REFERENCES

Ben Adlam and Jeffrey Pennington. Understanding double descent requires a fine-grained bias-variance decomposition. *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, 2020.

Ahmed M. Alaa and Mihaela van der Schaar. Discriminative jackknife: Quantifying uncertainty in deep learning via higher-order influence functions. *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.

Sanjeev Arora, Rong Ge, Behnam Neyshabur, and Yi Zhang. Stronger generalization bounds for deep nets via a compression approach. *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.

Sanjeev Arora, Simon S. Du, Wei Hu, Zhiyuan Li, Ruslan Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. *33rd Conference on Neural Information Processing Systems (NeurIPS)*, 2019a.

Sanjeev Arora, Simon S. Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019b.

Gregor Bachmann, Seyed-Mohsen Moosavi-Dezfooli, and Thomas Hofmann. Uniform convergence, adversarial spheres and a simple remedy. *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.

Peter Bartlett, Dylan J. Foster, and Matus Telgarsky. Spectrally-normalized margin bounds for neural networks. *31st Conference on Neural Information Processing Systems (Neurips)*, 2017.

Peter L. Bartlett, Nick Harvey, Chris Liaw, and Abbas Mehrabian. Nearly-tight vc-dimension and pseudodimension bounds for piecewise linear neural networks. *Journal of Machine Learning Research 20*, pp. 1–17, 2019.

Peter L. Bartlett, Philip M. Long, Gábor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070, 2020. ISSN 0027-8424.

Mikhail Belkin, Daniel J. Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116:15849 – 15854, 2019.

Yoshua Bengio and Yves Grandvalet. No unbiased estimator of the variance of k-fold cross-validation. *J. Mach. Learn. Res.*, 5:1089–1105, December 2004. ISSN 1532-4435.

Olivier Bousquet and André Elisseeff. Stability and generalization. *The Journal of Machine Learning Research*, 2:499–526, 2002.

K. O. Bowman, L. R. Shenton, and Paul C. Gailey. Distribution of the ratio of gamma variates. *Communications in Statistics - Simulation and Computation*, 27(1):1–19, 1998.

Leo Breiman. Heuristics of instability and stabilization in model selection. *The Annals of Statistics*, 24(6):2350 – 2383, 1996.

Gavin C. Cawley and Nicola L.C. Talbot. Efficient leave-one-out cross-validation of kernel fisher discriminant classifiers. *Pattern Recognition*, 36(11):2585–2592, 2003. ISSN 0031-3203.

Shuxiao Chen, Hangfeng He, and Weijie J. Su. Label-aware neural tangent kernel: Toward better generalization and local elasticity. *34th Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

Stéphane d'Ascoli, Maria Refinetti, Giulio Biroli, and Florent Krzakala. Double trouble in double descent : Bias and variance(s) in the lazy regime. *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of NAACL-HLT*, 2019.

Luc Devroye and Terry Wagner. Distribution-free inequalities for the deleted and holdout error estimates. *IEEE Transactions on Information Theory*, 25(2):202–207, 1979.

Simon S. Du, Kangcheng Hou, Barnabás Póczos, Ruslan Salakhutdinov, Ruosong Wang, and Keyulu Xu. Graph neural tangent kernel: Fusing graph neural networks with graph kernels. *34rd Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

Gintare Karolina Dziugaite and Daniel M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence*, 2017.

Gintare Karolina Dziugaite and Daniel M Roy. Data-dependent pac-bayes priors via differential privacy. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *32nd Conference on Neural Information Processing Systems (NeurIPS 2018)*, volume 31. Curran Associates, Inc., 2018.

André Elisseeff, Massimiliano Pontil, et al. Leave-one-out error and stability of learning algorithms with applications. *NATO science series sub series iii computer and systems sciences*, 190:111–130, 2003.

K. Fukunaga and D.M. Hummels. Leave-one-out procedures for nonparametric error estimates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(4):421–423, 1989.

Ouns El Harzli, Guillermo Valle-Pérez, and Ard A. Louis. Double-descent curves in neural networks: a new perspective using gaussian processes, 2021.

Trevor Hastie, Andrea Montanari, Saharon Rosset, and Ryan J. Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Thomas Hofmann, Bernhard Schölkopf, and Alexander J. Smola. Kernel methods in machine learning. *The Annals of Statistics*, 36(3), Jun 2008. ISSN 0090-5364.

Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269, 2017.

Kaixuan Huang, Yuqing Wang, Molei Tao, and Tuo Zhao. Why do deep residual networks generalize better than deep feedforward networks? – a neural tangent kernel perspective. *34rd Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *32rd Conference on Neural Information Processing Systems (NeurIPS)*, 2018.

Arthur Jacot, Berfin Şimşek, Francesco Spadaro, Clément Hongler, and Franck Gabriel. Implicit regularization of random feature models. *Proceedings of the International Conference on Machine Learning (ICML)*, 2020.

Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. *International Conference on Learning Representations (ICLR)*, 2019.

Michael Kearns and Dana Ron. Algorithmic stability and sanity-check bounds for leave-one-out cross-validation. *Neural Computation*, 11(6):1427–1453, 1999.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *Proceedings of the 5th International Conference on Learning Representations*, 2017.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario, 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

PA Lachenbruch. An almost unbiased method of obtaining confidence intervals for the probability of misclassification in discriminant analysis. *Biometrics*, 1967.

Yann LeCun and Corinna Cortes. MNIST handwritten digit database, 2010.

Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S. Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as gaussian processes. *International Conference on Learning Representations (ICLR)*, 2018.

