FOURIER CIRCUITS IN NEURAL NETWORKS: UNLOCKING THE POTENTIAL OF LARGE LANGUAGE MODELS IN MATHEMATICAL REASONING AND MODULAR ARITHMETIC

Jiuxiang Gu* **Chenyang Li†** **Yingyu Liang‡** **Zhenmei Shi§** **Zhao Song¶** **Tianyi Zhou‖**

ABSTRACT

In the evolving landscape of machine learning, a pivotal challenge lies in deciphering the internal representations harnessed by neural networks and Transformers. Building on recent progress toward comprehending how networks execute distinct target functions, our study embarks on an exploration of the underlying reasons behind networks adopting specific computational strategies. We direct our focus to the complex algebraic learning task of modular addition involving k inputs. Our research presents a thorough analytical characterization of the features learned by stylized one-hidden layer neural networks and one-layer Transformers in addressing this task. A cornerstone of our theoretical framework is the elucidation of how the principle of margin maximization shapes the features adopted by one-hidden layer neural networks. Let p denote the modulus, D_p denote the dataset of modular arithmetic with k inputs and m denote the network width. We demonstrate that a neuron count of $m \geq 2^{2k-2} \cdot (p-1)$, these networks attain a maximum $L_{2,k+1}$-margin on the dataset D_p. Furthermore, we establish that each hidden-layer neuron aligns with a specific Fourier spectrum, integral to solving modular addition problems. By correlating our findings with the empirical observations of similar studies, we contribute to a deeper comprehension of the intrinsic computational mechanisms of neural networks. Furthermore, we observe similar computational mechanisms in the attention matrix of the Transformer. This research stands as a significant stride in unraveling their operation complexities, particularly in the realm of complex algebraic tasks.

1 INTRODUCTION

The field of artificial intelligence has experienced a significant transformation with the development of large language models (LLMs), particularly through the introduction of the Transformer architecture. This advancement has revolutionized approaches to challenging tasks in natural language processing, notably in machine translation (Prato et al., 2020; Gao et al., 2020) and text generation (Luo et al., 2022). Consequently, models e.g., BERT Devlin et al. (2018), PaLM Chowdhery et al. (2022), LLaMA 2Touvron et al. (2023), ChatGPT (OpenAI, 2022), GPT4 (OpenAI, 2023) and so on, have become predominant in NLP.

Central to this study is the question of how these advanced models transcend mere pattern recognition to engage in what appears to be logical reasoning and problem-solving. This inquiry is not purely academic; it probes the core of "understanding" in artificial intelligence. While LLMs, such as GPT, demonstrate remarkable proficiency in human-like text generation, their capability in comprehending and processing mathematical logic is a topic of considerable debate.

* jigu@adobe.com. Adobe Research.
† lchenyang550@gmail.com. Fuzhou University.
‡ yingyul@hku.hk. The University of Hong Kong. yliang@cs.wisc.edu. University of Wisconsin-Madison.
§ zhmeishi@cs.wisc.edu. University of Wisconsin-Madison.
¶ zsong@adobe.com. Adobe Research.
‖ tzhou029@usc.edu. University of Southern California.

This line of investigation is crucial, given AI's potential to extend beyond text generation into deeper comprehension of complex subjects. Mathematics, often seen as the universal language, presents a uniquely challenging domain for these models Yousefzadeh & Cao (2023). Our research aims to determine if Transformers, noted for their NLP efficiency, can also demonstrate an intrinsic understanding of mathematical operations and reasoning.

In a recent surprising study of mathematical operations learning, Power et al. (2022) train Transformers on small algorithmic datasets, e.g., $a_1 + a_2 \mod p$ and we let p be a prime number, and show the "grokking" phenomenon, where models abruptly transition from bad generalization to perfect generalization after a large number of training steps. Nascent studies, such as those by Nanda et al. (2023), empirically reveal that Transformers can solve modular addition using Fourier-based circuits. They found that the Transformers trained by Stochastic Gradient Descent (SGD) not only reliably compute $a_1 + a_2 \mod p$, but also that the networks consistently employ a specific geometric algorithm. This algorithm, which involves composing integer rotations around a circle, indicates an inherent comprehension of modular arithmetic within the network's architecture. The algorithm relies on this identity: for any a_1, a_2 and $\zeta \in \mathbb{Z}_p \backslash \{0\}$, the following two quantities are equivalent

- $(a_1 + a_2) \bmod p$
- $\arg\max_{c \in \mathbb{Z}_p} \{\cos(2\pi\zeta(a_1 + a_2 - c)/p)\}$

Nanda et al. (2023) further show that the attention and MLP module in the Transformer imbues the neurons with Fourier circuit-like properties.

To study why networks arrive at Fourier-based circuits computational strategies, Morwani et al. (2024) theoretically study one-hidden layer neural network learning on two inputs modular addition task and certify that the trained networks will execute modular addition by employing Fourier features aligning closely with the previous empirical observations. However, the question remains whether neural networks can solve more complicated mathematical problems.

Inspired by recent developments in mechanistic interpretability Olah et al. (2020); Elhage et al. (2021; 2022) and the study of inductive biases Soudry et al. (2018); Vardi (2023) in neural networks, we extend our research to modular addition with three or more (k) inputs.

$$(a_1 + \cdots + a_k) \bmod p. \tag{1}$$

This approach offers insights into why certain representations and solutions emerge from neural network training. By integrating these insights with our empirical findings, we aim to provide a comprehensive understanding of neural networks' learning mechanisms, especially in solving the modular addition problem. We also determine the necessary number of neurons for the network to learn this Fourier method for modular addition.

Our paper's contributions are summarized as follows:

- **Expansion of Input for Cyclic Groups Problem:** We extend the input parameter range for the cyclic groups problem from a binary set to k-element sets.
- **Network's Maximum Margin:** Let $L_{a,b}$ be defined in Definition C.3. On the modular dataset D_p, we demonstrate that the maximum $L_{2,k+1}$-margin of a network is:

$$\gamma^* = \frac{2(k!)}{(2k+2)^{(k+1)/2}(p-1)p^{(k-1)/2}}.$$

- **Neuron Count in One-Hidden-Layer Networks:** We propose that in a general case, a one-hidden-layer network having $m \geq 2^{2k-2} \cdot (p-1)$ neurons can achieve the maximum $L_{2,k+1}$-margin solution. This ensures the network's capability to effectively solve the cyclic groups problem in a Fourier-based method, a finding corroborated by our experimental data.
- **Empirical Validation of Theoretical Findings:** We validate our theoretical finding that when $m \geq 2^{2k-2} \cdot (p-1)$, for each spectrum $\zeta \in \{1, \ldots, \frac{p-1}{2}\}$, there exists a hidden-neuron utilizes this spectrum.
- **Similar Empirical Findings in One-Layer Transformer:** We also have a similar observation in one-layer Transformer learning modular addition involving k inputs. For the

2-dimensional matrix $W_K W_Q$, where W_K, W_Q denotes the key and query matrix, it shows the superposition of two cosine waveforms in each dimension, each characterized by distinct frequencies.

- **Grokking under Different** k: We observe that as k increases, the grokking phenomenon becomes weak.

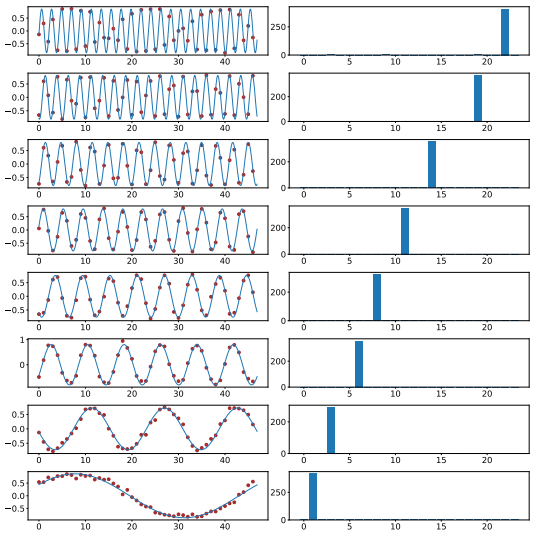


Figure 1: Cosine shape of the trained embeddings (hidden layer weights) and corresponding power of Fourier spectrum. The two-layer network with $m = 2944$ neurons is trained on $k = 4$-sum mod-$p = 47$ addition dataset. We even split the entire dataset ($p^k = 47^4$ data points) into training and test datasets. Each row represents a random neuron from the network. The left figure shows the final trained embeddings, with red dots indicating the true weight values, and the pale blue interpolation is achieved by identifying the function that shares the same Fourier spectrum. The right figure shows their Fourier power spectrum. The results in these figures are consistent with our analysis statements in Lemma F.1. See Figure 5, 7 in Appendix L.1 for similar results when k is 3 or 5.

2 Experiments

2.1 One-hidden Layer Neural Network

We conduct simulation experiments to verify our analysis. In Figure 1 and Figure 3, we use SGD to train a two-layer network with $m = 2944 = 2^{2k-2} \cdot (p-1)$ neurons, i.e., Eq. equation 3, on $k = 4$-sum mod-$p = 47$ addition dataset, i.e., Eq. equation 1. Figure 1 shows that the networks trained with SGD have single-frequency hidden neurons, which support our analysis in Lemma F.1. Furthermore, Figure 3 demonstrates that the network will learn all frequencies in the Fourier spectrum which is consistent with our analysis in Lemma F.2. Together, they verify our main results in Theorem D.1 and show that the network trained by SGD prefers to learn Fourier-based circuits. There are more similar results when k is 3 or 5 in Appendix L.1.

2.2 One-layer Transformer

We find similar results in the one-layer transformer. Recall that the m-heads attention layer can be written as

$$W^P \begin{pmatrix} W_1^{V\top} E \cdot \mathrm{softmax}\left(E^\top W_1^K W_1^{Q\top} E\right) \\ \cdots \\ W_m^{V\top} E \cdot \mathrm{softmax}\left(E^\top W_m^K W_m^{Q\top} E\right) \end{pmatrix}, \tag{2}$$

where E is input embedding and W^P, W^V, W^K, W^Q are projection, value, key and query matrix. We denote $W^K W^{Q\top}$ as W^{KQ} and call it attention matrix.

