why function space theory is needed? (non-lazy training regime)

$$\mathcal{F}_{\mathrm{NN},m} = \left\{ f_m(\boldsymbol{x};\boldsymbol{\Theta}) = \sum_{i=1}^{m} a_i \max\left(\langle \boldsymbol{w}_i, \boldsymbol{x}\rangle, 0\right) : \frac{1}{m}\sum_{i=1}^{m} |a_i| \|\boldsymbol{w}_i\|_1 < \infty \right\}$$

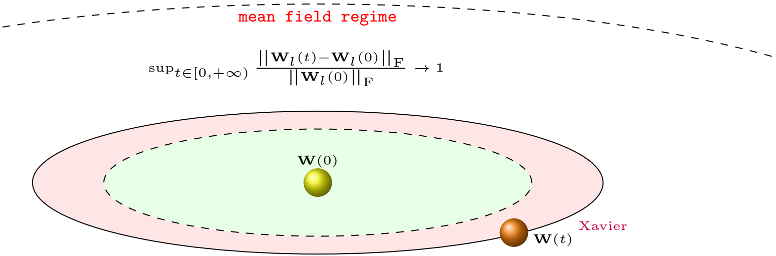


Figure: Training dynamics of two-layer ReLU NNs under different initializations [JGH18, COB19, LXMZ21].

Why function space theory is needed? (non-lazy training regime)

$$\mathcal{F}_{\mathrm{NN},m} = \left\{ f_m(\boldsymbol{x};\boldsymbol{\Theta}) = \sum_{i=1}^{m} a_i \max\left(\langle \boldsymbol{w}_i, \boldsymbol{x}\rangle, 0\right) : \frac{1}{m}\sum_{i=1}^{m} |a_i| \|\boldsymbol{w}_i\|_1 < \infty \right\}$$

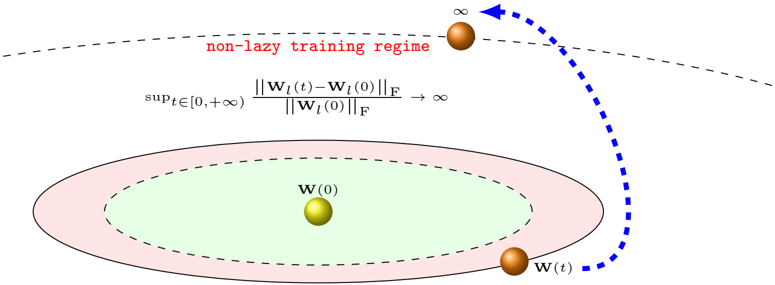
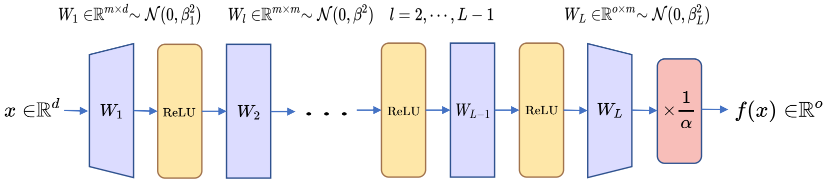


Figure: Training dynamics of two-layer ReLU NNs under different initializations [JGH18, COB19, LXMZ21].

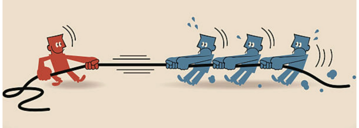
Architecture of DNNs



Initialization	Formulation
LeCun initialization	$\beta_1 = \sqrt{\frac{1}{d}},\ \beta = \beta_L = \sqrt{\frac{1}{m}}$
He initialization	$\beta_1 = \sqrt{\frac{2}{d}},\ \beta = \beta_L = \sqrt{\frac{2}{m}}$
NTK initialization	$\beta = \beta_1 = \sqrt{\frac{2}{m}},\ \beta_L = 1$

Over-parameterization helps or hurts robustness?[1]

Helps! [BS21]

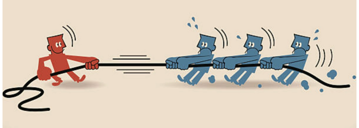


Hurts! [HJ22, WCC+21, HWE+21]

[1] Zhenyu Zhu, **Fanghui Liu**, Grigorios Chrysos, Volkan Cevher, *Robustness in deep learning: The good (width), the bad (depth), and the ugly (initialization)*. NeurIPS 2022.

