Grokking modular arithmetic can be explained by margin maximization

Mohamad Amin Mohamadi
University of British Columbia
lemohama@cs.ubc.ca

Zhiyuan Li
Toyota Technical Institute at Chicago
zhiyuanli@ttic.edu

Lei Wu
Peking University
leiwu@math.pku.edu.cn

Danica J. Sutherland
UBC and Amii
dsuth@cs.ubc.ca

Abstract

We present a margin-based generalization theory explaining the "grokking" phenomenon (Power et al., 2022), where the model generalizes long after overfitting to arithmetic datasets. Specifically, we study two-layer quadratic networks on mod-p arithmetic problems, and show that solutions with maximal margin normalized by ℓ_∞ norm generalize with $\tilde{O}(p^{5/3})$ samples. To the best of our knowledge, this is the first sample complexity bound strictly better than a trivial $O(p^2)$ complexity for modular addition. Empirically, we find that GD on unregularized $\ell - 2$ or cross entropy loss tend to maximize the margin. In contrast, we show that kernel-based models, such as networks that are well-approximated by their neural tangent kernel, need $\Omega(p^2)$ samples to achieve non-trivial ℓ_2 loss. Our theory suggests that grokking might be caused by overfitting in the kernel regime of early training, followed by generalization as gradient descent eventually leaves the kernel regime and maximizes the normalized margin.

1 Introduction

Power et al. (2022) demonstrated an intriguing phenomenon they called "grokking" when learning transformers on small algorithmic tasks: neural networks can find a generalizing solution long after they have overfit to the training dataset (with poor generalization). This observation has lead to a stream of recent works aimed at uncovering the mechanisms that can lead a network to "grok," and properties of the final solutions, on various algorithmic tasks.

Liu et al. (2022a) attributed the delayed generalization to the difficulty in learning the representations required for generalization. Thilak et al. (2022) showed that grokking with Adam optimizer and without any regularization only happens after a "slingshot" explosion in the training loss. Liu et al. (2022b) argued that grokking is linked to the weight norm of the learned solution, and can be reproduced on other non-algorithmic tasks through changing the scale of initialization. Barak et al. (2022) theoretically showed that a similar mechanism exists when learning parities in an online fashion. Nanda et al. (2023) reverse-engineered the final weights learned by the original Transformer. Most relatedly to our work, Gromov (2023) shows that a two-layer quadratic network with plain gradient descent on square loss can grok modular arithmetic, and gives an analytical expression for network weights that can solve the task.[1]

[1] Although the presented solution does solve the modular arithmetic tasks, in our experience gradient descent did not seem to find weights compatible with this construction, contrary to claims of Gromov (2023).

Mathematics of Modern Machine Learning Workshop at NeurIPS 2023.

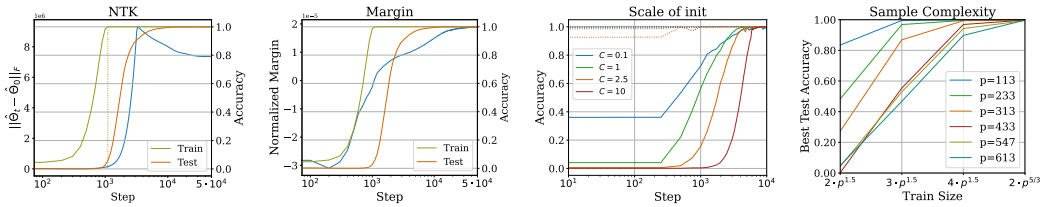

Figure 1: **From left to right, 1:** Change of empirical NTK[2] ($\|\hat{\Theta}_t - \hat{\Theta}_0\|_F$) is minimal until after overfitting. **2:** Normalized margin[3] steadily improves over training and can be used as a continuous progress measure for grokking. **3:** Smaller scale of initialization mitigates grokking, suggesting that Grokking might be caused by a transition from kernel to feature learning regime. **4:** Empirical evaluations support a sample complexity of $\tilde{\mathcal{O}}(p^{5/3})$ for GD trained with cross-entropy loss.[4]

Overall, however, a theoretical understanding of *why* grokking occurs has remained elusive. In this work, we present theoretical analysis for a concrete setting where grokking can be explained through a transition from kernel regime to the feature learning regime, caused by the margin-maximization implicit bias of GD when trained on cross-entropy loss. We prove that when initialized in the kernel regime, networks requires $\Omega(p^2)$ samples to generalize (with respect to the square loss), which leads them to overfit the training data with poor generalization. We further prove that when cross-entropy loss is used, the margin-maximization implicit bias of gradient descent drives it to generalize with $\tilde{\mathcal{O}}(p^{5/3})$ data points, making the transition from overfitting to generalizing possible. Empirical investigations suggest that this might explain grokking the modular addition task in the setting of two-layer NNs trained with cross-entropy loss.

2 Setup

We focus on the problem of modular addition $f(n, m) = n + m \mod p$, where p is a fixed integer and $n, m \in \mathbb{Z}_p$. The inputs are encoded as one-hot vectors of dimension p. We model the task of determining $f(n, m)$ as a multi-class classification problem, using one-hot labels. Following Gromov (2023), we use a two-layer feed-forward network with quadratic activation and no biases:

$$f(W, V; x) = V(Wx)^{\odot 2}, \tag{1}$$

where $x \in \mathbb{R}^{2p}$ is the concatenation of two one-hot variables corresponding to inputs n and m, and $a^{\odot 2}$ denotes the entry-wise square of a vector a. The network has h hidden units; we use $\theta = \mathrm{vec}\,(W, V)$ for the parameters of the network, where $W \in \mathbb{R}^{h \times 2p}$ and $V \in \mathbb{R}^{p \times h}$ are the weight matrices for the two layers of the network. Note that $f(\theta; x)$ is a 3-homogeneous function with respect to its parameters: letting $c\theta = \mathrm{vec}\,(cW, cV)$, we have

$$\forall c > 0; \quad f(c\theta, x) = c^3 f(\theta; x) \text{ for all } \theta \text{ and } x. \tag{2}$$

In all experiments, we initialize the networks according to the scheme of He et al. (2015), and minimize the cross-entropy loss using *vanilla* gradient descent (full-batch, no momentum, no weight decay). We use $\mathcal{D} = (\mathcal{X}, \mathcal{Y})$ to denote the full set of possible data, where $\mathcal{X} \in \mathbb{R}^{p^2 \times 2p}$ and $\mathcal{Y} \in \mathbb{R}^{p^2 \times p}$. We randomly partition $\mathcal{D}$ into $\mathcal{D}_{\text{train}}$ (on which we train) and $\mathcal{D}_{\text{test}}$ (used for evaluation).

3 Generalization Bound

Lyu & Li (2020) proved that gradient descent on homogeneous models with the cross-entropy (or similar) losses, in the absence of explicit regularization, maximizes the normalized margin. Specifically, although $\|\theta\| \to \infty$, it holds that $\theta/\|\theta\|_2$ converges to a solution of the following problem

[4] $\hat{\Theta}_t$ is the NTK on the training data, using the network at step t: $\hat{\Theta}_t \triangleq \nabla_\theta f(\theta_t; \mathcal{X}_{\text{train}}) \nabla_\theta f(\theta_t; \mathcal{X}_{\text{train}})^\top$.

[4] Normalized margin is defined as $q_{min}(\theta)/\|\theta\|_2^2$, please refer to Equation (4) for more details.

[4] With θ being the parameters distributed according to PyTorch's default initialization, we consider initializations of scale $C\theta$ where C varies in the plot.