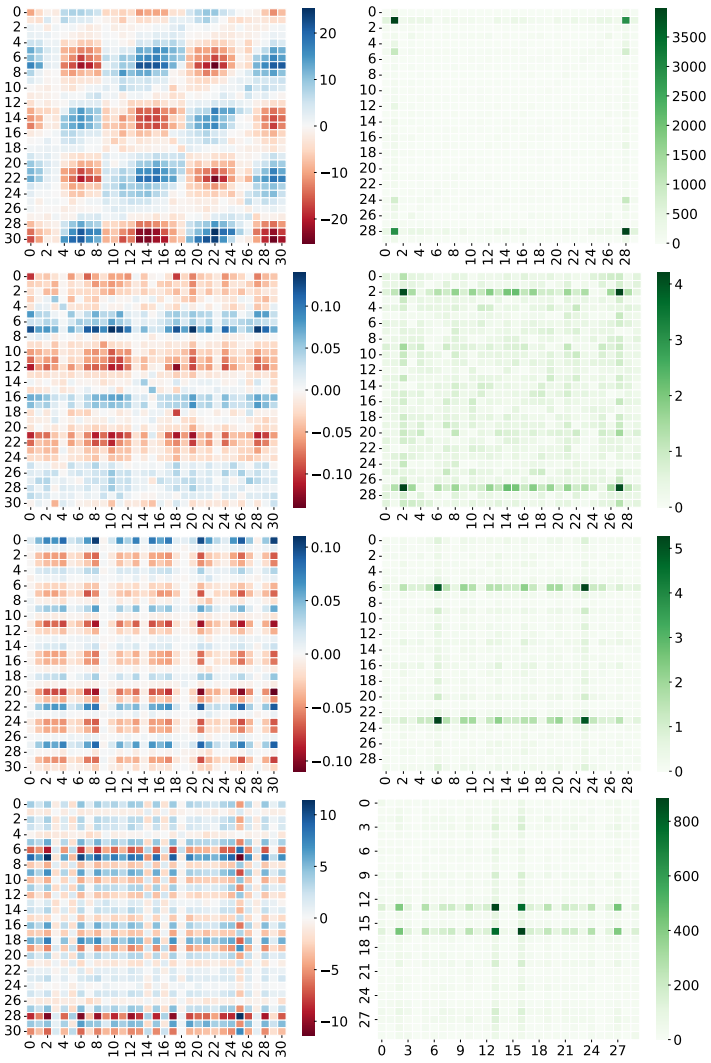
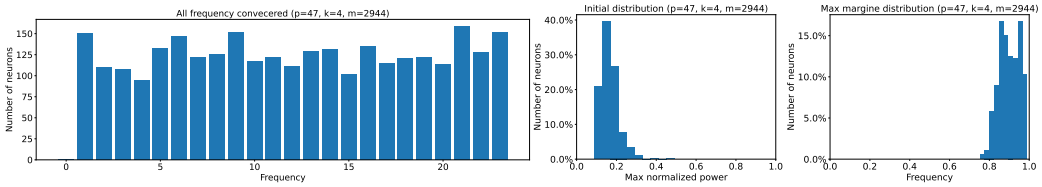


Figure 2: 2-dimension cosine shape of the trained W^{KQ} (attention weights) and their Fourier power spectrum. The one-layer transformer with attention heads $m = 160$ is trained on $k = 4$-sum mod-$p = 31$ addition dataset. We even split the whole datasets ($p^k = 31^4$ data points) into training and test datasets. Every row represents a random attention head from the transformer. The left figure shows the final trained attention weights being an apparent 2-dim cosine shape. The right figure shows their 2-dim Fourier power spectrum. The results in the figures are consistent with Figure 1. See Figure 9 and Figure 10 in Appendix L.2 for similar results when k is 3 or 5.

In Figure 2 , we train a one-layer transformer with $m = 160$ heads attention, i.e., Eq. equation 2, on $k = 4$-sum mod-$p = 31$ addition dataset, i.e., Eq. equation 1. Figure 2 shows that the SGD trained one-layer transformer learns 2-dim cosine shape attention matrices, which is similar to the one-hidden layer neural networks in Figure 1. This means that the attention layer has a learning mechanism similar to neural networks in the modular arithmetic task. It prefers to learn (2-dim) Fourier-based circuits when trained by SGD. There are more similar results when k is 3 or 5 in Appendix L.2.

2.3 Grokking under Different k

Following the experiments' protocol in Power et al. (2022), we show there is the grokking phenomenon under different k. We train two-layer transformers with $m = 160$ attention heads on $k = 2, 3, 4, 5$-sum mod-$p = 97, 31, 11, 5$ addition dataset with 50% of the data in the training. We use different p to guarantee the dataset sizes are roughly equal to each other. As k increases, the



(a) (b) (c)

Figure 3: All Fourier spectrum frequencies being covered and the maximum normalized power of the embeddings (hidden layer weights). The one-hidden layer network with $m = 2944$ neurons is trained on $k = 4$-sum mod-$p = 47$ addition dataset. We denote $\widehat{u}[i]$ as the Fourier transform of $u[i]$. Let $\max_i |\widehat{u}[i]|^2/(\sum |\widehat{u}[j]|^2)$ be the maximum normalized power. Mapping each neuron to its maximum normalized power frequency, (a) shows the final frequency distribution of the embeddings. Similar to our construction analysis in Lemma F.2, we have an almost uniform distribution over all frequencies. (b) shows the maximum normalized power of the neural network with random initialization. (c) shows, in frequency space, the embeddings of the final trained network are one-sparse, i.e., maximum normalized power being almost 1 for all neurons. This is consistent with our maximum-margin analysis results in Lemma F.2. See Figure 6 and Figure 8 in Appendix L.1 for similar results when k is 3 or 5.

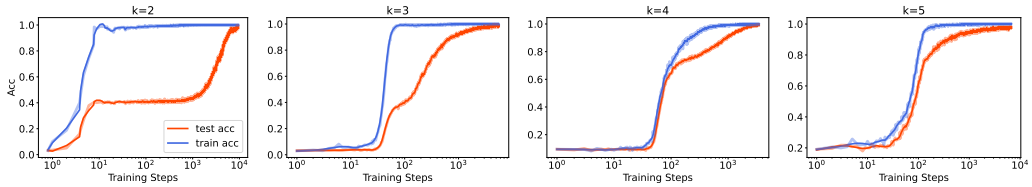


Figure 4: Grokking (models abruptly transition from bad generalization to perfect generalization after a large number of training steps) under learning modular addition involving $k = 2, 3, 4, 5$ inputs. We train two-layer transformers with $m = 160$ attention heads on $k = 2, 3, 4, 5$-sum mod-$p = 97, 31, 11, 5$ addition dataset with 50% of the data in the training set under AdamW Loshchilov & Hutter (2018) optimizer 1e-3 learning rate and 1e-4 weight decay. We use different p to guarantee the dataset sizes are roughly equal to each other. The blue curves show training accuracy and the red ones show validation accuracy. There is a grokking phenomenon in all figures. However, as k increases, the grokking phenomenon becomes weak. It implies that when the ground-truth function class becomes "complicated", the transformers need to train more steps to overfit the training datasets and the generalization tends to be better.

grokking phenomenon becomes weak. It implies that when the ground-truth function class becomes "complicated", the transformers need to train more steps to fit the training datasets and the generalization tends to be better. This is consistent with our analysis. In Theorem D.1, we show that we need $2^{2k-2} \cdot (p-1)$ neurons to get maximum-margin solution, and we can check this value is $2^2 \cdot (97-1) = 384, 2^4 \cdot (31-1) = 480, 2^6 \cdot (11-1) = 640, 2^8 \cdot (5-1) = 1024$ for $k = 2, 3, 4, 5$. It means that, when k becomes larger, although the whole input space size is roughly unchanged, we need more neurons to fit the optimal solution as the function class becomes more "complicated". Thus, with increasing k, if we train the same size of models under these settings, the models will slightly transfer from overfitting to underfitting, so the grokking phenomenon becomes weak.

3 Conclusion

We study neural networks and transformers learning on $(a_1 + \cdots + a_k) \bmod p$. We theoretically show that networks prefer to learn Fourier circuits. Our experiments on neural networks and transformers support our analysis. Finally, we study the grokking phenomenon under this new data setting.

REFERENCES

Josh Alman and Zhao Song. Fast attention requires bounded entries. In *NeurIPS*. arXiv preprint arXiv:2302.13214, 2023.

Josh Alman and Zhao Song. How to capture higher-order correlations? generalizing matrix softmax attention to kronecker computation. In *The Twelfth International Conference on Learning Representations*, 2024.

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.

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.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, March 2022. doi: 10.1162/coli_a_00422.

Avrim Blum, Merrick Furst, Jeffrey Jackson, Michael Kearns, Yishay Mansour, and Steven Rudich. Weakly learning dnf and characterizing statistical query learning using fourier analysis. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pp. 253–262, 1994.

Jean Bourgain. An improved estimate in the restricted isometry problem. In *Geometric aspects of functional analysis*, pp. 65–70. Springer, 2014.

Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Ido Bronstein, Alon Brutzkus, and Amir Globerson. On the inductive bias of neural networks for learning read-once dnfs. In *Uncertainty in Artificial Intelligence*, pp. 255–265. PMLR, 2022.

Nick Cammarata, Gabriel Goh, Shan Carter, Ludwig Schubert, Michael Petrov, and Chris Olah. Curve detectors. *Distill*, 5(6):e00024–003, 2020.

Emmanuel J Candes and Terence Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE transactions on information theory*, 52(12):5406–5425, 2006.

Yuan Cao, Zixiang Chen, Misha Belkin, and Quanquan Gu. Benign overfitting in two-layer convolutional neural networks. *Advances in neural information processing systems*, 35:25237–25250, 2022.

Sitan Chen, Jerry Li, and Zhao Song. Learning mixtures of linear regressions in subexponential time via fourier moments. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 587–600, 2020.

Xiang Chen, Zhao Song, Baocheng Sun, Junze Yin, and Danyang Zhuo. Query complexity of active learning for function family with nearly orthogonal basis. *arXiv preprint arXiv:2306.03356*, 2023.

Xue Chen, Daniel M Kane, Eric Price, and Zhao Song. Fourier-sparse interpolation without a frequency gap. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 741–750. IEEE, 2016.

Lenaic Chizat and Francis Bach. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. In *Conference on Learning Theory*. PMLR, 2020.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

Bilal Chughtai, Lawrence Chan, and Neel Nanda. A toy model of universality: Reverse engineering how networks learn group operations. *arXiv preprint arXiv:2302.03025*, 2023.