Jaehoon Lee, Lechao Xiao, Samuel S. Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. *33rd Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

Song Mei and Andrea Montanari. The generalization error of random features regression: Precise asymptotics and the double descent curve. *Communications on Pure and Applied Mathematics*, 06 2021. doi: 10.1002/cpa.22008.

F. Mosteller and J. Tukey. Data analysis, including statistics. *Handbook of Social Psychology*, 2, 1968.

Vaishnavh Nagarajan and J. Zico Kolter. Uniform convergence may be unable to explain generalization in deep learning. *33rd Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. *Proceedings of The 28th Conference on Learning Theory (PMLR)*, 2015.

Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *International Conference on Learning Representations (ICLR)*, 2018.

Pratik Patil, Yuting Wei, Alessandro Rinaldo, and Ryan Tibshirani. Uniform consistency of cross-validation estimators for high-dimensional ridge regression. In Arindam Banerjee and Kenji Fukumizu (eds.), *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 3178–3186. PMLR, 13–15 Apr 2021.

Vern I. Paulsen and Mrinal Raghupathi. *An Introduction to the Theory of Reproducing Kernel Hilbert Spaces*, pp. i–iv. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2016.

M. Pontil. Leave-one-out error and stability of learning algorithms with applications. *International Journal of Systems Science*, 2002.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In J. Platt, D. Koller, Y. Singer, and S. Roweis (eds.), *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2008.

David Rolnick, Andreas Veit, Serge Belongie, and Nir Shavit. Deep learning is robust to massive label noise, 2018.

Sarath Shekkizhar and Antonio Ortega. Deepnnk: Explaining deep models and their generalization using polytope interpolation, 2020.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations (ICLR)*, 2015.

M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society*, 36(2):111–147, 1974.

Gaël Varoquaux. Cross-validation failure: Small sample sizes lead to large error bars. *NeuroImage*, 180:68–77, 2018. ISSN 1053-8119. New advances in encoding and decoding of brain signals.

Aki Vehtari, Andrew Gelman, and Jonah Gabry. Practical bayesian model evaluation using leave-one-out cross-validation and waic. *Statistics and Computing*, 27(5):1413–1432, Aug 2016. ISSN 1573-1375.

Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2018.

Ch. Walck. Hand-book on statistical distributions for experimentalists. 1996.

J. Weston. Leave-one-out support vector machines. In *IJCAI*, 1999.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in Neural Information Processing Systems*, 2014.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *International Conference on Learning Representations (ICLR)*, 2017.

Tong Zhang. Leave-one-out bounds for kernel methods. *Neural Computation*, 15(6):1397–1437, 2003.

Yongli Zhang and Yuhong Yang. Cross-validation for selecting a model selection procedure. *Journal of Econometrics*, 187(1):95–112, 2015. ISSN 0304-4076.

A Omitted Proofs

We list all the omitted proofs in the following section.

A.1 Proof of Theorem 3.2

Theorem 3.2. *Consider a kernel* $\mathrm{K} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ *and the associated objective with regularization parameter* $\lambda > 0$ *under mean squared loss. Define* $\boldsymbol{A} = \boldsymbol{K}\left(\boldsymbol{K} + \lambda \mathbf{1}_n\right)^{-1}$. *Then it holds that*

$$L_{LOO}\left(\mathcal{Q}_\lambda^{ERM}\right) = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K}\left(\Delta_{ik}^\lambda\right)^2, \qquad A_{LOO}\left(\mathcal{Q}_\lambda^{ERM}\right) = \frac{1}{n}\sum_{i=1}^{n}\mathbb{1}_{\left\{(\boldsymbol{y}_i - \Delta_{i\bullet})^* = \boldsymbol{y}_i^*\right\}}$$

where the residuals $\Delta_{ik}^\lambda \in \mathbb{R}$ *for* $i = 1, \ldots, n$, $k = 1, \ldots, K$ *is given by* $\Delta_{ik}^\lambda = \frac{Y_{ik} - \hat{f}_k^\lambda(\boldsymbol{x}_i)}{1 - A_{ii}}$.

Proof. Recall that $\hat{f}_{\mathcal{S}}^\lambda$ solves the optimization problem

$$\hat{f}_{\mathcal{S}}^\lambda = \mathrm{argmin}_{f\in\mathcal{F}}\, L_{\mathcal{S}}^\lambda(f) := \mathrm{argmin}_{f\in\mathcal{F}}\left\{\sum_{i=1}^{n}\sum_{k=1}^{K}\left(f_k(\boldsymbol{x}_i) - Y_{ik}\right)^2 + \lambda||f||_{\mathcal{H}}^2\right\}$$

and predicting on the training data takes the form $\hat{f}_S(\boldsymbol{X}) = \boldsymbol{A}\boldsymbol{Y}$ for some $\boldsymbol{A} \in \mathbb{R}^{n\times n}$. Now consider the model $f_\lambda^{-i} := \hat{f}_{\mathcal{S}_{-i}}^\lambda$ obtained from training on $\mathcal{S}_{-i}$. W.L.O.G. assume that $i = n$. We want to understand the quantity $\hat{f}_{\lambda,k}^{-n}(\boldsymbol{x}_n)$, i.e. the k-th component of the prediction on $\boldsymbol{x}_i$. To that

end, consider the dataset $\mathcal{Z} := \mathcal{S}_{-n} \cup \left\{ \left(\boldsymbol{x}_n, \hat{f}_\lambda^{-n}(\boldsymbol{x}_n) \right) \right\}$. Notice that for any $f \in \mathcal{F}$, it holds