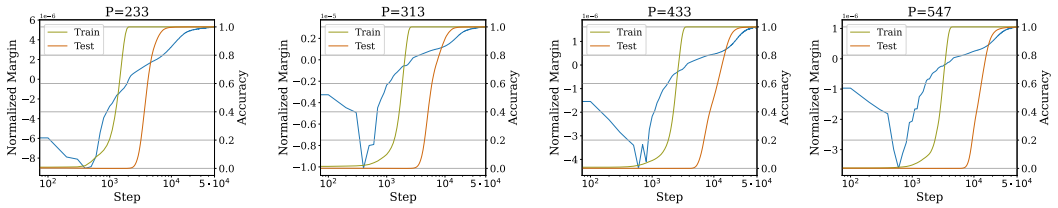

Figure 2: Evolution of normalized margin while training the NN with cross-entropy loss: although there's a large gap between overfitting and generalizing, normalized margin continuously improves and can be used as a progress measure towards generalization.

(when a solution exists):

$$\min \frac{1}{2}\|\theta\|_2^2 \quad \text{s.t.} \quad q_{\min}(\theta) \geq 1 \tag{3}$$

where $q_{\min}$ gives the minimum prediction margin over all training points,

$$q_{\min}(\theta) \triangleq \min_{(x,y)\in\mathcal{D}_{\text{train}}} yf(\theta;x). \tag{4}$$

Motivated by this implicit bias of gradient descent, we present a generalization bound for a two-layer network with quadratic activation with parameters close to the max-margin solution,[5] based on the PAC-Bayesian framework (McAllester, 2003). Our bound is based on Lemma 1 of Neyshabur et al. (2018), which provides a margin-based high probability generalization bound for any predictor based on the *margin loss*, which is like the standard zero-one loss but counts a prediction as correct only if it does so with a margin at least γ:

$$L_\gamma\left(f(\theta;\cdot),\mathcal{D}\right) \triangleq \mathbb{P}_{(x,y)\in\mathcal{D}}\left[f(\theta;x)[y] \leq \gamma + \max_{j\neq y} f(\theta;x)[j]\right]. \tag{5}$$

Theorem 1 (Informal). *Let $f:\mathbb{R}^{2p}\to\mathbb{R}^p$ be a quadratic network as described in Section 2, with $h=\mathcal{O}(p)$ hidden neurons. For any $\delta>0$, it holds with probability at least $1-\delta$ over the choice of training set $\mathcal{D}_{train}$ of size m that, for all θ satisfying $\|W\|_\infty=\mathcal{O}(1)$ and $\|V\|_\infty=\mathcal{O}(1)$,*

$$L_0(f,\mathcal{D}) \leq L_p(f,\mathcal{D}_{train}) + \widetilde{\mathcal{O}}\left(\sqrt{\frac{p^{5/3}}{m}}\right). \tag{6}$$

The proof of Theorem 1 is deferred to Appendix C.

Theorem 1 implies that gradient descent with cross-entropy loss has a sample complexity of $\tilde{\mathcal{O}}(p^{5/3})$ on the modular addition problem, confirming and explaining the previous observations in Power et al. (2022); Gromov (2023); Nanda et al. (2023); Liu et al. (2022b) and many other works on the minimum threshold for the fraction of data used to achieve generalization. In combination with Theorem 4.2 from Lyu & Li (2020), this gaurantees that given enough training data, gradient descent will eventually find a generalizing solution for the modular addition problem.

The assumption of constant L^∞ norm for the weights of the NN may seem too restrictive initially. In what follows, we presnt two cases where all the assumptions of Theorem 1 are satisfied.

First, consider using AdamW with a fixed weight decay to train the NN on the modular addition task. In Figure 3 we empirically observe that for different values of p, AdamW tends to converge to solutions whose L^∞ norm is constant, and is proportional to the inverse of the weight decay used.

Second, in Appendix B, we further present a manual construction of weights for the two-layer network that satisfies all the assumptions of Theorem 1, and achieves a normalized margin of $\Theta\left(\frac{1}{p^2}\right)$ – notably, much higher than the normalized margin of previous constructions, such as the one presented by Gromov (2023). This suggests that the solution to max-margin problem for our setup, equation 3, has a normalized margin of at least $\Omega\left(\frac{1}{p^2}\right)$.

[5] Appendix B presents a manual construction for which these assumptions are satisfied.

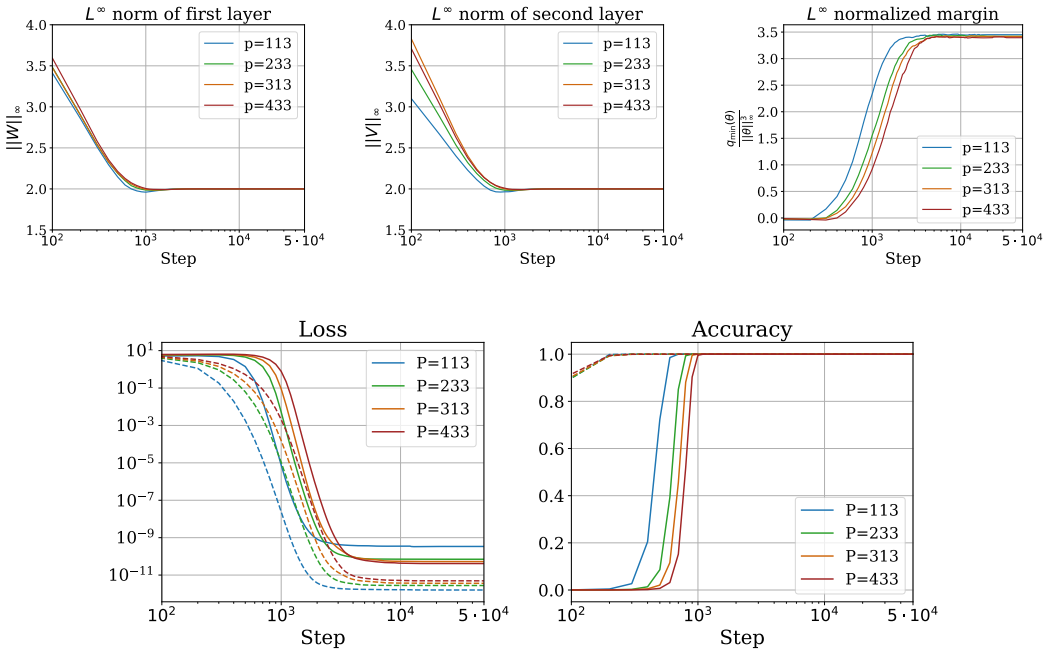

Figure 3: Evolution of the NN's weights in terms of L^∞ norm, the L^∞ normalized margin, loss and accuracy when AdamW with a weight decay of 0.5 is used to train the network. The amount of training data used for each experiment is $2p^{5/3}$, which leads to generalizartion as predicted by Theorem 1.

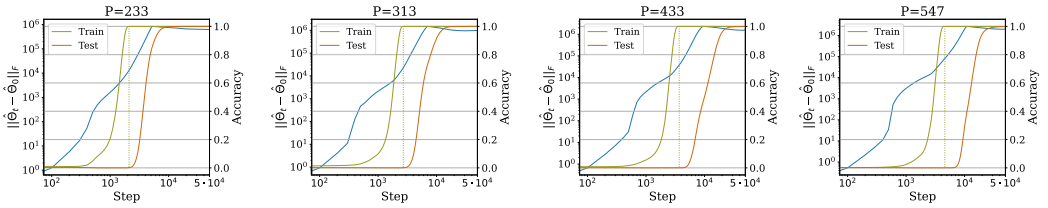

Figure 4: Evolution of the empirical NTK while training the NN with cross-entropy loss: although NTK changes during the overfitting phase of training, the change in NTK after generalization is orders of magnitude larger, implying a plausible transition from "kernel" to "rich" regime.