Alexandru Damian, Jason Lee, and Mahdi Soltanolkotabi. Neural networks can learn representations with gradient descent. In *Conference on Learning Theory*. PMLR, 2022.

Amit Daniely and Eran Malach. Learning parities with neural networks. *Advances in Neural Information Processing Systems*, 33:20356–20365, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1, 2021.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022.

Spencer Frei, Niladri S Chatterji, and Peter Bartlett. Benign overfitting without linearity: Neural network classifiers trained by gradient descent for noisy linear data. In *Conference on Learning Theory*, pp. 2668–2703. PMLR, 2022a.

Spencer Frei, Gal Vardi, Peter Bartlett, Nathan Srebro, and Wei Hu. Implicit bias in leaky relu networks trained on high-dimensional data. In *The Eleventh International Conference on Learning Representations*, 2022b.

Spencer Frei, Gal Vardi, Peter Bartlett, and Nathan Srebro. Benign overfitting in linear classifiers and leaky relu networks from kkt conditions for margin maximization. In *The Thirty Sixth Annual Conference on Learning Theory*, pp. 3173–3228. PMLR, 2023.

Peng Gao, Chiori Hori, Shijie Geng, Takaaki Hori, and Jonathan Le Roux. Multi-pass transformer for machine translation. *arXiv preprint arXiv:2009.11382*, 2020.

Yeqi Gao, Zhao Song, and Baocheng Sun. An $O(k \log n)$ time fourier set query algorithm. *arXiv preprint arXiv:2208.09634*, 2022.

Anna C Gilbert, Shan Muthukrishnan, and Martin Strauss. Improved time bounds for near-optimal sparse fourier representations. In *Wavelets XI*, volume 5914, pp. 59141A. International Society for Optics and Photonics, 2005.

Jiuxiang Gu, Chenyang Li, Yingyu Liang, Zhenmei Shi, and Zhao Song. Exploring the frontiers of softmax: Provable optimization, applications in diffusion model, and beyond. *manuscript*, 2024a.

Jiuxiang Gu, Yingyu Liang, Heshan Liu, Zhenmei Shi, Zhao Song, and Junze Yin. Conv-basis: A new paradigm for efficient attention inference and gradient computation in transformers. *manuscript*, 2024b.

Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Characterizing implicit bias in terms of optimization geometry. In *International Conference on Machine Learning*, pp. 1832–1841. PMLR, 2018a.

Suriya Gunasekar, Jason D Lee, Daniel Soudry, and Nati Srebro. Implicit bias of gradient descent on linear convolutional networks. *Advances in Neural Information Processing Systems*, 31, 2018b.

Haitham Hassanieh, Piotr Indyk, Dina Katabi, and Eric Price. Simple and practical algorithm for sparse fourier transform. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms (SODA)*, pp. 1183–1194. SIAM, 2012a.

Haitham Hassanieh, Piotr Indyk, Dina Katabi, and Eric Price. Nearly optimal sparse fourier transform. In *Proceedings of the forty-fourth annual ACM symposium on Theory of Computing (STOC)*, pp. 563–578, 2012b.

Ishay Haviv and Oded Regev. The restricted isometry property of subsampled fourier matrices. In *Geometric aspects of functional analysis*, pp. 163–179. Springer, 2017.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Piotr Indyk and Michael Kapralov. Sample-optimal Fourier sampling in any constant dimension. In *IEEE 55th Annual Symposium onFoundations of Computer Science (FOCS)*, pp. 514–523. IEEE, 2014.

Piotr Indyk, Michael Kapralov, and Eric Price. (nearly) sample-optimal sparse fourier transform. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms (SODA)*, pp. 480–499. SIAM, 2014.

Arthur Jacot. Implicit bias of large depth networks: a notion of rank for nonlinear functions. In *The Eleventh International Conference on Learning Representations*, 2022.

Samy Jelassi, Michael Sander, and Yuanzhi Li. Vision transformers provably learn spatial structure. *Advances in Neural Information Processing Systems*, 2022.

Ziwei Ji and Matus Telgarsky. The implicit bias of gradient descent on nonseparable data. In *Conference on Learning Theory*, pp. 1772–1798. PMLR, 2019.

Ziwei Ji and Matus Telgarsky. Directional convergence and alignment in deep learning. *Advances in Neural Information Processing Systems*, 33:17176–17186, 2020.

Yaonan Jin, Daogao Liu, and Zhao Song. Super-resolution and robust sparse continuous fourier transform in any constant dimension: Nearly linear time and sample complexity. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2023.

Michael Kapralov. Sparse Fourier transform in any constant dimension with nearly-optimal sample complexity in sublinear time. In *Symposium on Theory of Computing Conference (STOC)*, 2016.

Michael Kapralov. Sample efficient estimation and recovery in sparse FFT via isolation on average. In *58th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2017.

Yin Tat Lee, Zhao Song, and Qiuyi Zhang. Solving empirical risk minimization in the current matrix multiplication time. In *COLT*, 2019.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857, 2022.

Hongkang Li, Meng Wang, Sijia Liu, and Pin-Yu Chen. A theoretical understanding of shallow vision transformers: Learning, generalization, and sample complexity. In *The Eleventh International Conference on Learning Representations*, 2023a.

Yuchen Li, Yuanzhi Li, and Andrej Risteski. How do transformers learn topic structure: Towards a mechanistic understanding. In *Proceedings of the 40th International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2023b.

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, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. Biogpt: generative pre-trained transformer for biomedical text generation and mining. *Briefings in Bioinformatics*, 23(6), 2022.

Kaifeng Lyu and Jian Li. Gradient descent maximizes the margin of homogeneous neural networks. In *International Conference on Learning Representations*, 2019.

Kaifeng Lyu, Zhiyuan Li, Runzhe Wang, and Sanjeev Arora. Gradient descent on two-layer nets: Margin maximization and simplicity bias. *Advances in Neural Information Processing Systems*, 34:12978–12991, 2021.

Kaifeng Lyu, Jikai Jin, Zhiyuan Li, Simon S Du, Jason D Lee, and Wei Hu. Dichotomy of early and late phase implicit biases can provably induce grokking. In *The Twelfth International Conference on Learning Representations*, 2024.

Beren Millidge. Grokking'grokking', 2022.

Ankur Moitra. Super-resolution, extremal functions and the condition number of vandermonde matrices. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pp. 821–830, 2015.

Edward Moroshko, Blake E Woodworth, Suriya Gunasekar, Jason D Lee, Nati Srebro, and Daniel Soudry. Implicit bias in deep linear classification: Initialization scale vs training accuracy. *Advances in Neural Information Processing Systems*, 33, 2020.

Depen Morwani, Benjamin L Edelman, Costin-Andrei Oncescu, Rosie Zhao, and Sham Kakade. Feature emergence via margin maximization: case studies in algebraic tasks. In *The Twelfth International Conference on Learning Representations*, 2024.

Shikhar Murty, Pratyusha Sharma, Jacob Andreas, and Christopher D Manning. Grokking of hierarchical structure in vanilla transformers. *arXiv preprint arXiv:2305.18741*, 2023.

Vasileios Nakos, Zhao Song, and Zhengyu Wang. (nearly) sample-optimal sparse fourier transform in any dimension; ripless and filterless. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 1568–1577. IEEE, 2019.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. In *The Eleventh International Conference on Learning Representations*, 2023.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 5(3):e00024–001, 2020.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

OpenAI. Introducing ChatGPT. https://openai.com/blog/chatgpt, 2022. Accessed: 2023-09-10.

OpenAI. Gpt-4 technical report, 2023.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*, 2022.

Gabriele Prato, Ella Charlaix, and Mehdi Rezagholizadeh. Fully quantized transformer for machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 1–14, 2020.

Eric Price and Zhao Song. A robust sparse Fourier transform in the continuous setting. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pp. 583–600. IEEE, 2015.

Mark Rudelson and Roman Vershynin. On sparse reconstruction from fourier and gaussian measurements. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 61(8):1025–1045, 2008.

Clayton Sanford, Daniel Hsu, and Matus Telgarsky. Representational strengths and limitations of transformers. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. Analysing mathematical reasoning abilities of neural models. In *International Conference on Learning Representations*, 2018.

Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. *Advances in Neural Information Processing Systems*, 33:9573–9585, 2020.

Shai Shalev-Shwartz, Ohad Shamir, and Shaked Shammah. Failures of gradient-based deep learning. In *International Conference on Machine Learning*, pp. 3067–3075. PMLR, 2017.

Zhenmei Shi, Junyi Wei, and Yingyu Liang. A theoretical analysis on feature learning in neural networks: Emergence from inputs and advantage over fixed features. In *International Conference on Learning Representations*, 2022.

Zhenmei Shi, Jiefeng Chen, Kunyang Li, Jayaram Raghuram, Xi Wu, Yingyu Liang, and Somesh Jha. The trade-off between universality and label efficiency of representations from contrastive learning. In *The Eleventh International Conference on Learning Representations*, 2023a. URL https://openreview.net/forum?id=rvsbw2YthH_.

Zhenmei Shi, Yifei Ming, Ying Fan, Frederic Sala, and Yingyu Liang. Domain generalization via nuclear norm regularization. In *Conference on Parsimony and Learning (Proceedings Track)*, 2023b. URL https://openreview.net/forum?id=hJd66ZzXEZ.

Zhenmei Shi, Junyi Wei, and Yingyu Liang. Provable guarantees for neural networks via gradient feature learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023c.

Zhenmei Shi, Junyi Wei, Zhuoyan Xu, and Yingyu Liang. Why larger language models do in-context learning differently? In *R0-FoMo:Robustness of Few-shot and Zero-shot Learning in Large Foundation Models*, 2023d. URL https://openreview.net/forum?id=2J8xnFLMgF.

Zhao Song. *Matrix Theory: Optimization, Concentration and Algorithms*. PhD thesis, The University of Texas at Austin, 2019.

Zhao Song, Baocheng Sun, Omri Weinstein, and Ruizhe Zhang. Sparse fourier transform over lattices: A unified approach to signal reconstruction. *arXiv preprint arXiv:2205.00658*, 2022.

Zhao Song, Baocheng Sun, Omri Weinstein, and Ruizhe Zhang. Quartic samples suffice for fourier interpolation. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 1414–1425. IEEE, 2023a.

Zhao Song, Mingquan Ye, Junze Yin, and Lichen Zhang. A nearly-optimal bound for fast regression with ℓ_∞, guarantee. In *ICML*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32463–32482. PMLR, 2023b.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1): 2822–2878, 2018.