$$\begin{aligned}
L_{\mathcal{Z}}^{\lambda}(\hat{f}_\lambda^{-n}) &= \sum_{k=1}^{K} \left\{ \sum_{i=1}^{n-1} \left(\hat{f}_{\lambda,k}^{-n}(\boldsymbol{x}_i) - Y_{ik} \right)^2 + \left(\hat{f}_{\lambda,k}^{-n}(\boldsymbol{x}_n) - \hat{f}_{\lambda,k}^{-n}(\boldsymbol{x}_n) \right)^2 \right\} + \lambda ||\hat{f}_\lambda^{-n}||_{\mathcal{H}}^2 \\
&= \sum_{k=1}^{K} \sum_{i=1}^{n-1} \left(\hat{f}_{\lambda,k}^{-n}(\boldsymbol{x}_i) - Y_{ik} \right)^2 + \lambda ||\hat{f}_\lambda^{-n}||_{\mathcal{H}}^2 \\
&= L_{\mathcal{S}_{-n}}^{\lambda} \left(\hat{f}_\lambda^{-n} \right) \\
&\leq L_{\mathcal{S}_{-n}}^{\lambda}(f) \\
&\leq L_{\mathcal{S}_{-n}}^{\lambda}(f) + \sum_{k=1}^{K} \left(f_k(\boldsymbol{x}_n) - \hat{f}_{\lambda,k}^{-n}(\boldsymbol{x}_n) \right)^2 \\
&= L_{\mathcal{Z}}^{\lambda}(f)
\end{aligned}$$

where the first inequality follows because $\hat{f}^{-n}$ minimizes $L_{\mathcal{S}_{-n}}^{\lambda}$ by definition. Thus $\hat{f}_\lambda^{-n}$ also minimizes $L_{\mathcal{Z}}^{\lambda}$ and hence also takes the form

$$\hat{f}_\lambda^{-n}(\boldsymbol{X}) = \boldsymbol{A}\tilde{\boldsymbol{Y}}$$

where $\tilde{\boldsymbol{Y}} \in \mathbb{R}^{n \times K}$ such that $\tilde{Y}_{ik} = \begin{cases} Y_{ik} & \text{if } i \neq n \\ \hat{f}_{\lambda,k}^{-n}(\boldsymbol{x}_n) & \text{else} \end{cases}$. Now we care about the n-th prediction which is

$$\begin{aligned}
\hat{f}_{\lambda,k}^{-n}(\boldsymbol{x}_n) &= \sum_{j=1}^{n} A_{nj} \tilde{Y}_{jk} = \sum_{j=1}^{n-1} A_{nj} Y_{jk} + A_{nn} \hat{f}_{\lambda,k}^{-n}(\boldsymbol{x}_n) \\
&= \sum_{j=1}^{n} A_{nj} Y_{jk} - A_{nn} Y_{nk} + A_{nn} \hat{f}_{\lambda,k}^{-n}(\boldsymbol{x}_n) \\
&= \hat{f}_{\mathcal{S},k}^{\lambda}(\boldsymbol{x}_n) - A_{nn} Y_{nk} + A_{nn} \hat{f}_{\lambda,k}^{-n}(\boldsymbol{x}_n)
\end{aligned}$$

Solving for $\hat{f}_{\lambda,k}^{-n}(\boldsymbol{x}_n)$ gives

$$\hat{f}_{\lambda,k}^{-n}(\boldsymbol{x}_n) = \frac{\hat{f}_{\mathcal{S},k}^{\lambda}(\boldsymbol{x}_n) - A_{nn} Y_{nk}}{1 - A_{nn}} \tag{3}$$

Then, subtracting Y_{nk} leads to

$$Y_{nk} - \hat{f}_{\lambda,k}^{-n}(\boldsymbol{x}_n) = Y_{nk} - \frac{\hat{f}_{\mathcal{S},k}^{\lambda}(\boldsymbol{x}_n) - A_{nn} Y_{nk}}{1 - A_{nn}} = \frac{Y_{nk} - Y_{nk} A_{nn} - \hat{f}_{\mathcal{S},k}^{\lambda}(\boldsymbol{x}_n) + A_{nn} Y_{nk}}{1 - A_{nn}} = \frac{Y_{nk} - \hat{f}_{\mathcal{S},k}^{\lambda}(\boldsymbol{x}_n)}{1 - A_{nn}}$$

Squaring and summing the expression over n and k results in the formula.

For the accuracy, we know that we correctly predict if the maximal coordinate of $\hat{f}_\lambda^{-n}(\boldsymbol{x}_n)$ agrees with the maximal coordinate of $\boldsymbol{y}_n$, i.e.

$$\operatorname{argmax}_k \hat{f}_{\lambda,k}^{-n}(\boldsymbol{x}_n) = \operatorname{argmax}_k Y_{nk}$$

From equation 3, we notice that

$$\begin{aligned}
\operatorname{argmax}_k \hat{f}_{\lambda,k}^{-n}(\boldsymbol{x}_n) &= \operatorname{argmax}_k \frac{\hat{f}_{\mathcal{S},k}^{\lambda}(\boldsymbol{x}_n) - A_{nn} Y_{nk}}{1 - A_{nn}} = \operatorname{argmax}_k \frac{\hat{f}_{\mathcal{S},k}^{\lambda}(\boldsymbol{x}_n) - Y_{nk} + Y_{nk} - A_{nn} Y_{nk}}{1 - A_{nn}} \\
&= \operatorname{argmax}_k -\Delta_{nk}^{\lambda} + Y_{nk} \\
&= (\boldsymbol{y}_n - \boldsymbol{\Delta}_{n\bullet})^*
\end{aligned}$$

We thus have to check the indicator $\mathbb{1}_{\{(\boldsymbol{y}_n - \Delta_{n\bullet})^* = \boldsymbol{y}_n^*\}}$ and sum it over n to obtain the result. $\square$

A.2 BINARY CLASSIFICATION

Here we state the corresponding results in the case of binary classification. The formulation for the accuracy changes slightly as now the sign of the classifier serves as the prediction.