4 *Why* does grokking happen?

Theorem 1 establishes that, eventually, gradient descent will find a generalizing solution on this problem. It alone, though, does not explain the main surprise of grokking: that this occurs quite abruptly, long after overfitting to the dataset.

We will now argue that grokking might be caused by transitioning from the "kernel regime," where the neural network can be well-approximated by a kernel model, to the "feature-learning regime," where this is not true. With certain initialization schemes, gradient descent in wide networks follows a kernel regime at least for some time (Jacot et al., 2018; Arora et al., 2019; Lee et al., 2019; Chizat et al., 2019; Yang & Hu, 2021), but eventually the margin-maximization bias may lead to a departure from the kernel regime.

We first define the first-order Taylor expansion as $\hat{f}_i(\theta; x) \triangleq f_i(\theta_0; x) + \langle \nabla f_i(\theta_0; x), \theta - \theta_0 \rangle$. For simplicity, we focus on the case where the number of hidden neurons is sufficiently large and the Neural Tangent Kernel $K(x, x') \triangleq \langle \nabla f_0(\theta_0; x), \nabla f_0(\theta_0; x') \rangle$ converges to a fixed limit $K_\infty(x, x')$, for each pair of x and x'. We also assume that each parameter is initialized independently from a

symmetric distribution, except we use the "doubling trick"[6] to ensure $f(\theta_0; x) = 0$. This ensures the kernel satisfies the permutation invariant property (see Definition 13), which plays a crucial role in our proof. Our proof can in principle be generalized to random finite-width neural networks in the kernel regime using results from Appendix C of Li et al. (2020).

Theorem 2 (Informal version of Theorem 15)**.** *There exist constants $C > 0$, such that when the width is sufficiently large and the number of samples is $m \leq Cp^2$, it holds that the expectation of the best test ℓ_2 loss GD in kernel regime can achieve, is at least half of the test loss of the trivial solution,* i.e.*, the all-0 predictor:*

$$\mathbb{E}_{x_1,\ldots,x_n} L(\hat{f}_{\theta_t}) \geq \frac{1}{2}\mathbb{E}_{x_1,\ldots,x_n} L(\hat{f}_{\theta_0}). \tag{7}$$

Here $L(f)$ denotes the population ℓ_2 loss of function $f : \mathbb{R}^{2p} \to \mathbb{R}^p$, and θ_t is the GD iterate on loss $L(\hat{f}_\theta)$. This lower bound holds for any learning rate schedule and step t.

Theorem 2 is proved in Appendix D. This result shows that any permutation-invariant kernel method has a sample complexity of $\Omega(p^2)$ on the modular addition task, when mesaured with ℓ_2 loss. Arora et al. (2019); Lee et al. (2019) show that when the scale of the initialization of a network is large enough, gradient descent operates in a locally-linear fashion, such that at any stage of training, the outputs of the network being trained are similar to that of a closed-form kernel regression problem obtained with the eNTK of the network. Hence, assuming that the network is initialized near the kernel regime, it needs $\Omega(p^2)$ samples to generalize.

Figures 1 and 4 show that in practice, our network seems to be operating "close" to the kernel regime until shortly before generalization occurs, when the NTK changes drastically.

To further evaluate our hypothesis, in Figure 1 we change the scale of initialization, and observe that smaller initialization scales mitigate the gap between train and test accuracy throughout the training.

5 Discussion

Theorems 1 and 2, in combination with our various empirical results, suggest that grokking on modular arithmetic problems (at least with quadratic networks) can be explained as the transition from the kernel regime to margin maximization.

Overall, this further supports our hypothesis that grokking is a result of natural training dynamics of the training algorithm used that is "influenced" by different explicit or implicit regularization, as opposed to being a direct result of different regularizations involved in training.

References

Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. *Advances in neural information processing systems*, 32, 2019.

Boaz Barak, Benjamin Edelman, Surbhi Goel, Sham Kakade, Eran Malach, and Cyril Zhang. Hidden progress in deep learning: Sgd learns parities near the computational limit. *Advances in Neural Information Processing Systems*, 35:21750–21764, 2022.

Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. In *NeurIPS*, 2019.

Andrey Gromov. Grokking modular arithmetic, 2023.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

[6] Neurons of the last layer are duplicated such that the new neurons having the same input weights and opposite output weights as the old ones, ensuring that the NN output is zero. For more details, please refer to Hu et al. (2019), Appendix A.

Wei Hu, Zhiyuan Li, and Dingli Yu. Simple and effective regularization methods for training on noisily labeled data with generalization guarantee. *arXiv preprint arXiv:1905.11368*, 2019.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. *Advances in neural information processing systems*, 32, 2019.

Zhiyuan Li, Yi Zhang, and Sanjeev Arora. Why are convolutional nets more sample-efficient than fully-connected nets? *arXiv preprint arXiv:2010.08515*, 2020.

Ziming Liu, Ouail Kitouni, Niklas S Nolte, Eric Michaud, Max Tegmark, and Mike Williams. Towards understanding grokking: An effective theory of representation learning. *Advances in Neural Information Processing Systems*, 35:34651–34663, 2022a.

Ziming Liu, Eric J Michaud, and Max Tegmark. Omnigrok: Grokking beyond algorithmic data. *arXiv preprint arXiv:2210.01117*, 2022b.

Kaifeng Lyu and Jian Li. Gradient descent maximizes the margin of homogeneous neural networks. In *ICLR*, 2020.

David A. McAllester. Simplified pac-bayesian margin bounds. In *Annual Conference Computational Learning Theory*, 2003. URL https://api.semanticscholar.org/CorpusID:14620324.

Mohamad Amin Mohamadi, Wonho Bae, and Danica J Sutherland. A fast, well-founded approximation to the empirical neural tangent kernel. In *International Conference on Machine Learning*, pp. 25061–25081. PMLR, 2023.

Neel Nanda, Lawrence Chan, Tom Liberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. *arXiv preprint arXiv:2301.05217*, 2023.

Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. A PAC-bayesian approach to spectrally-normalized margin bounds for neural networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Skz_WfbCZ.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets, 2022.

Vimal Thilak, Etai Littwin, Shuangfei Zhai, Omid Saremi, Roni Paiss, and Joshua Susskind. The slingshot mechanism: An empirical study of adaptive optimizers and the grokking phenomenon. *arXiv preprint arXiv:2206.04817*, 2022.

Martin J. Wainwright. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge University Press, 2019.

Greg Yang and Edward J. Hu. Tensor programs iv: Feature learning in infinite-width neural networks. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 11727–11737. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/yang21c.html.

A Experimental Setup

In this section we briefly explain the setup used for our experimental evaluations. In all experiments, we have used *vanilla* gradient descent with cross-entropy or squared loss, depending on the context of the experiment, for 100,000 up to 500,000 steps. To accelerate the training wih cross-entropy loss, we use the "normalized" GD trick, where the learning rate of each step is scaled by the inverse of the norm of the gradient:

$$\theta_{t+1} = \theta_t - \lambda \frac{\nabla_\theta \ell(\theta_t)}{\|\nabla_\theta \ell(\theta_t)\|_2} \tag{8}$$

where ℓ denotes the loss function and λ denotes the learning rate. The learning rate in the presented experiments was set to 0.1 and was kept constant during the training.

We used the JAX framework to implement and run the experiments on V100 or A100 GPU machines. To enable the computation of the empirical NTKs with limited memory requirements, we used Mohamadi et al. (2023)'s pseudo-NTK approximation.