Over-parameterization helps or hurts robustness?[1]

Helps! [BS21]



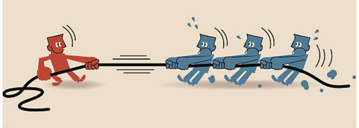
Hurts! [HJ22, WCC$^+$21, HWE$^+$21]

- initialization (e.g., lazy training, non-lazy training)
- architecture (e.g., width, depth)

[1] Zhenyu Zhu, **Fanghui Liu**, Grigorios Chrysos, Volkan Cevher, *Robustness in deep learning: The good (width), the bad (depth), and the ugly (initialization)*. NeurIPS 2022.

Over-parameterization helps or hurts robustness?[1]

Helps! [BS21]



Hurts! [HJ22, WCC$^+$21, HWE$^+$21]

- initialization (e.g., lazy training, non-lazy training)
- architecture (e.g., width, depth)

Definition (perturbation stability)

The perturbation stability of a ReLU DNN $f(\boldsymbol{x}; \boldsymbol{W})$ is

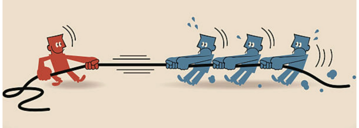
$$\mathscr{P}(f, \epsilon) = \mathbb{E}_{\boldsymbol{x}, \hat{\boldsymbol{x}}, \boldsymbol{W}} \left\| \nabla_{\boldsymbol{x}} f(\boldsymbol{x}; \boldsymbol{W})^\top (\boldsymbol{x} - \hat{\boldsymbol{x}}) \right\|_2, \quad \hat{\boldsymbol{x}} \sim \text{Unif}(\mathbb{B}(\epsilon, \boldsymbol{x})),$$

where ϵ is the perturbation radius.

[1] Zhenyu Zhu, **Fanghui Liu**, Grigorios Chrysos, Volkan Cevher, *Robustness in deep learning: The good (width), the bad (depth), and the ugly (initialization)*. NeurIPS 2022.

Over-parameterization helps or hurts robustness?[1]

Helps! [BS21]



Hurts! [HJ22, WCC+21, HWE+21]

- initialization (e.g., lazy training, non-lazy training)
- architecture (e.g., width, depth)

Definition (perturbation stability)

The perturbation stability of a ReLU DNN $f(\boldsymbol{x}; \boldsymbol{W})$ is

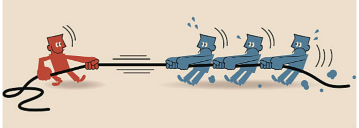
$$\mathscr{P}(f, \epsilon) = \mathbb{E}_{\boldsymbol{x}, \hat{\boldsymbol{x}}, \boldsymbol{W}(0)} \left\| \nabla_{\boldsymbol{x}} f(\boldsymbol{x}; \boldsymbol{W})^\top (\boldsymbol{x} - \hat{\boldsymbol{x}}) \right\|_2, \quad \hat{\boldsymbol{x}} \sim \text{Unif}(\mathbb{B}(\epsilon, \boldsymbol{x})),$$

where ϵ is the perturbation radius.

[1] Zhenyu Zhu, **Fanghui Liu**, Grigorios Chrysos, Volkan Cevher, *Robustness in deep learning: The good (width), the bad (depth), and the ugly (initialization)*. NeurIPS 2022.