Proposition A.1. *Consider a kernel* $\mathrm{K} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ *and the associated objective with regularization parameter* $\lambda > 0$ *under mean squared loss. Define* $\boldsymbol{A} = \boldsymbol{K}\left(\boldsymbol{K} + \lambda \mathbf{1}_n\right)^{-1}$. *Then it holds that*

$$L_{LOO}\left(\mathcal{Q}_\lambda^{ERM}\right) = \frac{1}{n}\sum_{i=1}^{n}\left(\Delta_i^\lambda\right)^2, \qquad A_{LOO}\left(\mathcal{Q}_\lambda^{ERM}\right) = \frac{1}{n}\sum_{i=1}^{n}\mathbb{1}_{\{y_i\Delta_i^\lambda<1\}}$$

where the residuals $\Delta_i^\lambda \in \mathbb{R}$ *for* $i = 1, \ldots, n$ *is given by* $\Delta_i^\lambda = \frac{y_i - \hat{f}^\lambda(\boldsymbol{x}_i)}{1 - A_{ii}}$.

Proof. Notice that the result for L_{LOO} is analogous to the proof for Theorem 3.2 by setting $K = 1$. For binary classification we use the sign of the classifier as a decision rule, i.e. the classifier predicts correctly if $y\hat{f}^\lambda(\boldsymbol{x}) > 0$. We can thus calculate that

$$\begin{aligned}
y_n \hat{f}_\lambda^{-n}(\boldsymbol{x}_n) &= \frac{y_n \hat{f}_{\mathcal{S}}^\lambda(\boldsymbol{x}_n) - A_{nn}y_n^2}{1 - A_{nn}} = \frac{y_n \hat{f}_{\mathcal{S}}^\lambda(\boldsymbol{x}_n) - y_n^2 + y_n^2 - y_n^2 A_{nn}}{1 - A_{nn}} \\
&= y_n^2 - y_n \frac{y_n - \hat{f}_{\mathcal{S}}^\lambda(\boldsymbol{x}_n)}{1 - A_{nn}} \\
&= 1 - y_n \frac{y_n - \hat{f}_{\mathcal{S}}^\lambda(\boldsymbol{x}_n)}{1 - A_{nn}} \\
&= 1 - y_n \Delta_n^\lambda
\end{aligned}$$

Thus, the n-th sample is correctly classified if and only if

$$1 - y_n\Delta_n^\lambda > 0 \iff y_n\Delta_n^\lambda < 1$$

We now just count the correct predictions for the accuracy, i.e.

$$A_{\text{LOO}}\left(\mathcal{Q}_\lambda^{\text{ERM}}\right) = \frac{1}{n}\sum_{i=1}^{n}\mathbb{1}_{\{y_i\Delta_i^\lambda<1\}}$$

□

A.3 PROOF OF COROLLARY 3.3

Corollary 3.3. *Consider the eigendecomposition* $\boldsymbol{K} = \boldsymbol{V}\,\mathrm{diag}(\boldsymbol{\omega})\boldsymbol{V}^T$ *for* $\boldsymbol{V} \in O(n)$ *and* $\boldsymbol{\omega} \in \mathbb{R}^n$. *Denote its rank by* $r = \mathrm{rank}(\boldsymbol{K})$. *Then it holds that the residuals* $\Delta_{ik}^\lambda \in \mathbb{R}$ *can be expressed as*

$$\Delta_{ik}^\lambda(r) = \sum_{l=1}^{n} Y_{lk} \frac{\sum_{k=r+1}^{n} V_{ik}V_{lk} + \sum_{k=1}^{r}\frac{\lambda}{\lambda+\omega_k}V_{ik}V_{lk}}{\sum_{k=r+1}^{n} V_{ik}^2 + \sum_{k=1}^{r}\frac{\lambda}{\lambda+\omega_k}V_{ik}^2}$$

Moreover for zero regularization, i.e. $\lambda \to 0$, *it holds that*

$$\Delta_{ik}^\lambda(r) \to \Delta_{ik}(r) = \begin{cases} \sum_{l=1}^{n} Y_{lk}\frac{\sum_{j=r+1}^{n} V_{ij}V_{lj}}{\sum_{j=r+1}^{n} V_{ij}^2} & \text{if } r < n \\ \sum_{l=1}^{n} Y_{lk}\frac{\sum_{j=1}^{n}\frac{1}{\omega_j}V_{ij}V_{lj}}{\sum_{j=1}^{n}\frac{1}{\omega_j}V_{ij}^2} & \text{if } r = n \end{cases}$$