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.

Yuandong Tian, Yiping Wang, Beidi Chen, and Simon Du. Scan and snap: Understanding training dynamics and token composition in 1-layer transformer. *Advances in Neural Information Processing Systems*, 2023.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Alexander Tsigler and Peter L Bartlett. Benign overfitting in ridge regression. *Journal of Machine Learning Research*, 24(123):1–76, 2023.

Gal Vardi. On the implicit bias in deep-learning algorithms. *Communications of the ACM*, 66(6): 86–93, 2023.

Colin Wei, Jason D Lee, Qiang Liu, and Tengyu Ma. Regularization matters: Generalization and optimization of neural nets vs their induced kernel. *Advances in Neural Information Processing Systems*, 32, 2019a.

Colin Wei, Jason D Lee, Qiang Liu, and Tengyu Ma. Regularization matters: Generalization and optimization of neural nets vs their induced kernel. *Advances in Neural Information Processing Systems*, 32, 2019b.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022.

Zhiwei Xu, Yutong Wang, Spencer Frei, Gal Vardi, and Wei Hu. Benign overfitting and grokking in relu networks for xor cluster data. In *The Twelfth International Conference on Learning Representations*, 2024a.

Zhuoyan Xu, Zhenmei Shi, Junyi Wei, Yin Li, and Yingyu Liang. Improving foundation models for few-shot learning via multitask finetuning. In *ICLR 2023 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2023. URL https://openreview.net/forum?id=szNb8Hp3d3.

Zhuoyan Xu, Zhenmei Shi, and Yingyu Liang. Do large language models have compositional ability? an investigation into limitations and scalability. In *ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2024b. URL https://openreview.net/forum?id=4XPeF0SbJs.

Zhuoyan Xu, Zhenmei Shi, Junyi Wei, Fangzhou Mu, Yin Li, and Yingyu Liang. Towards few-shot adaptation of foundation models via multitask finetuning. In *The Twelfth International Conference on Learning Representations*, 2024c.

Roozbeh Yousefzadeh and Xuenan Cao. Large language models' understanding of math: Source criticism and extrapolation. *arXiv preprint arXiv:2311.07618*, 2023.

Appendix

CONTENTS

A Related Work

Max Margin Solutions in Neural Networks. Bronstein et al. (2022) demonstrated that neurons in a one-hidden-layer ReLU network align with clauses in max margin solutions for read-once DNFs, employing a unique proof technique involving the construction of perturbed networks. Morwani et al. (2024) utilize max-min duality to certify maximum-margin solutions. Further, extensive research in the domain of margin maximization in neural networks, including works by Gunasekar et al. (2018b); Soudry et al. (2018); Gunasekar et al. (2018a); Wei et al. (2019b); Lyu & Li (2019); Ji & Telgarsky (2019); Moroshko et al. (2020); Chizat & Bach (2020); Ji & Telgarsky (2020); Lyu et al. (2021); Frei et al. (2022b; 2023); Shi et al. (2023b); Gu et al. (2024a) and more, has highlighted the implicit bias towards margin maximization inherent in neural network optimization. They provide a foundational understanding of the dynamics of neural networks and their inclination towards maximizing margins under various conditions and architectures.

Algebraic Tasks Learning Mechanism Interpretability. The study of neural networks trained on algebraic tasks has been pivotal in shedding light on their training dynamics and inductive biases. Notable contributions include the work of Power et al. (2022); Chughtai et al. (2023) on modular addition and subsequent follow-up studies, investigations into learning parities Daniely & Malach (2020); Barak et al. (2022); Shi et al. (2022; 2023d;c;a); Xu et al. (2024b), and research into algorithmic reasoning capabilities Saxton et al. (2018); Hendrycks et al. (2021); Lewkowycz et al. (2022); Damian et al. (2022). The field of mechanistic interpretability, focusing on the analysis of internal representations in neural networks, has also seen significant advancements through the works of Cammarata et al. (2020); Olsson et al. (2022) and others.

Grokking and Emergent Ability. The phenomenon known as "grokking" was initially identified by Power et al. (2022) and is believed to be a way of studying the emerging abilities of LLM Wei et al. (2022). This research observed a unique trend in two-layer transformer models engaged in algorithmic tasks, where there was a significant increase in test accuracy, surprisingly occurring well after these models had reached perfect accuracy in their training phase. In Millidge (2022), it was hypothesized that this might be the result of the SGD process that resembles a random path along what is termed the optimal manifold. Adding to this, Nanda et al. (2023) aligns with the findings of Belinkov (2022), indicating a steady advancement of networks towards algorithms that are better at generalization. Liu et al. (2022); Xu et al. (2024a); Lyu et al. (2024) developed smaller-scale examples of grokking and utilized these to map out phase diagrams, delineating multiple distinct learning stages. Furthermore, Thilak et al. (2022); Murty et al. (2023) suggested the possibility of grokking occurring naturally, even in the absence of explicit regularization. They attributed this to an optimization quirk they termed the slingshot mechanism, which might inadvertently act as a regularizing factor.

Theoretical Work About Fourier Transform. To calculate Fourier transform there are two main methodologies: one uses carefully chosen samples through hashing functions (referenced in works like Gilbert et al. (2005); Hassanieh et al. (2012b;a); Indyk et al. (2014); Indyk & Kapralov (2014); Kapralov (2016; 2017)) to achieve sublinear sample complexity and running time, while the other uses random samples (as discussed in Candes & Tao (2006); Rudelson & Vershynin (2008); Bourgain (2014); Haviv & Regev (2017); Nakos et al. (2019)) with sublinear sample complexity but nearly linear running time. There are many other works studying Fourier transform Price & Song (2015); Moitra (2015); Song (2019); Jin et al. (2023); Gao et al. (2022); Lee et al. (2019); Chen et al. (2020); Song et al. (2022); Chen et al. (2016); Song et al. (2023a); Chen et al. (2023); Song et al. (2023b); Gu et al. (2024b).

B Discussion

Connection to Parity and SQ Hardness. If we let $p = 2$, then $(a_1 + \cdots + a_k) \bmod p$ will degenerate to parity function, i.e., $b_1, \ldots, b_k \in \{\pm 1\}$ and determining $\prod_{i=1}^k b_i$. Parity functions serve as a fundamental set of learning challenges in computational learning theory, often used to demonstrate computational obstacles Shalev-Shwartz et al. (2017). In particular, (n, k)-sparse parity problem is notorious hard to learn, i.e., Statistical Query (SQ) hardness Blum et al. (1994). Daniely

& Malach (2020) showed that one-hidden layer networks need an $\Omega(\exp(k))$ number of neurons or an $\Omega(\exp(k))$ number of training steps to successfully learn it by SGD. In our work, we are studying Eq. equation 3, which is a more general function than parity and indeed is a learning hardness. Our Theorem D.1 states that we need $\Omega(\exp(k))$ number of neurons to represent the maximum-margin solution, which well aligns with existing works. Our experiential results in Section 2.1 are also consistent. Hence, our modular addition involving k inputs function class is a good data model to analyze and test the model learning ability, i.e., approximation power, optimization, and generalization.

High Order Correlation Attention. Sanford et al. (2023); Alman & Song (2023; 2024) state that, when $k = 3$, $a_1 + a_2 + a_3 \bmod p$ is hard to be captured by traditional attention. Thus, they introduce high-order attention to capture high-order correlation from the input sequence. However, in Section 2.2, we show that one-layer transformers have a strong learning ability and can successfully learn modular arithmetic tasks even when $k = 5$. This implies that the traditional attention may be more powerful than we expect. We conjecture that the layer norm and residual connection contribute as they are ignored by most transformer learning theoretical analysis work Jelassi et al. (2022); Li et al. (2023a;b); Tian et al. (2023).

Grokking, Benign Overfitting, and Implicit Bias. Recently, Xu et al. (2024a) connects the grokking phenomenon to benign overfitting Bartlett et al. (2020); Cao et al. (2022); Tsigler & Bartlett (2023); Frei et al. (2022a; 2023). It shows how the network undergoes a grokking period from catastrophic to benign overfitting. Lyu et al. (2024) uses implicit bias Soudry et al. (2018); Gunasekar et al. (2018a); Ji & Telgarsky (2019); Shah et al. (2020); Moroshko et al. (2020); Chizat & Bach (2020); Lyu et al. (2021); Jacot (2022); Xu et al. (2023; 2024c) to explain grokking, where grokking happens if the early phase bias implies an overfitting solution while late phase bias implies a generalizable solution. The intuition from the benign overfitting and the implicit bias well align with our observation in Section 2.3. It is interesting and valuable to rigorously analyze the grokking or emergent ability under different function class complexities, e.g., Eq equation 1. We leave this challenge problem as a future work.

C Problem Setup

Part of our notations are following Morwani et al. (2024). The setting of our one-hidden-layer neural network is in Section C.1. In Section C.2, we define the margin of the neuron network. We introduce a lemma that connects the training neuron network to solving the maximum-margin problem in Section C.3.

C.1 Data and Network Setup

We denote $\mathbb{Z}_p$ as the modular group on p integers, e.g., $\mathbb{Z}_p = [p]$, where $p > 2$ is a given prime number. We denote $\mathcal{X} := \mathbb{Z}_p^k$ as the input space and $\mathcal{Y} := \mathbb{Z}_p$ as the output space. We denote $D_p := \{((a_1, \ldots, a_k), \sum_{i\in[k]} a_i) : a_1, \ldots, a_k \in \mathbb{Z}_p\}$ as the modular dataset.