Over-parameterization helps or hurts robustness?[1]

Helps! [BS21]



Hurts! [HJ22, WCC$^+$21, HWE$^+$21]

- initialization (e.g., lazy training, non-lazy training)
- architecture (e.g., width, depth)

Definition (perturbation stability)

The perturbation stability of a ReLU DNN $f(\boldsymbol{x};\boldsymbol{W})$ is

$$\mathscr{P}(f,\epsilon) = \mathbb{E}_{\boldsymbol{x},\hat{\boldsymbol{x}}} \left\| \nabla_{\boldsymbol{x}} f(\boldsymbol{x};\boldsymbol{W})^\top (\boldsymbol{x}-\hat{\boldsymbol{x}}) \right\|_2 , \quad \hat{\boldsymbol{x}} \sim \text{Unif}(\mathbb{B}(\epsilon,\boldsymbol{x})) ,$$

where ϵ is the perturbation radius.

[1] Zhenyu Zhu, **Fanghui Liu**, Grigorios Chrysos, Volkan Cevher, *Robustness in deep learning: The good (width), the bad (depth), and the ugly (initialization)*. NeurIPS 2022.

Main results (Lazy-training regime)

Theorem: perturbation stability $\lesssim \mathrm{Func}(m, L, \beta)$

Assumption	Initialization	Our bound for $\mathscr{P}(f,\epsilon)/\epsilon$	Trend of width m [1]	Trend of depth L [1]
$\|\boldsymbol{x}\|_2 = 1$	LeCun initialization	$\left(\sqrt{\frac{L^3 m}{d}e^{-m/L^3}} + \sqrt{\frac{1}{d}}\right)\left(\frac{\sqrt{2}}{2}\right)^{L-2}$	$\nearrow\searrow$	$\searrow$
	He initialization	$\sqrt{\frac{L^3 m}{d}e^{-m/L^3}} + \sqrt{\frac{1}{d}}$	$\nearrow\searrow$	$\nearrow$
	NTK initialization	$\sqrt{\frac{L^3 m}{d}e^{-m/L^3}} + 1$	$\nearrow\searrow$	$\nearrow$

[1] The larger perturbation stability means worse average robustness.

Takeaway messages: **the good (width), the bad (depth), the ugly (initialization)**

Main results (Lazy-training regime)

Theorem: perturbation stability $\lesssim \mathrm{Func}(m, L, \beta)$

Assumption	Initialization	Our bound for $\mathscr{P}(f,\epsilon)/\epsilon$	Trend of width m [1]	Trend of depth L [1]
$\|\boldsymbol{x}\|_2 = 1$	LeCun initialization	$\left(\sqrt{\frac{L^3 m}{d}} e^{-m/L^3} + \sqrt{\frac{1}{d}}\right)(\frac{\sqrt{2}}{2})^{L-2}$	$\nearrow \searrow$	$\searrow$
	He initialization	$\sqrt{\frac{L^3 m}{d}} e^{-m/L^3} + \sqrt{\frac{1}{d}}$	$\nearrow \searrow$	$\nearrow$
	NTK initialization	$\sqrt{\frac{L^3 m}{d}} e^{-m/L^3} + 1$	$\nearrow \searrow$	$\nearrow$

[1] The larger perturbation stability means worse average robustness.

Takeaway messages: **the good (width), the bad (depth), the ugly (initialization)**

- width **helps** robustness in the over-parameterized regime

Main results (Lazy-training regime)

Theorem: perturbation stability $\lesssim \mathrm{Func}(m, L, \beta)$

Assumption	Initialization	Our bound for $\mathscr{P}(f,\epsilon)/\epsilon$	Trend of width m [1]	Trend of depth L [1]
$\|\boldsymbol{x}\|_2 = 1$	LeCun initialization	$\left(\sqrt{\frac{L^3 m}{d}} e^{-m/L^3} + \sqrt{\frac{1}{d}}\right)(\frac{\sqrt{2}}{2})^{L-2}$	$\nearrow \searrow$	$\searrow$
	He initialization	$\sqrt{\frac{L^3 m}{d}} e^{-m/L^3} + \sqrt{\frac{1}{d}}$	$\nearrow \searrow$	$\nearrow$
	NTK initialization	$\sqrt{\frac{L^3 m}{d}} e^{-m/L^3} + 1$	$\nearrow \searrow$	$\nearrow$