Proof. Define $\boldsymbol{A} = \boldsymbol{K}\left(\boldsymbol{K} + \lambda\mathbf{1}_n\right)^{-1} \in \mathbb{R}^{n\times n}$. Recall that $\hat{f}_{\mathcal{S}}^\lambda(\boldsymbol{X}) = \boldsymbol{A}\boldsymbol{y}$ and thus $\hat{f}_k^\lambda(\boldsymbol{x}_i) = \sum_{j=1}^{n} A_{ij}Y_{jk}$. Let us first simplify $\boldsymbol{A}$:

$$\begin{aligned}
\boldsymbol{A} &= \boldsymbol{K}\left(\boldsymbol{K} + \lambda\mathbf{1}_n\right)^{-1} = \boldsymbol{V}\,\mathrm{diag}\left(\boldsymbol{\omega}\right)\boldsymbol{V}^T\left(\boldsymbol{V}\,\mathrm{diag}\left(\boldsymbol{\omega}\right)\boldsymbol{V}^T + \lambda\mathbf{1}_n\right)^{-1} \\
&= \boldsymbol{V}\,\mathrm{diag}\left(\frac{\boldsymbol{\omega}}{\boldsymbol{\omega}+\lambda}\right)\boldsymbol{V}^T
\end{aligned}$$

We can characterize the off-diagonal elements for $i \neq j$ as follows:

$$A_{ij} = \sum_{k=1}^{n} V_{ik} \frac{\omega_i}{\omega_i + \lambda} V_{jk} = \sum_{k=1}^{r} V_{ik} V_{jk} \frac{\omega_i}{\omega_i + \lambda} = \sum_{k=1}^{r} V_{ik} V_{jk} - \sum_{k=1}^{r} \frac{\lambda}{\omega_k + \lambda} V_{ik} V_{jk}$$
$$= - \sum_{k=r+1}^{n} V_{ik} V_{jk} - \sum_{k=1}^{r} \frac{\lambda}{\omega_k + \lambda} V_{ik} V_{jk}$$

where we made use of the fact that $\boldsymbol{V}_{i\bullet} \perp \boldsymbol{V}_{j\bullet}$. The diagonal elements $i = j$ on the other hand can be written as

$$A_{ii} = \sum_{k=1}^{n} V_{ik}^2 \frac{\omega_i}{\omega_i + \lambda} = \sum_{k=1}^{r} V_{ik}^2 \frac{\omega_k}{\omega_k + \lambda} = \sum_{k=1}^{r} V_{ik}^2 - \sum_{k=1}^{r} V_{ik}^2 \frac{\lambda}{\omega_k + \lambda}$$
$$= 1 - \sum_{k=r+1}^{n} V_{ik}^2 - \sum_{k=1}^{r} V_{ik}^2 \frac{\lambda}{\omega_k + \lambda}$$

where we have made use of the fact that $\boldsymbol{V}_{i\bullet}$ is a unit vector. Plugging-in the quantities into L_{LOO} results in

$$\Delta_{ik}^{\lambda} = \frac{Y_{ik} - \hat{f}_k^{\lambda}(\boldsymbol{x}_i)}{1 - A_{ii}} = \frac{Y_{ik} - \sum_{l=1}^{n} A_{il} Y_{lk}}{1 - A_{ii}} = \frac{Y_{ik} - A_{ii} y_i - \sum_{l \neq i}^{n} A_{il} Y_{lk}}{1 - A_{ii}}$$
$$= \frac{Y_{ik} \left(\sum_{j=r+1}^{n} V_{ij}^2 + \sum_{k=1}^{r} V_{ij}^2 \frac{\lambda}{\omega_j + \lambda} \right) + \sum_{l \neq i}^{n} Y_{lk} \left(\sum_{j=r+1}^{n} V_{ij} V_{lj} + \sum_{j=1}^{r} \frac{\lambda}{\omega_j + \lambda} V_{ij} V_{lj} \right)}{\sum_{j=r+1}^{n} V_{ij}^2 + \sum_{j=1}^{r} V_{ij}^2 \frac{\lambda}{\omega_l + \lambda}}$$
$$= \frac{\sum_{l=1}^{n} Y_{lk} \left(\sum_{j=r+1}^{n} V_{ij} V_{lj} + \sum_{j=1}^{r} \frac{\lambda}{\omega_j + \lambda} V_{ij} V_{lj} \right)}{\sum_{j=r+1}^{n} V_{ij}^2 + \sum_{j=1}^{r} V_{ij}^2 \frac{\lambda}{\omega_k + \lambda}}$$

Now crucially, in the full rank case $r = n$, we have empty sums, i.e. $\sum_{l=r+1}^{n} = \sum_{l=n+1}^{n} = 0$ and we obtain

$$\Delta_{ik}^{\lambda} = \frac{\sum_{l=1}^{n} Y_{lk} \sum_{j=1}^{r} \frac{\lambda}{\omega_j + \lambda} V_{ij} V_{lj}}{\sum_{j=1}^{r} V_{ij}^2 \frac{\lambda}{\omega_j + \lambda}} = \frac{\sum_{l=1}^{n} Y_{lk} \sum_{j=1}^{r} \frac{1}{\omega_j + \lambda} V_{ij} V_{lj}}{\sum_{j=1}^{r} V_{ij}^2 \frac{1}{\omega_j + \lambda}}$$
$$\xrightarrow{\lambda \to 0} \sum_{l=1}^{n} Y_{lk} \frac{\sum_{j=1}^{r} \frac{1}{\omega_j} V_{ij} V_{lj}}{\sum_{j=1}^{r} \frac{1}{\omega_j} V_{ij}^2}$$

On the other hand, in the rank deficient case $r < n$ we can cancel the regularization term:

$$\Delta_{ik}^{\lambda} \xrightarrow{\lambda \to 0} \sum_{l=1}^{n} Y_{lk} \frac{\sum_{j=r+1}^{n} V_{ij} V_{lj}}{\sum_{j=r+1}^{n} V_{ij}^2}$$

Plugging this into the formulas for L_{LOO}^{λ} and A_{LOO}^{λ} concludes the proof. □