Suppose we have a $L_{2,k+1}$ (matrix) norm $\|\cdot\|_{2,k+1}$ (Definition C.3) and a class of parameterized functions $\{f(\theta, \cdot) \mid \theta \in \mathbb{R}^U\}$, where $f : \mathbb{R}^U \times \mathcal{X} \to \mathbb{R}^{\mathcal{Y}}$ and a matrix set $\Theta := \{\|\theta\|_{2,k+1} \leq 1\}$. We consider single-hidden layer neural networks with polynomial activation functions and without biases. With parameters $\theta \in \Theta$, we denote $f(\theta, x)$ as the network's output for an input x. For one-hidden layer networks, we denote f as:

$$f(\theta, x) := \sum_{i=1}^{m} \phi(\theta_i, x),$$

where $\theta := \{\theta_1, \ldots, \theta_m\}$. Here, we denote ϕ as one neuron and $\theta_i \in \Omega$ are the corresponding weights in that neuron, where Ω is a vector set. Let Ω' be a subset of Ω. For every $i \in [m]$, when either there exists $\alpha_i > 0$ such that $\alpha_i\theta_i \in \Omega'$ or $\theta_i = 0$, then we say the parameter set $\theta = \{\theta_1, \ldots, \theta_m\}$ has directional support on Ω'. A single neuron is represented as:

$$\phi(\{u_1, \ldots, u_k, w\}, x_1, \ldots, x_k) := (u_1^\top x_1 + \cdots + u_k^\top x_k)^k w,$$

where $u_1, \ldots, u_k, w \in \mathbb{R}^p$ are the neuron's weights, and $x_1, \ldots, x_k \in \mathbb{R}^p$ are the network inputs. Inputs $x_1, \ldots, x_k$ are one-hot vectors representing group elements. For input elements $(a_1, \ldots, a_k)$, a neuron simplifies to

$$\phi(\{u_1, \ldots, u_k, w\}, a_1, \ldots, a_k) = (u_1(a_1) + \cdots + u_k(a_k))^k w,$$

with $u_1(a_1)$ being the a_1-th component of u_1. The output of the neuron is in p-dimension for cross-entropy loss. With $\theta = \{u_{i,1}, \ldots, u_{i,k}, w_i\}_{i=1}^m$, the network is denoted as:

$$f(\theta, a_1, \ldots, a_k) := \sum_{i=1}^{m} \phi(\{u_{i,1}, \ldots, u_{i,k}, w_i\}, a_1, \ldots .a_k). \tag{3}$$

We define our regularized training objective function.

Definition C.1. *Let l be the cross-entropy loss. Our regularized training objective function is*

$$\mathcal{L}_\lambda(\theta) := \frac{1}{|D_p|} \sum_{(x,y)\in D_p} l(f(\theta, x), y) + \lambda \|\theta\|_{2,k+1}.$$

C.2 Definition and Notation

We define the vector norm and matrix norm as the following.

Definition C.2 (L_b (vector) norm)**.** *Given a vector $v \in \mathbb{R}^n$ and $b \geq 1$, we have $\|v\|_b := (\sum_{i=1}^n |v_i|^b)^{1/b}$.*

Definition C.3 ($L_{a,b}$ (matrix) norm)**.** *The $L_{a,b}$ norm of a network with parameters $\theta = \{\theta_i\}_{i=1}^m$ is $\|\theta\|_{a,b} := (\sum_{i=1}^m \|\theta_i\|_a^b)^{1/b}$, where θ_i denotes the vector of concatenated parameters for a single neuron.*

Definition C.4. *We denote $g : \mathbb{R}^U \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ as the margin function, where for given $(x, y) \in D_p$,*

$$g(\theta, x, y) := f(\theta, x)[y] - \max_{y' \in \mathcal{Y}\setminus\{y\}} f(\theta, x)[y'].$$

Definition C.5. *The margin for a given dataset D_p is denoted as $h : \mathbb{R}^U \to \mathbb{R}$ where*

$$h(\theta) := \min_{(x,y)\in D_p} g(\theta, x, y).$$

For parameter θ, its normalized margin is denoted as $h(\theta/\|\theta\|_{2,k+1})$. For simplicity, we define γ^* to be the maximum normalized margin as the following:

Definition C.6. *The minimum of the regularized objective is denoted as $\theta_\lambda \in \arg\min_{\theta\in\mathbb{R}^U} \mathcal{L}_\lambda(\theta)$. We define the normalized margin of θ_λ as $\gamma_\lambda := h(\theta_\lambda/\|\theta_\lambda\|_{2,k+1})$. We define the maximum normalized margin as $\gamma^* := \max_{\theta\in\Theta} h(\theta)$, where $\Theta := \{\|\theta\|_{2,k+1} \leq 1\}$.*

Let $\mathcal{P}(D_p)$ denote as a set containing any distributions over the dataset D_p. We see that γ^* can be rewritten as

$$\begin{aligned}
\gamma^* &= \max_{\theta\in\Theta} h(\theta) \\
&= \max_{\theta\in\Theta} \min_{(x,y)\in D_p} g(\theta, x, y) \\
&= \max_{\theta\in\Theta} \min_{q\in\mathcal{P}(D_p)} \mathop{\mathbb{E}}_{(x,y)\sim q} [g(\theta, x, y)],
\end{aligned} \tag{4}$$

where the first step follows from Definition C.6, the second step follows from Definition C.5 and the last step follows from the linearity of the expectation.

Definition C.7. *We define a pair (θ^*, q^*) when satisfying*

$$q^* \in \arg\min_{q\in\mathcal{P}(D_p)} \mathop{\mathbb{E}}_{(x,y)\sim q} [g(\theta^*, x, y)] \tag{5}$$

$$\theta^* \in \arg\min_{\theta\in\Theta} \mathop{\mathbb{E}}_{(x,y)\sim q^*} [g(\theta, x, y)]. \tag{6}$$

This means that q^* is among the entities that minimize the expected margin based on θ^*, while θ^* is among the entities that maximize the expected margin relative to q^*. The max-min inequality, as referenced in Boyd & Vandenberghe (2004), indicates that presenting such a duo adequately proves θ^* to be a maximum margin solution.

Recall that there is a "max" operation in Definition C.4, which makes the swapping of expectation and summation infeasible, which means that the expected network margin cannot be broken down into the expected margins of individual neurons. To tackle this problem, the class-weighted margin is proposed. Let $\tau : D_p \to \Delta(\mathcal{Y})$ allocate weights to incorrect labels for every data point. Given (x, y) in D_p and for any $y' \in \mathcal{Y}$, we have $\tau(x, y)[y'] \geq 0$ and $\sum_{y' \in \mathcal{Y} \setminus \{y\}} \tau(x, y)[y'] = 1$. Then, we denote g' as the following to solve the issue.

Definition C.8. *Draw $(x, y) \in D_p$. The class-weighted margin g' is defined as*

$$g'(\theta, x, y) := f(\theta, x)[y] - \sum_{y' \in \mathcal{Y} \setminus \{y\}} \tau(x, y)[y'] f(\theta, x)[y'].$$

We have g' uses a weighted sum rather than max, so $g(\theta, x, y) \leq g'(\theta, x, y)$. Following linearity of the expectation, we can get the expected class-weighted margin as

$$\mathop{\mathbb{E}}_{(x,y)}[g'(\theta, x, y)] = \sum_{i=1}^{m} \mathop{\mathbb{E}}_{(x,y)}[\phi(\theta_i, x)[y] - \sum_{y' \in \mathcal{Y} \setminus \{y\}} \tau(x, y)[y'] \phi(\theta_i, x)[y']].$$

C.3 Preliminary

We denote ν as the network's homogeneity constant, where the equation $f(\alpha\theta, x) = \alpha^\nu f(\theta, x)$ holds for any x and any scalar $\alpha > 0$. Specifically, we focus on networks with homogeneous neurons that satisfy $\phi(\alpha\theta_i, x) = \alpha^\nu \phi(\theta_i, x)$ for any $\alpha > 0$. Note that our one-hidden layer networks (Eq. equation 3) are $k + 1$ homogeneous. As the following Lemma states, when λ is small enough during training homogeneous functions, we have the $\mathcal{L}_\lambda$ global optimizers' normalized margin converges to γ^*.

Lemma C.9 (Wei et al. (2019a), Theorem 4.1)**.** *Let f be a homogeneous function. For any norm $\|\cdot\|$, if $\gamma^* > 0$, we have $\lim_{\lambda \to 0} \gamma_\lambda = \gamma^*$.*

Therefore, we can replace comprehending the global minimizes by exploring the maximum-margin solution as a surrogate, enabling us to bypass complex analyses in non-convex optimization.

Furthermore, Morwani et al. (2024) states that under the following condition, the maximum-margin solutions and class-weighted maximum-margin (g') solutions are equivalent with each other.

Condition C.10 (Condition C.1 in page 8 in Morwani et al. (2024))**.** *We have $g'(\theta^*, x, y) = g(\theta^*, x, y)$ for all $(x, y) \in \mathrm{spt}(q^*)$. It means:*

$$\{y' \in \mathcal{Y} \setminus \{y\} : \tau(x, y)[y'] > 0\} \subseteq \mathop{\arg\max}_{y' \in \mathcal{Y} \setminus \{y\}} f(\theta^*, x)[y'].$$

Hence, by satisfying these conditions, we will concentrate on describing the class-weighted maximum-margin solutions.

D Main Result

We characterize the Fourier features to perform modular addition with k input in the one-hidden-layer neuron network. We show that every neuron only focus on a distinct Fourier frequency. Additionally, within the network, there is at least one neuron of each frequency. When we consider the uniform class weighting so that $\mathcal{L}_\lambda(\theta)$ is based on τ,

$$\tau(a_1, \ldots, a_k)[c'] := 1/(p-1) \;\; \forall c' \neq a_1 + \cdots + a_k, \tag{7}$$

we have the following main result:

Theorem D.1 (Informal version of Theorem K.2)**.** *Let $f(\theta, x)$ be the one-hidden layer networks defined in Section C. If $m \geq 2^{2k-1} \cdot \frac{p-1}{2}$, then the max $L_{2,k+1}$-margin network satisfies:*

- *The maximum $L_{2,k+1}$-margin for a given dataset D_p is:*
$$\gamma^* = \frac{2(k!)}{(2k+2)^{(k+1)/2}(p-1)p^{(k-1)/2}}.$$
- *For each neuron $\phi(\{u_1, \ldots, u_k, w\}; a_1, \ldots, a_k)$, there is a constant scalar $\beta \in \mathbb{R}$ and a frequency $\zeta \in \{1, \ldots, \frac{p-1}{2}\}$ satisfying*
$$\begin{aligned} u_1(a_1) &= \beta \cdot \cos(\theta^*_{u_1} + 2\pi\zeta a_1/p) \\ u_2(a_2) &= \beta \cdot \cos(\theta^*_{u_2} + 2\pi\zeta a_2/p) \\ &\cdots \\ u_k(a_k) &= \beta \cdot \cos(\theta^*_{u_k} + 2\pi\zeta a_k/p) \\ w(c) &= \beta \cdot \cos(\theta^*_w + 2\pi\zeta c/p), \end{aligned}$$
where $\theta^*_{u_1}, \ldots, \theta^*_{u_k}, \theta^*_w \in \mathbb{R}$ are some phase offsets satisfying $\theta^*_{u_1} + \cdots + \theta^*_{u_k} = \theta^*_w$.
- *For each frequency $\zeta \in \{1, \ldots, \frac{p-1}{2}\}$, there exists one neuron using this frequency only.*

E Notations and Definitions

We use $\mathbf{i}$ to denote $\sqrt{-1}$. Let $z = a + \mathbf{i}b$ denote a complex number where a and b are real numbers. Then we have $\overline{z} = a - \mathbf{i}b$ and $|z| := \sqrt{a^2 + b^2}$.