B Construction

In this section, we present a construction of weights that interpolates the dataset. We begin by constructing 8 matrices of size $p \times 2p$ denoted by $W^{(i)}$.

$$\begin{aligned}
W^{(1)}_{k,(n,m)} &= \begin{pmatrix} \cos\left(\frac{2\pi k}{p} n\right) & \cos\left(\frac{2\pi k}{p} m\right) \end{pmatrix} \\
W^{(2)}_{k,(n,m)} &= \begin{pmatrix} \cos\left(\frac{2\pi k}{p} n\right) & -\cos\left(\frac{2\pi k}{p} m\right) \end{pmatrix} \\
W^{(3)}_{k,(n,m)} &= \begin{pmatrix} \sin\left(\frac{2\pi k}{p} n\right) & \sin\left(\frac{2\pi k}{p} m\right) \end{pmatrix} \\
W^{(4)}_{k,(n,m)} &= \begin{pmatrix} \sin\left(\frac{2\pi k}{p} n\right) & -\sin\left(\frac{2\pi k}{p} m\right) \end{pmatrix} \\
W^{(5)}_{k,(n,m)} &= \begin{pmatrix} \sin\left(\frac{2\pi k}{p} n\right) & \cos\left(\frac{2\pi k}{p} m\right) \end{pmatrix} \\
W^{(6)}_{k,(n,m)} &= \begin{pmatrix} \sin\left(\frac{2\pi k}{p} n\right) & -\cos\left(\frac{2\pi k}{p} m\right) \end{pmatrix} \\
W^{(7)}_{k,(n,m)} &= \begin{pmatrix} \cos\left(\frac{2\pi k}{p} n\right) & \sin\left(\frac{2\pi k}{p} m\right) \end{pmatrix} \\
W^{(8)}_{k,(n,m)} &= \begin{pmatrix} -\cos\left(\frac{2\pi k}{p} n\right) & \sin\left(\frac{2\pi k}{p} m\right) \end{pmatrix}
\end{aligned} \tag{9}$$

$$V_{q,k} = \begin{pmatrix} \cos\left(\frac{2\pi k}{p} q\right) \\ -\cos\left(\frac{2\pi k}{p} q\right) \\ -\cos\left(\frac{2\pi k}{p} q\right) \\ \cos\left(\frac{2\pi k}{p} q\right) \\ \sin\left(\frac{2\pi k}{p} q\right) \\ -\sin\left(\frac{2\pi k}{p} q\right) \\ \sin\left(\frac{2\pi k}{p} q\right) \\ -\sin\left(\frac{2\pi k}{p} q\right) \end{pmatrix}^\top . \tag{10}$$

The construction of the first layer is to stack $W^{(i)}$ for $i \in [1, 8]$ to construct $W \in \mathbb{R}^{8p \times 2p}$. The weights of the second layer are given in Equation (10).

To show that this construction solves the modular addition problem analytically, we will analytically perform the inference step for two arbitrary inputs n, m. We denote $h(x) = (Wx)^2 \in \mathbb{R}^{8p}$ as the post-activations of the first layer, which is given by (after dropping x for simplicity)

$$h_{8k:8(k+1)} = \begin{pmatrix} \cos\left(\frac{2\pi k}{p} n\right) + \cos\left(\frac{2\pi k}{p} m\right) \\ \cos\left(\frac{2\pi k}{p} n\right) - \cos\left(\frac{2\pi k}{p} m\right) \\ \sin\left(\frac{2\pi k}{p} n\right) + \sin\left(\frac{2\pi k}{p} m\right) \\ \sin\left(\frac{2\pi k}{p} n\right) - \sin\left(\frac{2\pi k}{p} m\right) \\ \sin\left(\frac{2\pi k}{p} n\right) + \cos\left(\frac{2\pi k}{p} m\right) \\ \sin\left(\frac{2\pi k}{p} n\right) - \cos\left(\frac{2\pi k}{p} m\right) \\ \cos\left(\frac{2\pi k}{p} n\right) + \sin\left(\frac{2\pi k}{p} m\right) \\ \cos\left(\frac{2\pi k}{p} m\right) - \sin\left(\frac{2\pi k}{p} n\right) \end{pmatrix}^2 . \tag{11}$$

Note that for each k, we have that

$$h_{8k} - h_{8k+1} = \cos\left(\frac{2\pi k}{p}2n\right) + \cos\left(\frac{2\pi k}{p}2m\right) + 2\cos\left(\frac{2\pi k}{p}(n+m)\right) \tag{12}$$

and

$$h_{8k+2} - h_{8k+3} = \cos\left(\frac{2\pi k}{p}2n\right) + \cos\left(\frac{2\pi k}{p}2m\right) - 2\cos\left(\frac{2\pi k}{p}(n+m)\right) \tag{13}$$

and

$$h_{8k+4} - h_{8k+5} = 4\sin\left(\frac{2\pi k}{p}n\right)\cos\left(\frac{2\pi k}{p}m\right) \tag{14}$$

and

$$h_{8k+6} - h_{8k+7} = 4\cos\left(\frac{2\pi k}{p}n\right)\sin\left(\frac{2\pi k}{p}m\right). \tag{15}$$

Hence,

$$h_{8k} - h_{8k+1} - h_{8k+2} + h_{8k+3} = 4\cos\left(\frac{2\pi k}{p}(n+m)\right) \tag{16}$$

and

$$h_{8k+4} - h_{8k+5} + h_{8k+6} - h_{8k+7} = 4\sin\left(\frac{2\pi k}{p}(n+m)\right). \tag{17}$$

Using the fact that $\cos(a-b) = \cos(a)\cos(b) - \sin(a)\sin(b)$, we can see that

$$\begin{aligned}
\langle V_{q,:}, h(x)\rangle &= 4\sum_{k=0}^{p-1}\cos\left(\frac{2\pi k}{p}q\right)\cos\left(\frac{2\pi k}{p}(n+m)\right) - \sin\left(\frac{2\pi k}{p}q\right)\sin\left(\frac{2\pi k}{p}(n+m)\right) \\
&= 4\sum_{k=0}^{p-1}\cos\left(\frac{2\pi k}{p}(m+n-q)\right) \\
&= 4p\,\delta\left((m+n-q) \bmod p = 0\right)
\end{aligned} \tag{18}$$

where the last equality follows from Euler's identity.

Remark 3. *We need at most $4p$ hidden neurons to interpolate the modular addition task.*

Observing the fact that $\cos(2\pi - a) = \cos(a)$, we can see that

$$\sum_{k=0}^{p-1}\cos\left(\frac{2\pi k}{p}(m+n-q)\right) = 1 + 2\sum_{k=1}^{\lfloor p/2\rfloor}\cos\left(\frac{2\pi k}{p}(m+n-q)\right) \tag{19}$$

where we replaced $\cos\left(\frac{2\pi 0}{p}(m+n-q)\right)$ with 1. Based on Equation (19), we can cut out half of the weights of the first and second layer, and only construct the frequencies up to $\lfloor p/2\rfloor$, which results in only needing $4p$ hidden neurons to construct the interpolating solution.

C Margin-Based Generalization Bound

We begin by providing some background and notation on sub-exponential random variables, which will be later used in the proof of our margin-based generalization bound.

C.1 Background on sub-exponential variables

The following proofs rely heavily on concentration inequalities for sub-exponential random variables; we will first review some background on these quantities.