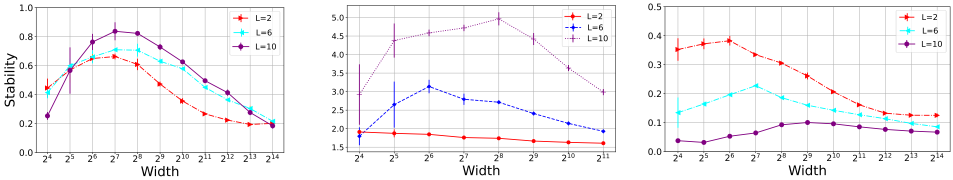
[1] The larger perturbation stability means worse average robustness.

Takeaway messages: **the good (width), the bad (depth), the ugly (initialization)**

- width **helps** robustness in the over-parameterized regime
- depth **helps** robustness in LeCun initialization but **hurts** robustness in He/NTK initialization

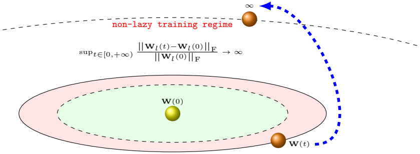
Experiments: robustness under lazy-training regime

Metrics	Ours (NTK initialization)	[WCC+21]	[HWE+21]
$\mathscr{P}(\boldsymbol{f}, \epsilon)/\epsilon$	$\sqrt{\frac{L^3 m}{d} e^{-m/L^3} + 1}$	$L^2 m^{1/3} \sqrt{\log m + \sqrt{mL}}$	$2^{\frac{3L-5}{2}} \sqrt{L}$



(a) He initialization (b) NTK initialization (c) LeCun initialization

Main results (Non-lazy training regime)



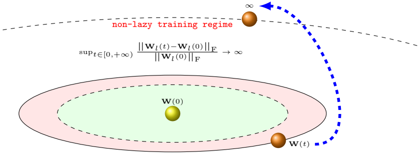
sufficient condition for DNNs

for large enough m and $m \gg d$, w.h.p, DNNs fall into non-lazy training regime if

$$\alpha \gg (m^{3/2} \sum_{i=1}^{L} \beta_i)^L .$$

e.g., $L = 2$, $\alpha = 1$, $\beta_1 = \beta_2 = \beta \sim \frac{1}{m^c}$ with $c > 1.5$

Main results (Non-lazy training regime)



sufficient condition for DNNs

for large enough m and $m \gg d$, w.h.p, DNNs fall into non-lazy training regime if

$$\alpha \gg (m^{3/2} \sum_{i=1}^{L} \beta_i)^L .$$

e.g., $L = 2$, $\alpha = 1$, $\beta_1 = \beta_2 = \beta \sim \frac{1}{m^c}$ with $c > 1.5$

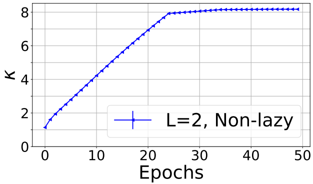
Theorem (non-lazy training regime for two-layer NNs)

Under this setting with $m \gg n^2$ and standard assumptions, then

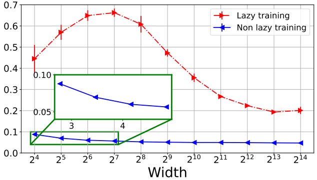
$$\frac{\mathscr{P}(\boldsymbol{f}_t, \epsilon)}{\epsilon} \le \widetilde{\mathcal{O}}\left(\frac{n}{m^{c+1.5}}\right), \ w.h.p$$