For any positive integer n, we use $[n]$ to denote set $\{1, 2, \cdots, n\}$. We use $\mathbb{E}[]$ to denote expectation. We use $\Pr[]$ to denote probability. We use $z^\top$ to denote the transpose of a vector z.

Considering a vector z, we denote the ℓ_2 norm as $\|z\|_2 := (\sum_{i=1}^n z_i^2)^{1/2}$. We denote the ℓ_1 norm as $\|z\|_1 := \sum_{i=1}^n |z_i|$. The number of non-zero entries in vector z is defined as $\|z\|_0$. $\|z\|_\infty$ is defined as $\max_{i \in [n]} |z_i|$.

We denote $\mathbb{Z}_p$ as the modular group on p integers, e.g., $\mathbb{Z}_p = [p]$, where $p > 2$ is a given prime number. We denote $\mathcal{X} := \mathbb{Z}_p^k$ as the input space and $\mathcal{Y} := \mathbb{Z}_p$ as the output space. We denote $D_p := \{((a_1, \ldots, a_k), \sum_{i \in [k]} a_i) : a_1, \ldots, a_k \in \mathbb{Z}_p\}$ as the modular dataset.

Suppose we have a $L_{2,k+1}$ (matrix) norm $\|\cdot\|_{2,k+1}$ (Definition C.3) and a class of parameterized functions $\{f(\theta, \cdot) \mid \theta \in \mathbb{R}^U\}$, where $f : \mathbb{R}^U \times \mathcal{X} \to \mathbb{R}^{\mathcal{Y}}$ and a matrix set $\Theta := \{\|\theta\|_{2,k+1} \leq 1\}$. We consider single-hidden layer neural networks with polynomial activation functions and without biases. With parameters $\theta \in \Theta$, we denote $f(\theta, x)$ as the network's output for an input x. For one-hidden layer networks, we denote f as:
$$f(\theta, x) := \sum_{i=1}^m \phi(\theta_i, x),$$
where $\theta := \{\theta_1, \ldots, \theta_m\}$. Here, we denote ϕ as one neuron and $\theta_i \in \Omega$ are the corresponding weights in that neuron, where Ω is a vector set. Let Ω' be a subset of Ω. For every $i \in [m]$, when either there exists $\alpha_i > 0$ such that $\alpha_i \theta_i \in \Omega'$ or $\theta_i = 0$, then we say the parameter set $\theta = \{\theta_1, \ldots, \theta_m\}$ has directional support on Ω'. A single neuron is represented as:
$$\phi(\{u_1, \ldots, u_k, w\}, x_1, \ldots, x_k) := (u_1^\top x_1 + \cdots + u_k^\top x_k)^k w,$$
where $u_1, \ldots, u_k, w \in \mathbb{R}^p$ are the neuron's weights, and $x_1, \ldots, x_k \in \mathbb{R}^p$ are the network inputs. Inputs $x_1, \ldots, x_k$ are one-hot vectors representing group elements. For input elements $(a_1, \ldots, a_k)$, a neuron simplifies to
$$\phi(\{u_1, \ldots, u_k, w\}, a_1, \ldots, a_k) = (u_1(a_1) + \cdots + u_k(a_k))^k w,$$
with $u_1(a_1)$ being the a_1-th component of u_1. The output of the neuron is in p-dimension for cross-entropy loss. With $\theta = \{u_{i,1}, \ldots, u_{i,k}, w_i\}_{i=1}^m$, the network is denoted as:
$$f(\theta, a_1, \ldots, a_k) := \sum_{i=1}^m \phi(\{u_{i,1}, \ldots, u_{i,k}, w_i\}, a_1, \ldots .a_k).$$

We define our regularized training objective function.

Definition E.1. *Let l be the cross-entropy loss. Our regularized training objective function is*

$$\mathcal{L}_\lambda(\theta) := \frac{1}{|D_p|} \sum_{(x,y)\in D_p} l(f(\theta, x), y) + \lambda \|\theta\|_{2,k+1}.$$

Definition E.2. *We define Θ^* as follows*

- $\Theta^* := \arg\max_{\theta\in\Theta} h(\theta)$.

F TECHNIQUE OVERVIEW

In this section, we propose techniques overview of the proof for our main result. We use $\mathbf{i}$ to denote $\sqrt{-1}$. Let $f : \mathbb{Z}_p \to \mathbb{C}$. Then, for each frequency $j \in \mathbb{Z}_p$, we define f discrete Fourier transform (DFT) as

$$\widehat{f}(j) := \sum_{\zeta\in\mathbb{Z}_p} f(\zeta) \exp(-2\pi\mathbf{i} \cdot j\zeta/p).$$

Let $\mathcal{B}$ denote the ball that $\|w\|^2 + \|u_1\|^2 + \cdots + \|u_k\|^2 \le 1$. Let $\Omega_q^{'*}$ be defined as Definition H.6. We first show how we get the single neuron class-weighted maximum-margin solution set $\Omega_q'^*$.

Lemma F.1 (Informal version of Lemma H.8)**.** *If for any $\zeta \in \{1, \ldots, \frac{p-1}{2}\}$, there exists a scaling constant $\beta \in \mathbb{R}$,*

$$\begin{aligned} u_1(a_1) &= \beta \cdot \cos(\theta^*_{u_1} + 2\pi\zeta a_1/p) \\ u_2(a_2) &= \beta \cdot \cos(\theta^*_{u_2} + 2\pi\zeta a_2/p) \\ &\cdots \\ u_k(a_k) &= \beta \cdot \cos(\theta^*_{u_k} + 2\pi\zeta a_k/p) \\ w(c) &= \beta \cdot \cos(\theta^*_w + 2\pi\zeta c/p) \end{aligned}$$

where $\theta^*_{u_1}, \ldots, \theta^*_{u_k}, \theta^*_w \in \mathbb{R}$ are some phase offsets satisfying $\theta^*_{u_1} + \cdots + \theta^*_{u_k} = \theta^*_w$.

Then, we have the following

$$\Omega_q'^* = \{(u_1, \ldots, u_k, w)\},$$

and

$$\gamma^* = \frac{2(k!)}{(2k+2)^{(k+1)/2}(p-1)p^{(k-1)/2}}.$$

Proof sketch of Lemma F.1. The proof establishes the maximum-margin solution's sparsity in the Fourier domain through several key steps. Initially, by Lemma H.7, focus is directed to maximizing Eq. equation 17.

For odd p, Eq. equation 17 can be reformulated with magnitudes and phases of $\widehat{u}_i$ and $\widehat{w}$ (discrete Fourie transform of u_i and w), leading to an equation involving cosine of their phase differences. Plancherel's theorem is then employed to translate the norm constraint to the Fourier domain. This allows for the optimization of the cosine term in the sum, effectively reducing the problem to maximizing the product of magnitudes of $\widehat{u}_i$ and $\widehat{w}$ (Eq. equation 21).

By applying the inequality of arithmetic and geometric means we have an upper bound for the optimization problem. To achieve the upper bound, equal magnitudes are required for all $\widehat{u}_i$ and $\widehat{w}$ at a single frequency, leading to Eq. equation 23. The neurons are finally expressed in the time domain, demonstrating that they assume a specific cosine form with phase offsets satisfying certain conditions. □

See formal proof in Appendix H.3.

Next, we show the number of neurons required to solve the problem and the property of these neurons. We demonstrate how to use these neurons to construct the network θ^*.

Lemma F.2 (Informal version of Lemma I.3)**.** *Let* $\cos_\zeta(x)$ *denote* $\cos(2\pi\zeta x/p)$*. Then, we have the maximum* $L_{2,k+1}$*-margin solution* θ^* *will consist of* $2^{2k-1} \cdot \frac{p-1}{2}$ *neurons* $\theta_i^* \in \Omega_q^{'*}$ *to simulate* $\frac{p-1}{2}$ *type of cosine computation, each cosine computation is uniquely determined a* $\zeta \in \{1, \ldots, \frac{p-1}{2}\}$*. In particular, for each* ζ *the cosine computation is* $\cos_\zeta(a_1 + \cdots + a_k - c), \forall a_1, \ldots, a_k, c \in \mathbb{Z}_p$*.*

Proof sketch of Lemma F.2. Our goal is to show that $2^{2k-1} \cdot \frac{p-1}{2}$ neurons $\theta_i^* \in \Omega_q^{'*}$ are able to simulate $\frac{p-1}{2}$ type of $\cos$ computation.

We first observe that when the $\cos$ of a sum, $\cos(a_1 + a_2)$, is expanded, we will remove one $\cos$ product and we will add zero or two $\sin$ products. On the other hand, expanding the sine of a sum, $\sin(a_1 + a_2)$, we may remove one $\sin$ product and we will add one $\sin$ product as well.

The second observation is about the sign of the terms resulting from these expansions. It notes that a negative sign -1 appears in a term only when a $\cos$ is split, and adding two sine products. Therefore, if the number of sine products in a term is divisible by 4 with a remainder of 2 (i.e., $\%4 = 2$), the term will have a negative sign. In all other cases, the term will have a positive sign.

By using these two observations, we have the following expansion function of $\cos_\zeta(x)$, which denotes $\cos(2\pi\zeta x/p)$.

$$\cos_\zeta(\sum_{i=1}^k a_i) = \sum_{b\in\{0,1\}^k} \prod_{i=1}^k \cos^{1-b_i}(a_i) \cdot \sin^{b_i}(a_i) \cdot \mathbf{1}[\sum_{i=1}^k b_i \% 2 = 0] \cdot (-1)^{\mathbf{1}[\sum_{i=1}^k b_i \% 4 = 2]}.$$

Note that we have 2^k terms in the above equation. By using the following fact in Lemma I.1,

$$2^k \cdot k! \cdot \prod_{i=1}^k a_i = \sum_{c\in\{-1,+1\}^k} (-1)^{(k-\sum_{i=1}^k c_i)/2} (\sum_{j=1}^k c_j a_j)^k,$$

each term can be constructed by 2^{k-1} neurons. Therefore, we need $2^{k-1}2^k$ total neurons. To simulate $\frac{p-1}{2}$ type of simulation, we need $2^{2k-1}\frac{p-1}{2}$ neurons.