A real-valued random variable X with mean μ is called *sub-exponential* (see e.g. Wainwright, 2019) if there are non-negative parameters (ν, α) such that

$$\mathbb{E}[e^{\lambda(X-\mu)}] \leq e^{\frac{\nu^2\lambda^2}{2}} \quad \text{for all } |\lambda| < \frac{1}{\alpha}. \tag{20}$$

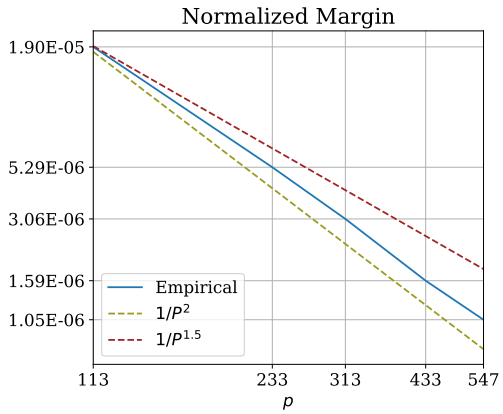

Figure 5: Normalized margin for different values of p after training with ce loss for 500,000 steps of normalized GD.

We use $X \sim SE(\nu, \alpha)$ to denote that X is a sub-exponential random variable with parameters (ν, α), but note that this is not a particular distribution.

One famous sub-exponential random variable is the product of the absolute value of two standard normal distributions, $z_i \sim \mathcal{N}(0, 1)$, such that the two factors are either independent ($X_1 = |z_1||z_2| \sim SE(\nu_p, \alpha_p)$ with mean $2/\pi$) or the same ($X_2 = z^2 \sim SE(2, 4)$ with mean 1). We now present a few lemmas regarding sub-exponential random variables that will come in handy in the later subsections of the appendix.

Lemma 4. *If a random variable X is sub-exponential with parameters (ν, α), then the random variable sX where $s \in \mathbb{R}^+$ is also sub-exponential with parameters $(s\nu, s\alpha)$.*

Proof. Consider $X \sim SE(\nu, \alpha)$ and $X' = sX$ with $\mathbb{E}[X'] = s\mathbb{E}[X]$, then according to the definition of a sub-exponential random variable

$$\begin{aligned} &\mathbb{E}\left[\exp\left(\lambda(X-\mu)\right)\right] \leq \exp(\frac{\nu^2\lambda^2}{2}) \quad \text{for all } |\lambda| < \frac{1}{\alpha} \\ &\implies \mathbb{E}\left[\exp\left(\frac{\lambda}{s}(sX - s\mu)\right)\right] \leq \exp(\frac{\nu^2 s^2 \frac{\lambda^2}{s^2}}{2}) \quad \text{for all } |\frac{\lambda}{s}| < \frac{1}{s\alpha} \\ &\xrightarrow{\lambda' = \frac{\lambda}{s}} \mathbb{E}\left[\exp\left(\lambda'(X' - \mu')\right)\right] \leq \exp(\frac{\nu^2 s^2 \lambda'^2}{2}) \quad \text{for all } |\lambda'| < \frac{1}{s\alpha} \end{aligned} \tag{21}$$

Defining $\alpha' = s\alpha$ and $\nu' = s\nu$ we recover that $X' \sim SE(s\nu, s\alpha)$. □

Proposition 5. *If the random variables X_i for $i \in [1 - N]$ for $N \in \mathbb{N}^+$ are all sub-exponential with parameters (ν_i, α_i) and independent, then $\sum_{i=1}^N X_i \in SE(\sqrt{\sum_{i=1}^N \nu_i^2}, \max_i \alpha_i)$, and $\frac{1}{N}\sum_{i=1}^N X_i \sim SE\left(\frac{1}{\sqrt{N}}\sqrt{\frac{1}{N}\sum_{i=1}^N \nu_i^2}, \frac{1}{N}\max_i \alpha_i\right)$.*

Proof. This is a simplification of the discussion prior to equation 2.18 of Wainwright (2019). □

Proposition 6. *For a random variable $X \sim SE(\nu, \alpha)$, the following concentration inequality holds:*

$$\Pr\left(|X - \mu| \geq t\right) \leq 2\exp\left(-\min\left(\frac{t^2}{2\nu^2}, \frac{t}{2\alpha}\right)\right).$$

Proof. Direct from multiplying the result derived in Equation 2.18 of Wainwright (2019) by a scalar. □

Corollary 7. *For a random variable $X \sim SE(\nu, \alpha)$, the following inequality holds with probability at least $1-\delta$:*

$$|X-\mu| < \max\left(\nu\sqrt{2\log\frac{2}{\delta}}, 2\alpha\log\frac{2}{\delta}\right).$$

A sub-Gaussian random variable, $SG(\nu)$, is one which satisfies equation 20 for all λ, i.e. it is the limit of $SE(\nu,\alpha)$ as $\alpha \to 0$.

Proposition 8 (Chernoff bound)**.** *If X is $SG(\nu)$, then with probability at least $1-\delta$, $|X-\mu| \leq \nu\sqrt{2\log\frac{2}{\delta}}$.*

Proposition 9 (Hoeffding's inequality)**.** *If $X_1, \ldots, X_n$ are independent variables with means μ_i and each $SG(\nu_i)$, then $|\sum_{i=1}^n X_i - \sum_{i=1}^n \mu_i| \leq \sqrt{2\left(\sum_{i=1}^n \nu_i^2\right)\log\frac{2}{\delta}}$ with probability at least $1-\delta$.*

C.2 Generalization Bound

We are now ready to state the main theorem for proving an upper bound on the number of training points needed to generalize.

Theorem 10. *Let $f : \mathbb{R}^{2p} \to \mathbb{R}^p$ be a two-layer network defined in Equation (1) with $h = \mathcal{O}(p)$ hidden neurons and quadratic activation, parameterized by $\theta = vec\,(W, V)$ where $W \in \mathbb{R}^{h\times 2p}$ and $V \in \mathbb{R}^{p\times h}$. For any $\delta > 0$, it holds with probability at least $1-\delta$ over the choice of training set $\mathcal{D}_{train}$ of size m that, for all θ satisfying **1)** $\|W\|_\infty = \mathcal{O}(1)$, **2)** $\|V\|_\infty = \mathcal{O}(1)$ and **3)** $\sum_i^h V_{qi} = 0$ for all $q \in [p]$,*

$$L_0(f,\mathcal{D}) \leq L_p(f,\mathcal{D}_{train}) + \tilde{\mathcal{O}}\left(\sqrt{\frac{p^{5/3}}{m}}\right). \tag{22}$$

Proof. We'll start by obtaining high probability bounds over the output logits of the network after perturbing each scalar weight with $\mathcal{N}(0,\sigma^2)$ noise. Let x represent the two-hot vector corresponding to inputs m and n and f_q represnet the q'th output logit of the neural network f. $\tilde{W}$ and $\tilde{V}$ denote the perturbation noises. Moreover, assume $C_V = \|V\|_\infty$ and $C_W = \|W\|_\infty$ are positive constants.

$$\begin{aligned}\tilde{f}_q\left(W+\tilde{W}, V+\tilde{V}; x\right) &= \left(V_q+\tilde{V}_q\right)\left(\left(W+\tilde{W}\right)x\right)^{\odot 2} \\ &= f_q\left(W,V;x\right) + V_q\left(\left(\tilde{W}x\right)^{\odot 2} + 2\,Wx\odot\tilde{W}x\right) + \tilde{V}_q\left(\left(W+\tilde{W}\right)x\right)^{\odot 2}\end{aligned} \tag{23}$$