A.4 Proof of Proposition 4.1

Proposition 4.1. *Consider a kernel with spectral decomposition $\boldsymbol{K} = \boldsymbol{V} \operatorname{diag}(\boldsymbol{\omega}) \boldsymbol{V}^T$ for $\boldsymbol{V} \in \mathbb{R}^{n \times n}$ orthogonal and $\boldsymbol{\omega} \in \mathbb{R}^n$. Assume that $\operatorname{rank}(\boldsymbol{K}) = n$. Then it holds that the leave-one-out error $L_{LOO}(\tilde{\boldsymbol{Y}}; \boldsymbol{Y})$ for a model trained on $\tilde{\mathcal{S}} = \{(\boldsymbol{x}_i, \tilde{\boldsymbol{y}}_i)\}_{i=1}^n$ but evaluated on $\mathcal{S} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$ is given by*

$$L_{LOO}(\tilde{\boldsymbol{Y}}; \boldsymbol{Y}) = \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{K} \left(\tilde{\Delta}_{ik} + Y_{ik} - \tilde{Y}_{ik} \right)^2, \qquad A_{LOO}(\tilde{\boldsymbol{Y}}; \boldsymbol{Y}) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{\left\{ (\tilde{\boldsymbol{y}}_i - \tilde{\boldsymbol{\Delta}}_{i\bullet})^* = \boldsymbol{y}_i^* \right\}}$$

where $\tilde{\Delta}_{ik} = \Delta_{ik}(\tilde{\boldsymbol{Y}}) \in \mathbb{R}$ is defined as in Corollary 3.3.

Proof. Denote by $\tilde{\mathcal{S}}_{-i}$ the dataset $\{(\boldsymbol{x}_j, \tilde{y}_j)\}_{j\neq i}^n$. Denote by $\tilde{f}_\lambda^{-i}$ the model trained on $\tilde{\mathcal{S}}_{-i}$. Recall from the proof of Theorem 3.2 that

$$\tilde{f}_{\lambda,k}^{-n}(\boldsymbol{x}_n) = \frac{\tilde{f}_k^\lambda(\boldsymbol{x}_n) - A_{nn}\tilde{Y}_{nk}}{1 - A_{nn}}$$

Instead of subtracting the same label $\tilde{Y}_{nk}$, we now subtract the evaluation label Y_{nk}:

$$\begin{aligned} Y_{nk} - \tilde{f}_{\lambda,k}^{-n}(\boldsymbol{x}_n) &= \frac{Y_{nk} - A_{nn}Y_{nk} - \tilde{f}_k^\lambda(\boldsymbol{x}_n) + A_{nn}\tilde{Y}_{nk}}{1 - A_{nn}} = \frac{(1 - A_{nn})(Y_{nk} - \tilde{Y}_{nk}) + \tilde{Y}_{nk} - \tilde{f}_k^\lambda(\boldsymbol{x}_n)}{1 - A_{nn}} \\ &= (Y_{nk} - \tilde{Y}_{nk}) + \frac{\tilde{Y}_{nk} - \tilde{f}_k^\lambda(\boldsymbol{x}_n)}{1 - A_{nn}} \\ &= (Y_{nk} - \tilde{Y}_{nk}) + \tilde{\Delta}_{ik}^\lambda \end{aligned}$$

The second term $\tilde{\Delta}_{ik}^\lambda$ is now the summand of the standard leave-one-out error where we evaluate on $\tilde{\boldsymbol{Y}}$. We can hence re-use Theorem 3.3 to decompose it. Squaring and summing over n concludes the LOO loss result. For the accuracy, we notice that a similar derivation as for Theorem 3.2 applies:

$$\begin{aligned} \operatorname{argmax}_k \tilde{f}_{\lambda,k}^{-n}(\boldsymbol{x}_n) &= \operatorname{argmax}_k \frac{\tilde{f}_k^\lambda(\boldsymbol{x}_n) - A_{nn}\tilde{Y}_{nk}}{1 - A_{nn}} = \operatorname{argmax}_k \frac{\hat{f}_k^\lambda(\boldsymbol{x}_n) - \tilde{Y}_{nk} + \tilde{Y}_{nk} - A_{nn}\tilde{Y}_{nk}}{1 - A_{nn}} \\ &= \operatorname{argmax}_k -\tilde{\Delta}_{nk}^\lambda + \tilde{Y}_{nk} \\ &= \left(\tilde{\boldsymbol{y}}_n - \tilde{\boldsymbol{\Delta}}_{n\bullet}\right)^* \end{aligned}$$

We thus have to check the indicator against the true label $\boldsymbol{y}_n$, i.e. $\mathbb{1}_{\{(\tilde{\boldsymbol{y}}_n - \tilde{\Delta}_{n\bullet})^* = \boldsymbol{y}_n^*\}}$ and sum it over n to obtain the result. □

A.5 Proof of Theorem 4.3

Theorem 4.3. *For large enough $n \in \mathbb{N}$, we can estimate as*

$$L_{LOO}^n(m^*) \gtrapprox 2nA$$

where $A \sim \Gamma(\frac{1}{2}, 1)$ is independent of n. $L_{LOO}^n(m^*)$ hence diverges a.s. with $n \to \infty$.