Then, using the Lemma G.1, we construct the network θ^*. Finally, by using the Lemma G.2 from Morwani et al. (2024), we know that it is the maximum-margin solution. □

See formal proof in Appendix I.3. Now, we are ready to prove our main results.

Proof sketch of Theorem D.1. By Lemma F.1, we get γ^* and the single-neuron class-weighted maximum-margin solution set $\Omega_q^{'*}$. By satisfying Condition C.10, we know it is used in the maximum-margin solution.

By Lemma F.2, we can construct the network θ^* that uses neurons in $\Omega_q^{'*}$. By Lemma G.2, we know that it is the maximum-margin solution. Finally, by Lemma J.2, we know that all frequencies are covered. □

See formal proof in Appendix K.2.

G Tools from Previous Work

Section G.1 states that we can use the single neuron level optimization to get the maximum-margin network. Section G.2 introduces the maximum-margin for multi-class.

G.1 Tools from Previous Work: Implying Single/Combined Neurons

Lemma G.1 (Lemma 5 in page 8 in Morwani et al. (2024)). *If the following conditions hold*

- *Given* $\Theta := \{\theta : \|\theta\|_{a,b} \leq 1\}$.
- *Given* $\Theta_q'^* := \arg\max_{\theta\in\Theta} \mathbb{E}_{(x,y)\sim q}[g'(\theta, x, y)]$.
- *Given* $\Omega := \{\theta_i : \|\theta_i\|_a \leq 1\}$.
- *Given* $\Omega_q'^* := \arg\max_{\theta_i\in\Omega} \mathbb{E}_{(x,y)\sim q}[\psi'(\theta, x, y)]$.

Then:

- *Let* $\theta \in \Theta_q'^*$. *We have* θ *only has directional support on* $\Omega_q'^*$.
- *Given* $\theta_1^*, \ldots, \theta_m^* \in \Omega_q'^*$, *we have for any set of neuron scalars where* $\sum_{i=1}^m \alpha_i^\nu = 1, \alpha_i \geq 0$, *the weights* $\theta = \{\alpha_i\theta_i^*\}_{i=1}^m$ *is in* $\Theta_q'^*$.

Given q^*, then we can get the θ^* satisfying

$$\theta^* \in \arg\min_{\theta\in\Theta} \mathop{\mathbb{E}}_{(x,y)\sim q^*}[g'(\theta, x, y)]. \tag{8}$$

G.2 Tools from Previous Work: Maximum Margin for Multi-Class

Lemma G.2 (Lemma 6 in page 8 in Morwani et al. (2024)). *If the following conditions hold*

- *Given* $\Theta = \{\theta : \|\theta\|_{a,b} \leq 1\}$ *and* $\Theta_q'^* = \arg\max_{\theta\in\Theta} \mathbb{E}_{(x,y)\sim q}[g'(\theta, x, y)]$.
- *Given* $\Omega = \{\theta_i : \|\theta_i\|_a \leq 1\}$ *and* $\Omega_q'^* = \arg\max_{\theta_i\in\Omega} \mathbb{E}_{(x,y)\sim q}[\psi'(\theta, x, y)]$.
- *Suppose that* $\exists\{\theta^*, q^*\}$ *such that Equations equation 5 and equation 8, and C.10 holds.*

Then, we can show:

- $\theta^* \in \arg\max_{\theta\in\Theta} g(\theta, x, y)$
- $\forall\widehat{\theta} \in \arg\max_{\theta\in\Theta} \min_{(x,y)\in D} g(\theta, x, y)$ *the below properties hold:*

  - $\widehat{\theta}$ *only has directional support on* $\Omega_{q^*}'^*$.

  - $\forall(x, y) \in \mathrm{spt}(q^*), f(\widehat{\theta}, x, y) - \max_{y'\in\mathcal{Y}\setminus\{y\}} f(\widehat{\theta}, x, y') = \gamma^*$.

H Class-weighted Max-margin Solution of Single Neuron

Section H.1 introduces some definitions. Section H.2 shows how we transfer the problem to discrete Fourier space. Section H.3 proposes the weighted margin of the single neuron. Section H.4 shows how we transfer the problem to discrete Fourier space for general k version. Section H.5 provides the solution set for general k version and the maximum weighted margin for a single neuron.

H.1 Definitions

Definition H.1. *When* $k = 3$, *let*

$$\eta_{u_1,u_2,u_3,w}(\delta) := \mathop{\mathbb{E}}_{a_1,a_2,a_3}[(u_1(a_1) + u_2(a_2) + u_3(a_3))^3 w(a_1 + a_2 + a_3 - \delta)].$$

Definition H.2. *When* $k = 3$, *provided the following conditions are met*

- *We denote* $\mathcal{B}$ *as the ball that* $\|u_1\|^2 + \|u_2\|^2 + \|u_3\|^2 + \|w\|^2 \leq 1$.

We define

$$\Omega_q'^* = \mathop{\arg\max}_{u_1,u_2,u_3,w\in\mathcal{B}} (\eta_{u_1,u_2,u_3,w}(0) - \mathop{\mathbb{E}}_{\delta\neq 0}[\eta_{u_1,u_2,u_3,w}(\delta)]).$$

H.2 TRANSFER TO DISCRETE FOURIER SPACE

The goal of this section is to prove the following Lemma,

Lemma H.3. *When $k = 3$, provided the following conditions are met*

- *We denote $\mathcal{B}$ as the ball that $\|u_1\|^2 + \|u_2\|^2 + \|u_3\|^2 + \|w\|^2 \leq 1$.*
- *We define $\Omega_q^{'*}$ in Definition H.2.*
- *We adopt the uniform class weighting: $\forall c' \neq a_1 + a_2 + a_3,\ \ \tau(a_1, a_2, a_3)[c'] := 1/(p-1)$.*

We have the following

$$\Omega_q'^* = \underset{u_1,u_2,u_3,,w\in\mathcal{B}}{\arg\max} \frac{6}{(p-1)p^3} \sum_{j\neq 0} \widehat{u}_1(j)\widehat{u}_2(j)\widehat{u}_3(j)\widehat{w}(-j).$$

Proof. We have

$$\begin{aligned}
\eta_{u_1,u_2,u_3,w}(\delta) &= \mathop{\mathbb{E}}_{a_1,a_2,a_3}[(u_1(a_1) + u_2(a_2) + u_3(a_3))^3 w(a_1 + a_2 + a_3 - \delta)] \\
&= \mathop{\mathbb{E}}_{a_1,a_2,a_3}[(u_1(a_1)^3 + 3u_1(a_1)^2u_2(a_2) + 3u_1(a_1)^2u_3(a_3) + 3u_1(a_1)u_2(a_2)^2 \\
&\quad + 6u_1(a_1)u_2(a_2)u_3(a_3) + 3u_1(a_1)u_3(a_3)^2 + u_2(a_2)^3 + 3u_2(a_2)^2u_3(a_3) \\
&\quad + 3u_2(a_2)u_3(a_3)^2 + u_3(a_3)^3)w(a_1 + a_2 + a_3 - \delta)].
\end{aligned}$$

Recall $\mathcal{B}$ is defined as Lemma Statement.

The goal is to solve the following mean margin maximization problem:

$$\begin{aligned}
&\underset{u_1,u_2,u_3,w\in\mathcal{B}}{\arg\max} (\eta_{u_1,u_2,u_3,w}(0) - \mathop{\mathbb{E}}_{\delta\neq 0}[\eta_{u_1,u_2,u_3,w}(\delta)]) \\
&= \frac{p}{p-1}(\eta_{u_1,u_2,u_3,w}(0) - \mathop{\mathbb{E}}_{\delta}[\eta_{u_1,u_2,u_3,w}(\delta)]),
\end{aligned} \tag{9}$$

where the equation follows $\tau(a_1, a_2, a_3)[c'] := 1/(p-1)\ \ \forall c' \neq a_1 + a_2 + a_3$ and $1 - \frac{1}{p-1} = \frac{p}{p-1}$.

First, note that

$$\begin{aligned}
&\mathop{\mathbb{E}}_{a_1,a_2,a_3}[u_1(a_1)^3 w(a_1 + a_2 + a_3 - \delta)] \\
&= \mathop{\mathbb{E}}_{a_1}[u_1(a_1)^3 \mathop{\mathbb{E}}_{a_2,a_3}[w(a_1 + a_2 + a_3 - \delta)]] \\
&= 0,
\end{aligned}$$

where the first step follows from taking out the $u_1(a_1)$ from the expectation for a_2, a_3, and the last step is from the definition of w.

Similarly for the $u_2(a_2)^3$,$u_3(a_3)^3$ components of η, they equal to 0.

Note that

$$\begin{aligned}
&\mathop{\mathbb{E}}_{a_1,a_2,a_3}[u_1(a_1)^2 u_2(a_2) w(a_1 + a_2 + a_3 - \delta)] \\
&= \mathop{\mathbb{E}}_{a_1}[u_1(a_1)^2 \mathop{\mathbb{E}}_{a_2}[u_2(a_2) \mathop{\mathbb{E}}_{a_3}[w(a_1 + a_2 + a_3 - \delta)]]] \\
&= 0,
\end{aligned}$$

where the first step follows from simple algebra and the last step comes from the definition of w.

Similarly for the $u_1(a_1)^2u_3(a_3)$, $u_2(a_2)^2u_1(a_1)$, $u_2(a_2)^2u_3(a_3)$, $u_3(a_3)^2u_1(a_1)$, $u_3(a_3)^2u_2(a_2)$ components of η, they equal to 0.