Starting with the first term, we can see that

$$\begin{aligned}V_q\left(\tilde{W}x\right)^{\odot 2} = \sum_{i=1}^h V_{qi}\left(\tilde{W}_{im}+\tilde{W}_{in}\right)^{\odot 2} &= 2\sigma^2\sum_{i=1}^h V_{qi}\,\chi_i \\ &\sim \sqrt{2}\sigma^2 C_V\, SE(2\sqrt{h}, 4)\end{aligned} \tag{24}$$

where $\left(\tilde{W}_{im}+\tilde{W}_{in}\right)^{\odot 2}$ is distributed as a chi-squared random variable denoted by χ_i (as the second power of sum of two i.i.d Gaussian random variables each having a variance of σ^2). $\chi^2(1)$ is a sub-exponential random variable with parameters $SE(2,4)$ with mean 1. Based on assumption **3**, we can see that the sum has a zero mean . We can further apply Corollary 7 to show that with probability at least $1-\delta_1$ over randomness of perturbation

$$\left|V_q\left(\tilde{W}x\right)^{\odot 2}\right| \leq \sigma^2\max\left(4C_V\sqrt{h\log\frac{2}{\delta_1}}, 4\sqrt{2}\log\frac{2}{\delta_1}\right). \tag{25}$$

For the second term, we can see that

$$2V_q\, Wx \odot \tilde{W}x = 2\sum_{i=1}^{h} V_{qi}(W_{im} + W_{in})(\tilde{W}_{im} + \tilde{W}_{in}) = 2\sqrt{2}\sigma \sum_{i=1}^{h} V_{qi}(W_{im} + W_{in})\mathcal{N}_i \quad (26)$$

where $(\tilde{W}_{im} + \tilde{W}_{in})$ is distributed as a Gaussian with $\mathcal{N}(0, 2)$ parameters and is replaced with $\sqrt{2}\mathcal{N}_i$ where $\mathcal{N}_i \sim \mathcal{N}(0, 1)$. Once again, we can see that this sum has a zero mean. Applying Proposition 9 on this sum we can show that with probability at least $1 - \delta_2$ over randomness of perturbation

$$\left|2V_q\, Wx \odot \tilde{W}x\right| \leq 2\sigma C_V C_W C_2 \sqrt{2h \log \frac{2}{\delta_2}} \quad (27)$$

where C_2 is a positive constant.

Accordingly, we can decompose the second term into three sums. For the first component, we can see that

$$\tilde{V}_q\, (Wx)^2 \sim \mathcal{N}\left(0, \sigma^2 \left\|(Wx)^2\right\|_2^2\right). \quad (28)$$

Applying the Proposition 8 on this Gaussian random variable, one can see that with probability at least $1 - \delta_3$ over randomness of perturbation

$$\left|\tilde{V}_q\, (Wx)^2\right| \leq 4\sigma \sqrt{h \log \frac{2}{\delta_3}} \quad (29)$$

and for the second term in the decomposition we can see that

$$2\tilde{V}_q\, Wx \odot \tilde{W}x = 2\sum_{i=1}^{h} \tilde{V}_{qi}(W_{im} + W_{in})(\tilde{W}_{im} + \tilde{W}_{in}) \sim 2\sqrt{2}\sigma^2 \sum_{i=1}^{h}(W_{im} + W_{in})\mathcal{N}_i^{(1)}\mathcal{N}_i^{(2)} \quad (30)$$

where $\mathcal{N}_i^{(1)}$ and $\mathcal{N}_i^{(2)}$ are random variables distributed as $\mathcal{N}(0, 1)$. Note that the sum has a zero mean, the product of two independent Gaussian distributions can be written as the sum of two chi-squared distributions. Applying this technique and Corollary 7 we can see that with probability at least $1 - \delta_4$ over randomness of perturbation

$$\left|2\tilde{V}_q\, Wx \odot \tilde{W}x\right| \leq C_3 C_W \sigma^2 \max\left(\sqrt{h \log \frac{2}{\delta_4}}, \log \frac{2}{\delta_4}\right) \quad (31)$$

where C_3 is a positive constant. Finally, for the last term, we can show that,

$$\tilde{V}_q \left(\tilde{W}x\right)^2 = \sum_{i=1}^{h} \tilde{V}_{qi} \left(\tilde{W}_{im} + \tilde{W}_{in}\right)^2 \sim \sqrt{2}\sigma^3 \sum_{i=1}^{h} \mathcal{N}_i \chi_i \quad (32)$$

where χ_i is a random variable distributed according to $\chi^2(1)$ and $\mathcal{N}_i$ is a random variable distributed according to $\mathcal{N}(0, 1)$. The sum has mean zero. To bound this sum, we can first treat the chi-squared variables as bounded random variables with high probability, and pull them out of the sum. Then, we can apply the Hoeffding's inequality to bound the sum of Gaussians. Note that for each $y_i \sim \chi^2(1)$, we have that

$$\Pr\left[|y_i - 1| < C_4 \max\left(\sqrt{2 \log \frac{2}{\delta}}, \log \frac{2}{\delta}\right)\right] \geq 1 - \delta \quad (33)$$

where C_4 is a positive constant. Applying a union bound on all y_i for $i \in [h]$ we have that

$$\Pr\left[\forall i \in [h]; |y_i - 1| < C_4 \max\left(\sqrt{2 \log \frac{2h}{\delta}}, \log \frac{2h}{\delta}\right)\right] \geq 1 - \delta. \quad (34)$$

Pulling this out of the sum, we can see that

$$\Pr\left[\left|\tilde{V}_q\left(\tilde{W}\right)^2 x\right| \leq \sqrt{2}C_4'\sigma^3 \log\frac{2h}{\delta}\sum_{i=1}^{h}\mathcal{N}(0,1)\right] \geq 1-\delta \tag{35}$$

where C_4' is a positive constant. Applying Proposition 9 to bound the sum of Gaussians, we can show that

$$\Pr\left[\left|\tilde{V}_q\left(\tilde{W}\right)^2 x\right| \leq \sqrt{2}C_4'\sigma^3 \log\frac{2h}{\delta}\sqrt{2h\log\frac{2}{\delta'}}\right] \geq 1-\delta-\delta'. \tag{36}$$

Combining the two high probability events, we can conclude that with probability at least $1-\delta_5$ over perturbation noise

$$\left|\tilde{V}_q\left(\tilde{W}x\right)^2\right| \leq C_4''\sigma^3\sqrt{h}\left(\log\frac{2h}{\delta_5}\right)^{3/2} \tag{37}$$

where C_4'' is a positive constant.

Applying a union bound on $\delta_1, \cdots, \delta_5$, and then another union bound on each logit, we can show that for each x,

$$\max_q \left|\tilde{f}_q(x) - f_q(x)\right| \leq \tilde{\mathcal{O}}\left(\sqrt{h}\sigma^3\left(\log\frac{2h}{\delta}\right)^{3/2}\right) \tag{38}$$

with probability at least $1-\delta$ over the perturbation noise.

Since p is the margin of solution achieved with W, V, applying Lemma 1 from Neyshabur et al. (2018) with $\sigma^2 = h^{\frac{1}{3}}$ concludes the proof.

□

Proposition 11. *Condition **3** from Theorem 10 which implies that* $\forall q, \sum_{i=1}^{h} V_{qi} = 0$ *is not necessary.*

Proof. Assume we have a network with weights $\theta = (W, V)$ that satisfies all conditions of Theorem 10 except $\forall q, \sum_{i=1}^{h} V_{qi} = 0$. We can construct a new network with weights $\theta' = (W', V')$ such that $W' = \begin{bmatrix} W \\ W \end{bmatrix}$ and $V' = [V \quad -V]$. This network has the same outputs as the original one with parameters θ, while each row in V' has a zero sum. Hence, Theorem 10 shows that the constructed network follows the provided genearlization bound, which subsequently shows that the original network with parameters θ does so too, since the outputs of these two networks are exactly identical.