- width helps robustness in the over-parameterized regime in both lazy/non-lazy training regime

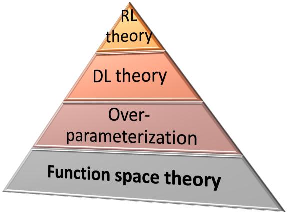
Experiment: Non-lazy training regime

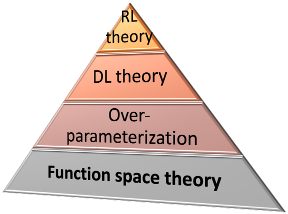


(a) lazy training ratio *vs.* epochs

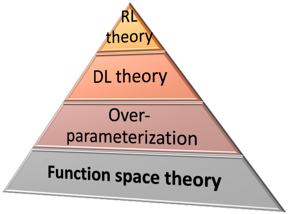


(b) perturbation stability


RL theory
DL theory
Over-parameterization
Function space theory



What is the role of over-parameterization in DNNs from the function space perspective?



What is the role of over-parameterization in DNNs from the function space perspective?

Take away messages:

- initialization, function spaces
- the good (width), the bad (depth), the ugly (initialization)

- ICASSP 2023 Tutorial - “Neural networks: the good, the bad, and the ugly”
- CVPR 2023 Tutorial - “Deep learning theory for computer vision"

- ICASSP 2023 Tutorial - “Neural networks: the good, the bad, and the ugly”
- CVPR 2023 Tutorial - “Deep learning theory for computer vision"

Thanks for your attention!

Q & A

my homepage www.lfhsgre.org for more information!

References I

[0] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok, *Synthesizing robust adversarial examples*, International Conference on Machine Learning, PMLR, 2018, pp. 284–293.
(Cited on pages 3 and 4.)

[0] Sébastien Bubeck and Mark Sellke, *A universal law of robustness via isoperimetry*, Advances in Neural Information Processing Systems, 2021, pp. 28811–28822.
(Cited on pages 11, 12, 13, 14, and 15.)

[0] Lenaic Chizat, Edouard Oyallon, and Francis Bach, *On lazy training in differentiable programming*, Advances in Neural Information Processing Systems, 2019, pp. 2933–2943.
(Cited on pages 5, 6, 8, and 9.)

[0] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song, *Robust physical-world attacks on deep learning visual classification*, IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1625–1634.
(Cited on pages 3 and 4.)

[0] Hamed Hassani and Adel Javanmard, *The curse of overparametrization in adversarial training: Precise analysis of robust generalization for random features regression*, arXiv preprint arXiv:2201.05149 (2022).
(Cited on pages 11, 12, 13, 14, and 15.)

References II

[0] Hanxun Huang, Yisen Wang, Sarah Erfani, Quanquan Gu, James Bailey, and Xingjun Ma, *Exploring architectural ingredients of adversarially robust deep neural networks*, Advances in Neural Information Processing Systems, 2021, pp. 5545–5559.
(Cited on pages 11, 12, 13, 14, 15, and 19.)

[0] Arthur Jacot, Franck Gabriel, and Clément Hongler, *Neural tangent kernel: Convergence and generalization in neural networks*, Advances in Neural Information Processing Systems, 2018, pp. 8571–8580.
(Cited on pages 5, 6, 8, and 9.)

[0] Tao Luo, Zhi-Qin John Xu, Zheng Ma, and Yaoyu Zhang, *Phase diagram for two-layer relu neural networks at infinite-width limit*, Journal of Machine Learning Research **22** (2021), no. 71, 1–47.
(Cited on pages 5, 6, 8, and 9.)

[0] Boxi Wu, Jinghui Chen, Deng Cai, Xiaofei He, and Quanquan Gu, *Do wider neural networks really help adversarial robustness?*, Advances in Neural Information Processing Systems, 2021, pp. 7054–7067.
(Cited on pages 11, 12, 13, 14, 15, and 19.)