Proof. First we notice that for $m = m^*$, by definition it holds that $r = n - 1$, which simplifies the LOO expression to

$$L_{\text{LOO}}^n(m^*) \xrightarrow{\lambda \to 0} \frac{1}{n}\left(\sum_{i=1}^n \frac{1}{V_{in}^2}\right)\left(\sum_{i=1}^n y_i V_{in}\right)^2$$

For notational convenience, we will introduce $\boldsymbol{v} \in \mathbb{R}^n$ such that $v_i := V_{in}$. We will now bound the both factors one-by-one. The first part is a simple application of Proposition B.1 and Proposition B.5:

$$L_{LOO}^n(m^*) = \frac{1}{n}\left(\sum_{i=1}^n v_i\right)^2 \sum_{i=1}^n \frac{1}{v_i^2} \geq n^2 \frac{1}{n}\left(\sum_{i=1}^n v_i\right)^2 \overset{(d)}{=} n^2 B$$

Now for large enough n, we can use Lemma B.6 to make the following approximation in distribution:

$$nB \approx 2\frac{n-1}{2} B \xrightarrow{(d)} 2A$$

where $A \sim \Gamma(\frac{1}{2}, 1)$. Thus for large enough n, it holds that

$$L_{LOO}^n(m^*) \gtrapprox 2nA$$

As the approximation becomes exact for larger and larger n, we conclude that

$$L_{LOO}^n(m^*) \xrightarrow{n \to \infty} \infty \quad \text{a.s.}$$

□

B ADDITIONAL LEMMAS

In this section we present the additional technical Lemmas needed for the proofs of the main claims in A.

Lemma B.1. *Consider a unit vector $\boldsymbol{v} \in \mathbb{S}^{n-1}$. Then it holds that*

$$\sum_{i=1}^{n} \frac{1}{v_i^2} \geq n^2$$

Proof. Let's parametrize each v_i as

$$v_i = \frac{z_i}{\sqrt{\sum_{i=1}^{n} z_i^2}}$$

for $i = 1, \ldots, n$ and $\boldsymbol{z} \in \mathbb{R}^n$. One can easily check that $||\boldsymbol{v}||_2 = 1$ and hence $\boldsymbol{v} \in \mathbb{S}^{n-1}$. Plugging this in, we arrive at

$$\sum_{i=1}^{n} \frac{1}{v_i^2} = \sum_{i=1}^{n} \frac{\sum_{j=1}^{n} z_j^2}{z_i^2} = \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{z_j^2}{z_i^2} = n + \sum_{i=1}^{n} \sum_{j \neq i}^{n} \frac{z_j^2}{z_i^2}$$

We can re-arrange the sum into pairs

$$\frac{z_j^2}{z_i^2} + \frac{z_i^2}{z_j^2} = a^2 + \frac{1}{a^2} \geq 2$$

for $a^2 = \frac{z_j^2}{z_i^2} > 0$ and using the fact that $x + \frac{1}{x} \geq 2$ for $x \geq 0$. We can find $\frac{n(n-1)}{2}$ such summands, and thus

$$\sum_{i=1}^{n} \frac{1}{v_i^2} \geq n + 2\frac{n(n-1)}{2} = n^2$$

□

Lemma B.2. *Consider $\boldsymbol{v} \sim \mathcal{U}\left(\mathbb{S}^{n-1}\right)$ and any fixed orthogonal matrix $\boldsymbol{U} \in O(n)$. Then it holds that*

$$\boldsymbol{U}\boldsymbol{v} \overset{(d)}{=} \boldsymbol{v}$$

Proof. This is a standard result and can for instance be found in Vershynin (2018). □

Lemma B.3. *Consider $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \mathbb{1}_n)$. Then it holds that*

$$\boldsymbol{v} = \frac{\boldsymbol{w}}{||\boldsymbol{w}||_2} \sim \mathcal{U}\left(\mathbb{S}^{n-1}\right)$$

Proof. This is a standard result and can for instance be found in Vershynin (2018). □

Lemma B.4. *Consider two independent Gamma variables $X \sim \text{Gamma}(\alpha, \nu)$ and $Y \sim \text{Gamma}(\beta, \nu)$. Then it holds that*

$$\frac{X}{X+Y} \sim \text{Beta}\left(\alpha, \beta\right)$$

Proof. This is a standard result and can for instance be found in Bowman et al. (1998). □

Lemma B.5. *Consider $\boldsymbol{v} \sim \mathcal{U}\left(\mathbb{S}^{n-1}\right)$. Then it holds that*

$$\frac{1}{n}\left(\sum_{i=1}^{n} y_i v_i\right)^2 \sim \text{Beta}\left(\frac{1}{2}, \frac{n-1}{2}\right)$$

Proof. First realize that we can write

$$\frac{1}{\sqrt{n}}\sum_{i=1}^{n} y_i v_i = \tilde{\mathbf{1}}_n^T (\boldsymbol{v} \odot \boldsymbol{y}) \overset{(d)}{=} \tilde{\mathbf{1}}_n^T \boldsymbol{v}$$

where $\tilde{\mathbf{1}}_n = \left(\frac{1}{\sqrt{n}}, \dots, \frac{1}{\sqrt{n}}\right)$ with $||\tilde{\mathbf{1}}_n||_2 = 1$ and the fact that $\boldsymbol{v} \odot \boldsymbol{y} \overset{(d)}{=} \boldsymbol{v}$ for fixed $\boldsymbol{y} \in \{-1, 1\}^n$. The idea is now to choose $\boldsymbol{U} \in O(n)$ such that $\boldsymbol{U}^T \tilde{\mathbf{1}}_n = \boldsymbol{e}_1 = (1, 0, \dots, 0)$. Then by using Lemma B.2, it holds

$$\tilde{\mathbf{1}}_n^T \boldsymbol{v} \overset{(d)}{=} \tilde{\mathbf{1}}_n^T \boldsymbol{U} \boldsymbol{v} \overset{(d)}{=} \left(\boldsymbol{U}\tilde{\mathbf{1}}_n\right)^T \boldsymbol{v} \overset{(d)}{=} \boldsymbol{e}_1^T \boldsymbol{v} \overset{(d)}{=} v_1$$