□

D Kernel-Based Generalization Bound

In this section we present the formal version of Theorem 2 alongside a proof of it. Note that a kernel-based predictor h on a training data $\{(x_i, y_i)\}_{i=1}^n$ can be expressed as $h(x) = \sum_{i=1}^n \lambda_i K(x_i, x)$ where $\lambda_i; i \in [n]$ are constants. Assuming that the kernel's feature maps are of dimension d, the predictions are linear combinations of d-dimensional feature maps. We first present a general proof that every permutation invariant kernel requires $\Omega(p^2)$ training points to outperform the null predictor in terms of ℓ_2 loss, and then show that this theorem applies to the distribution of empirical NTKs at initialization.

Notations: We use $[p]$ to denote the set $\{1, \ldots, p\}$. We use S_p to denote the permutation groups over p elements and id is the identity mapping from $[p]$ to $[p]$. For any nonempty set $\mathcal{X}$, a symmetric function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a positive semi-definite kernel (p.s.d) kernel on $\mathcal{X}$ if for any $n \in \mathbb{N}$, any $x_1, \ldots, x_n$ and $\lambda_1, \ldots, \lambda_n \in \mathbb{R}$, it holds that $\sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j K(x_i, x_j) \geq 0$. For a subspace V of $\mathbb{R}^n$ and vector $x \in \mathbb{R}^n$, we define $\mathsf{dist}(x, V) \triangleq \min_{v \in V} \|x - v\|_2$.

Lemma 12. *For any subspace V of $\mathbb{R}^n$ and vector $x \in \mathbb{R}^n$, let $\{v_i\}_{i=1}^m$ be an orthonormal basis of V, it holds that $\mathsf{dist}^2(x, V) = \|x\|_2^2 - \sum_{i=1}^m \langle x, v_i \rangle^2$.*

The proof of Lemma 12 is straightforward and thus omitted.

For the ease of presentation, we will use the (i, j) and $e_i + e_{j+p}$ interchangebly for $i, j \in [p]$. For any $c \in [p]$, we define the population ℓ_2 loss on the cth logit of predictor $h : [p] \times [p] \to \mathbb{R}$ as

$$L^c(h) \triangleq \mathbb{E}_{i,j \sim [p]} \left(h(i, j) - \mathbf{1}_{i+j \equiv c \pmod p} \right)^2. \tag{39}$$

Definition 13 (Permutation Invariant Kernels)**.** We say a p.s.d. kernel K on $[p] \times [p]$ is *permutation invariant* iff for any permutation $\sigma_1, \sigma_2 \in S_p$, it holds that for all $i, j, k, l \in [p]$,

$$K((i, j), (k, l)) = K((\sigma_1(i), \sigma_2(j)), (\sigma_1(k), \sigma_2(l))).$$

We denote each function $K(x_t, \cdot) : [p] \times [p] \to \mathbb{R}$ as a matrix on $\mathbb{R}^{p \times p}$ by $v_t(\cdot)$. We also define function $h^c_{\sigma_1, \sigma_2}(i, j) \triangleq \mathbf{1}_{\sigma_1(i) + \sigma_2(j) \equiv c \pmod p}$. We can view a function mapping from $[p] \times [p] \to \mathbb{R}$ as a vector on $\mathbb{R}^{p \times p}$ and define inner products and dist on the function space, *i.e.*, $\langle h, h' \rangle \triangleq \sum_{i,j \in [p]} h(i, j) h'(i, j)$ and $\|h\|_2^2 \triangleq \langle h, h \rangle$.

Lemma 14. *For any integers $p, n \geq 1$, $c \in [p]$, permutation groups $\sigma_1, \sigma_2 \in S_p$ and permutation invariant kernel K on $[p] \times [p]$, suppose $x_t = (i_t, j_t) \overset{i.i.d.}{\sim} \mathrm{Unif}([p] \times [p])$ for each $t \in [n]$, it holds that*

$$\mathbb{E}_{x_1, \ldots, x_n} \inf_{\lambda_1, \ldots, \lambda_n \in \mathbb{R}} \left\| \sum_{t=1}^n \lambda_t K(x_t, \cdot) - h^c_{\mathsf{id}, \mathsf{id}} \right\|_2^2 = \mathbb{E}_{x_1, \ldots, x_n} \inf_{\lambda_1, \ldots, \lambda_n \in \mathbb{R}} \left\| \sum_{t=1}^n \lambda_t K(x_t, \cdot) - h^c_{\sigma_1, \sigma_2} \right\|_2^2 \tag{40}$$

Proof of Lemma 14. Note that $x_i \overset{i.i.d.}{\sim} \mathrm{Unif}([p], [p])$, for any permutation $\sigma \in [p]$, we have $\sigma(x_i) \overset{i.i.d.}{\sim} \mathrm{Unif}([p], [p])$. Thus we have that

$$\begin{aligned} &\mathbb{E}_{x_1, \ldots, x_n} \inf_{\lambda_1, \ldots, \lambda_n \in \mathbb{R}} \left\| \sum_{t=1}^n \lambda_t K(x_t, \cdot) - h^c_{\sigma_1, \sigma_2} \right\|_2^2 \\ =&\mathbb{E}_{x_1, \ldots, x_n} \inf_{\lambda_1, \ldots, \lambda_n \in \mathbb{R}} \left\| \sum_{t=1}^n \lambda_t K((\sigma_1^{-1}(i_t), \sigma_2^{-1}(j_t)), \cdot) - h^c_{\sigma_1, \sigma_2} \right\|_2^2 \end{aligned} \tag{41}$$

Applying the same argument again, we have that

$$\begin{aligned} &\left\| \sum_{t=1}^n \lambda_t K((\sigma_1^{-1}(i_t), \sigma_2^{-1}(j_t)), \cdot) - h^c_{\sigma_1, \sigma_2} \right\|_2^2 \\ =& \sum_{i,j \in [p]} \sum_{t=1}^n \left(\lambda_t K((\sigma_1^{-1}(i_t), \sigma_2^{-1}(j_t)), (i, j)) - h^c_{\sigma_1, \sigma_2}(i, j) \right)^2 \\ =& \sum_{i,j \in [p]} \sum_{t=1}^n \left(\lambda_t K((\sigma_1^{-1}(i_t), \sigma_2^{-1}(j_t)), (\sigma_1^{-1}(i), \sigma_2^{-1}(j))) - h^c_{\sigma_1, \sigma_2}(\sigma_1^{-1}(i), \sigma_2^{-1}(j)) \right)^2 \\ =& \sum_{i,j \in [p]} \sum_{t=1}^n \left(\lambda_t K((i_t, j_t), (i, j)) - h^c_{\mathsf{id}, \mathsf{id}}(i, j) \right)^2 = \left\| \sum_{t=1}^n \lambda_t K(x_t, \cdot) - h^c_{\mathsf{id}, \mathsf{id}} \right\|_2^2, \end{aligned} \tag{42}$$

which completes the proof.

□

Theorem 15. *For any integers $p, n \geq 1$, $c \in [p]$ and permutation invariant kernel K on $[p] \times [p]$, suppose $x_t = (i_t, j_t) \overset{i.i.d.}{\sim} \mathrm{Unif}([p] \times [p])$ for each $t \in [n]$, it holds that*

$$\mathbb{E}_{x_1,\ldots,x_n} \inf_{\lambda_1,\ldots,\lambda_n \in \mathbb{R}} L^c\left(\sum_{t=1}^n \lambda_t K(x_t, \cdot)\right) \geq \frac{1}{p}\left(1 - \frac{1}{p} - \frac{n}{p(p-1)}\right) \tag{43}$$

In other words, if $n \leq (1 - \Omega(1))p^2$, then expected population ℓ_2 loss is at least $\Omega(1/p)$ for each coordinate $c \in [p]$, which is of the same magnitude of the trivial all-zero predictor.