Thus, surprisingly, it suffices to understand the distribution of v_1. By Lemma B.3, we know that

$$v_1 \overset{(d)}{=} \frac{z_1}{\sqrt{z_1^2 + \dots + z_n^2}}$$

where $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \mathbb{1}_n)$. We are interested in the square of this expression,

$$\frac{1}{n}\left(\sum_{i=1}^{n} v_i\right)^2 \overset{(d)}{=} v_1^2 \overset{(d)}{=} \frac{z_1^2}{z_1^2 + \dots + z_n^2} \overset{(d)}{=} \frac{z_1^2}{z_1^2 + w}$$

where we define $w = \sum_{i=2}^{n} z_i^2$, clearly independent of z_1^2. Moreover, it holds that $z_1^2 \sim \text{Gamma}\left(\frac{1}{2}, \frac{1}{2}\right)$ and $w \sim \text{Gamma}\left(\frac{n-1}{2}, \frac{1}{2}\right)$. Thus, by Lemma B.4 we can conclude that

$$\frac{1}{n}\left(\sum_{i=1}^{n} v_i\right)^2 \sim \text{Beta}\left(\frac{1}{2}, \frac{n-1}{2}\right)$$

□

Lemma B.6. *Consider the sequence of Beta distributions* $X_n \sim \text{Beta}(k, n)$. *Then it holds that*

$$nX_n \xrightarrow{(d)} \text{Gamma}(k, 1)$$

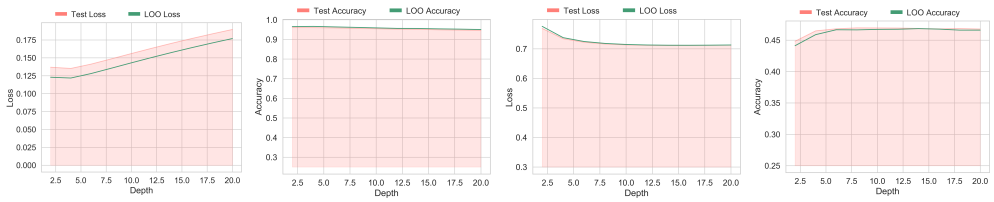
Proof. This is a standard result and can for instance be found in Walck (1996). □

C Further Experiments

In this section we present additional experimental results on the leave-one-out error.

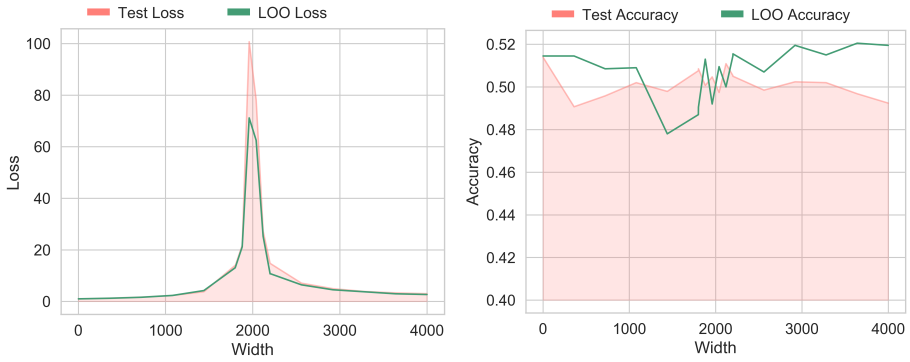
C.1 LOO as function of depth L

We study how the depth L of the NTK kernel $\Theta^{(L)}$ affects the performance of LOO loss and accuracy. We use the datasets *MNIST* and *CIFAR10* with $n = 5000$ and evaluate NTK models with depth ranging from 3 to 20. We present our findings in Figure 4. Again we see a very close match between LOO and the corresponding test quantity for *CIFAR10*. Interestingly the performance is slightly worse for very shallow models. For *MNIST* we see a gap between LOO loss and test loss, which is due to the very zoomed-in nature of the plot (the gap is actually only 0.015) as the loss values are very small in general. Indeed we observe an excellent match between the test and LOO accuracy.



(a) Loss (b) Accuracy (c) L_{LOO}, *CIFAR10* (d) A_{LOO}, *CIFAR10*

Figure 4: Test and LOO losses (a, c) and accuracies (b, d) as a function of depth L. We use fully-connected NTK model on *MNIST* and *CIFAR10*.



(a) Loss (b) Accuracy

Figure 5: Test and LOO losses (a) and accuracies (b) as a function of sample width m. We use a random feature model on binary *MNIST* with random labels.

C.2 Double Descent with Random Labels

Here we demonstrate how the spike in double descent is a very universal phenomenon as demonstrated by Theorem 4.3. We consider a random feature model of varying width m on binary *MNIST* with $n = 2000$, where the labels are fully randomized ($p = 1$), destroying thus any relationship between the inputs and targets. Of course, there will be no double descent behaviour in the test accuracy as the network has to perform random guessing at any width. We display this in Figure 5. We observe that indeed the model is randomly guessing throughout all the regimes of overparametrization. Both the test and LOO loss however, exhibit a strong spike around the interpolation threshold. This underlines the universal nature of the phenomenon, connecting with the fact that Theorem 4.3 does not need any assumptions on the targets.