Proof of Theorem 15. Note that $\inf_{\lambda_1,\ldots,\lambda_n \in \mathbb{R}} L^c\left(\sum_{t=1}^n \lambda_t K(x_t, \cdot)\right) = \frac{1}{p^2}\mathsf{dist}^2(V, h^c_{\mathsf{id},\mathsf{id}})$, the target inequality Equation (43) is equivalent to Equation (44).

$$\mathbb{E}_{x_1,\ldots,x_n}\mathsf{dist}^2(V, h^c_{\mathsf{id},\mathsf{id}}) \geq p - 1 - \frac{n}{p-1}, \tag{44}$$

where $V = \{\sum_{t=1}^n \lambda_t v_t \mid \lambda_t \in \mathbb{R}\}$ is the subspace spanned by $\{v_t\}_{t=1}^n$. By Lemma 14, it holds that

$$\begin{aligned}\mathbb{E}_{x_1,\ldots,x_n}\mathsf{dist}^2(V, h^c_{\mathsf{id},\mathsf{id}}) =& \mathbb{E}_{\sigma_1,\sigma_2\sim\mathrm{Unif}(S_p)}\mathbb{E}_{x_1,\ldots,x_n}\mathsf{dist}^2(V, h^c_{\sigma_1,\sigma_2}) &(45)\\ =& \mathbb{E}_{x_1,\ldots,x_n}\mathbb{E}_{\sigma_1,\sigma_2\sim\mathrm{Unif}(S_p)}\mathsf{dist}^2(V, h^c_{\sigma_1,\sigma_2}). &(46)\end{aligned}$$

Now we claim that for any n-dimensional subspace $V \subset \mathbb{R}^{p\times p}$, it holds that

$$\mathbb{E}_{\sigma_1,\sigma_2\sim\mathrm{Unif}(S_p)}\mathsf{dist}^2(V, h^c_{\sigma_1,\sigma_2}) \geq p - 1 - \frac{n}{p-1}. \tag{47}$$

If Equation (47) holds, then Equation (44) holds and we are done. Below we will prove Equation (47). We define $V' = V + \{\lambda_0 \mathbf{1}_{[p]\times[p]} \mid \lambda \in \mathbb{R}\}$ as a larger subspace containing V and the constant function $\mathbf{1}_{[p]\times[p]}$, where for any $i, j \in [p]$, it holds that $\mathbf{1}_{[p]\times[p]}(i, j) = 1$. By definition of V', we have

$$\mathbb{E}_{\sigma_1,\sigma_2\sim\mathrm{Unif}(S_p)}\mathsf{dist}^2(V, h^c_{\sigma_1,\sigma_2}) \geq \mathbb{E}_{\sigma_1,\sigma_2\sim\mathrm{Unif}(S_p)}\mathsf{dist}^2(V', h^c_{\sigma_1,\sigma_2}), \tag{48}$$

where $v'_0 = \frac{1}{p}\mathbf{1}_{[p]\times[p]}$ and $\{v'_t\}_{t=0}^n$ are a orthonormal basis of V'. Clearly $\left\langle v'_0, h^c_{\sigma_1,\sigma_2}\right\rangle = 1$, $\left\|h^c_{\mathsf{id},\mathsf{id}}\right\|_2^2 = p$. Thus by Lemma 12, it holds that

$$\begin{aligned}&\mathbb{E}_{\sigma_1,\sigma_2\sim\mathrm{Unif}(S_p)}\mathsf{dist}^2(V, h^c_{\sigma_1,\sigma_2}) &(49)\\ =&\mathbb{E}_{\sigma_1,\sigma_2\sim\mathrm{Unif}(S_p)}\left(\left\|h^c_{\sigma_1,\sigma_2}\right\|_2^2 - \sum_{t=0}^n \left\langle v'_t, h^c_{\sigma_1,\sigma_2}\right\rangle^2\right)\\ =&\mathbb{E}_{\sigma_1,\sigma_2\sim\mathrm{Unif}(S_p)}\left(p - 1 - \sum_{t=1}^n \left\langle v'_t, h^c_{\sigma_1,\sigma_2}\right\rangle^2\right)\\ \geq& p - 1 - n \cdot \sup_{\|v\|=1, \left\langle v, \mathbf{1}_{[p]\times[p]}\right\rangle = 0} \mathbb{E}_{\sigma_1,\sigma_2\sim\mathrm{Unif}(S_p)}\left\langle v, h^c_{\sigma_1,\sigma_2}\right\rangle^2, &(50)\end{aligned}$$

where the last step is because the subspace V' only depends on $x_1, \ldots, x_n$ but not σ_1, σ_2. Further note that $h^c_{\sigma_1,\sigma_2}(i, j) = 1 \iff \sigma_1(i) + \sigma_2(j) \equiv c \pmod p \iff j \equiv \sigma_2^{-1}(c - \sigma_1(i))$ and

$\sigma_2^{-1} \circ (c - \sigma_1)$ is also uniformly distributed in S_p for any $c \in [p]$, we have that

$$
\begin{aligned}
&\mathbb{E}_{\sigma_1,\sigma_2\sim\mathrm{Unif}(S_p)} \left\langle v, h^c_{\sigma_1,\sigma_2} \right\rangle^2 \\
=&\mathbb{E}_{\sigma\sim\mathrm{Unif}(S_p)} \left(\sum_{i\in[p]} v(i,\sigma(i)) \right)^2 \\
=&\mathbb{E}_{\sigma\sim\mathrm{Unif}(S_p)} \sum_{i\in[p]} v^2(i,\sigma(i)) + \sum_{\substack{i,j\in[p]\\ i\neq j}} v(i,\sigma(i))v(j,\sigma(j)) \\
=&\frac{1}{p} \sum_{i,j\in[p]} v(i,j)^2 + \frac{1}{p(p-1)} \sum_{\substack{i,j,k,l\in[p]\\ i\neq k, j\neq l}} v(i,j)v(k,l) \\
\leq&\frac{1}{p} \sum_{i,j\in[p]} v^2(i,j) + \frac{1}{p(p-1)} \left(\sum_{i,j\in[p]} v^2(i,j) + \left(\sum_{i,j\in[p]} v(i,j) \right)^2 \right),
\end{aligned} \tag{51}
$$

where the last step is due to Lemma 16.

Therefore

$$
\sup_{\|v\|=1, \left\langle v, \mathbf{1}_{[p]\times[p]} \right\rangle = 0} \mathbb{E}_{\sigma_1,\sigma_2\sim\mathrm{Unif}(S_p)} \left\langle v, h^c_{\sigma_1,\sigma_2} \right\rangle^2 \leq \frac{1}{p-1}, \tag{52}
$$

which completes the proof of Theorem 15.

□

Lemma 16. *For any $v \in \mathbb{R}^{p\times p}$, we have that*

$$
\sum_{i\in[p]} \left(\sum_{j\in[p]} v_{i,j} \right)^2 + \sum_{j\in[p]} \left(\sum_{i\in[p]} v_{i,j} \right)^2 + \sum_{\substack{i,j,k,l\in[p]\\ i\neq k, j\neq l}} v_{i,j}v_{k,l} = \sum_{i,j\in[p]} v_{i,j}^2 + \left(\sum_{i,j\in[p]} v_{i,j} \right)^2. \tag{53}
$$

Proof of Lemma 16. Proof of this lemma is straightforward from applying the multinomial theorem. □