Hence, we can rewrite Eq. equation 9 as

$$\arg\max_{u_1,u_2,u_3,w\in\mathcal{B}} \frac{6p}{p-1}(\widetilde{\eta}_{u_1,u_2,u_3,w}(0) - \mathop{\mathbb{E}}_{\delta}[\widetilde{\eta}_{u_1,u_2,u_3,w}(\delta)]),$$

where

$$\widetilde{\eta}_{u_1,u_2,u_3,w}(\delta) := \mathop{\mathbb{E}}_{a_1,a_2,a_3}[u_1(a_1)u_2(a_2)u_3(a_3)w(a_1+a_2+a_3-\delta)].$$

Let $\rho := e^{2\pi \mathbf{i}/p}$, and let $\widehat{u}_1, \widehat{u}_2, \widehat{u}_3, \widehat{w}$ be the DFT of u_1, u_2, u_3, and w respectively:

$$\begin{aligned}
&\widetilde{\eta}_{u_1,u_2,u_3,w}(\delta) \\
&= \mathop{\mathbb{E}}_{a_1,a_2,a_3}[(\frac{1}{p}\sum_{j_1=0}^{p-1}\widehat{u}_1(j_1)\rho^{j_1a_1})(\frac{1}{p}\sum_{j_2=0}^{p-1}\widehat{u}_2(j_2)\rho^{j_2a_2})(\frac{1}{p}\sum_{j_3=0}^{p-1}\widehat{u}_3(j_3)\rho^{j_3a_3})(\frac{1}{p}\sum_{j_4=0}^{p-1}\widehat{w}(j_4)\rho^{j_4(a_1+a_2+a_3-\delta)})] \\
&= \frac{1}{p^4}\sum_{j_1,j_2,j_3,j_4}\widehat{u}_1(j_1)\widehat{u}_2(j_2)\widehat{u}_3(j_3)\widehat{w}(j_4)\rho^{-j_4\delta}(\mathop{\mathbb{E}}_{a_1}[\rho^{(j_1+j_4)a_1}])(\mathop{\mathbb{E}}_{a_2}[\rho^{(j_2+j_4)a_2}])(\mathop{\mathbb{E}}_{a_3}[\rho^{(j_3+j_4)a_3}]) \\
&= \frac{1}{p^4}\sum_{j}\widehat{u}_1(j)\widehat{u}_2(j)\widehat{u}_3(j)\widehat{w}(-j)\rho^{j\delta}
\end{aligned}$$

where the first step follows from $\rho := e^{2\pi \mathbf{i}/p}$ and $\widehat{u}_1, \widehat{u}_2, \widehat{u}_3, \widehat{w}$ are the discrete Fourier transforms of u_1, u_2, u_3, w, the second step comes from simple algebra, the last step is from that only terms where $j_1 + j_4 = j_2 + j_4 = j_3 + j_4 = 0$ survive.

Hence, we need to maximize

$$\begin{aligned}
&\frac{6p}{p-1}(\widetilde{\eta}_{u_1,u_2,u_3,w}(0) - \mathop{\mathbb{E}}_{\delta}[\widetilde{\eta}_{u_1,u_2,u_3,w}(\delta)]) \\
&= \frac{6p}{p-1}(\frac{1}{p^4}\sum_{j}\widehat{u}_1(j)\widehat{u}_2(j)\widehat{u}_3(j)\widehat{w}(-j) - \frac{1}{p^4}\sum_{j}\widehat{u}_1(j)\widehat{u}_2(j)\widehat{u}_3(j)\widehat{w}(-j)(\mathop{\mathbb{E}}_{\delta}\rho^{j\delta})) \\
&= \frac{6}{(p-1)p^3}\sum_{j\neq 0}\widehat{u}_1(j)\widehat{u}_2(j)\widehat{u}_3(j)\widehat{w}(-j). \\
&= \frac{6}{(p-1)p^3}\sum_{j\in[-(p-1)/2,+(p-1)/2]\setminus 0}\widehat{u}_1(j)\widehat{u}_2(j)\widehat{u}_3(j)\widehat{w}(-j). \qquad (10)
\end{aligned}$$

where the first step is from $\widetilde{\eta}_{u_1,u_2,u_3,w}(\delta)$ definition, the second step is from $\mathbb{E}_\delta \rho^{j\delta} = 0$ when $j \neq 0$, and the last step follows from simple algebra.

□

H.3 Get Solution Set

Lemma H.4. *When $k = 3$, provided the following conditions are met*

- *We denote $\mathcal{B}$ as the ball that $\|u_1\|^2 + \|u_2\|^2 + \|u_3\|^2 + \|w\|^2 \leq 1$.*
- *We define $\Omega_q^{'*}$ in Definition H.2.*
- *We adopt the uniform class weighting: $\forall c' \neq a_1 + a_2 + a_3,\ \ \tau(a_1, a_2, a_3)[c'] := 1/(p-1)$.*
- *For any $\zeta \in \{1, \ldots, \frac{p-1}{2}\}$, there exists a scaling constant $\beta \in \mathbb{R}$ and*

$$\begin{aligned}
u_1(a_1) &= \beta \cdot \cos(\theta^*_{u_1} + 2\pi\zeta a_1/p) \\
u_2(a_2) &= \beta \cdot \cos(\theta^*_{u_2} + 2\pi\zeta a_2/p)
\end{aligned}$$

$$u_3(a_3) = \beta \cdot \cos(\theta^*_{u_3} + 2\pi\zeta a_3/p)$$
$$w(c) = \beta \cdot \cos(\theta^*_w + 2\pi\zeta c/p)$$

where $\theta^*_{u_1}, \theta^*_{u_2}, \theta^*_{u_3}, \theta^*_w \in \mathbb{R}$ *are some phase offsets satisfying* $\theta^*_{u_1} + \theta^*_{u_2} + \theta^*_{u_3} = \theta^*_w$.

Then, we have the following

$$\Omega'^*_q = \{(u_1, u_2, u_3, w)\},$$

and

$$\max_{u_1,u_2,u_3,w\in\mathcal{B}} (\eta_{u_1,u_2,u_3,w}(0) - \mathop{\mathbb{E}}_{\delta\neq 0}[\eta_{u_1,u_2,u_3,w}(\delta)]) = \frac{3}{16} \cdot \frac{1}{p(p-1)}.$$

Proof. By Lemma H.3, we only need to maximize Equation equation 10.

Thus, the mass of $\widehat{u}_1, \widehat{u}_2, \widehat{u}_3$, and $\widehat{w}$ must be concentrated on the same frequencies. For all $j \in \mathbb{Z}_p$, we have

$$\widehat{u}_1(-j) = \overline{\widehat{u}_1(j)}, \widehat{u}_2(-j) = \overline{\widehat{u}_2(j)}, \widehat{u}_3(-j) = \overline{\widehat{u}_3(j)}, \widehat{w}(-j) = \overline{\widehat{w}(j)} \tag{11}$$

as u_1, u_2, u_3, w are real-valued.

For all $j \in \mathbb{Z}_p$ and for u_1, u_2, u_3, w, we denote $\theta_{u_1}, \theta_{u_2}, \theta_{u_3}, \theta_w \in [0, 2\pi)^p$ as their phase, e.g.:

$$\widehat{u}_1(j) = |\widehat{u}_1(j)| \exp(\mathbf{i}\theta_{u_1}(j)).$$

Consider the odd p, Equation equation 10 becomes:

$$\begin{aligned}
\textit{equation 10} &= \frac{6}{(p-1)p^3} \sum_{j\in[-(p-1)/2,+(p-1)/2]\setminus 0} \widehat{u}_1(j)\widehat{u}_2(j)\widehat{u}_3(j)\widehat{w}(-j) \\
&= \frac{6}{(p-1)p^3} \sum_{j=1}^{(p-1)/2} (\widehat{u}_1(j)\widehat{u}_2(j)\widehat{u}_3(j)\overline{\widehat{w}(j)} + \overline{\widehat{u}_1(j)\widehat{u}_2(j)\widehat{u}_3(j)}\widehat{w}(j)) \\
&= \frac{6}{(p-1)p^3} \sum_{j=1}^{(p-1)/2} |\widehat{u}_1(j)||\widehat{u}_2(j)||\widehat{u}_3(j)||\widehat{w}(j)| \cdot \\
&\quad \big(\exp(\mathbf{i}(\theta_{u_1}(j) + \theta_{u_2}(j) + \theta_{u_3}(j) - \theta_w(j)) + \exp(\mathbf{i}(-\theta_{u_1}(j) - \theta_{u_2}(j) - \theta_{u_3}(j) + \theta_w(j))\big) \\
&= \frac{12}{(p-1)p^3} \sum_{j=1}^{(p-1)/2} |\widehat{u}_1(j)||\widehat{u}_2(j)||\widehat{u}_3(j)||\widehat{w}(j)| \cos(\theta_{u_1}(j) + \theta_{u_2}(j) + \theta_{u_3}(j) - \theta_w(j)).
\end{aligned}$$

where the first step comes from definition equation 10, the second step follows from Eq. equation 11, the third step comes from $\widehat{u}_1(-j) = \overline{\widehat{u}_1(j)}$ and $\widehat{u}_1(j) = |\widehat{u}_1(j)| \exp(\mathbf{i}\theta_{u_1}(j))$, the last step follow from Euler's formula.

Thus, we need to optimize:

$$\max_{u_1,u_2,u_3,w\in\mathcal{B}} \frac{12}{(p-1)p^3} \sum_{j=1}^{(p-1)/2} |\widehat{u}_1(j)||\widehat{u}_2(j)||\widehat{u}_3(j)||\widehat{w}(j)| \cos(\theta_{u_1}(j) + \theta_{u_2}(j) + \theta_{u_3}(j) - \theta_w(j)). \tag{12}$$

The norm constraint $\|u_1\|^2 + \|u_2\|^2 + \|u_3\|^2 + \|w\|^2 \leq 1$ is equivalent to

$$\|\widehat{u}_1\|^2 + \|\widehat{u}_2\|^2 + \|\widehat{u}_3\|^2 + \|\widehat{w}\|^2 \leq p$$

by using Plancherel's theorem. Thus, we need to select them in such a way that

$$\theta_{u_1}(j) + \theta_{u_2}(j) + \theta_{u_3}(j) = \theta_w(j),$$

ensuring that, for each j, the expression $\cos(\theta_{u_1}(j) + \theta_{u_2}(j) + \theta_{u_3}(j) - \theta_w(j)) = 1$ is maximized, except in cases where the scalar of the j-th term is 0.

This further simplifies the problem to:

$$\max_{|\widehat{u}_1|,|\widehat{u}_2|,|\widehat{u}_3|,|\widehat{w}|:\|\widehat{u}_1\|^2+\|\widehat{u}_2\|^2+\|\widehat{u}_3\|^2+\|\widehat{w}\|^2\leq p} \frac{12}{(p-1)p^3} \sum_{j=1}^{(p-1)/2} |\widehat{u}_1(j)||\widehat{u}_2(j)||\widehat{u}_3(j)||\widehat{w}(j)|. \quad (13)$$

Then, we have

$$|\widehat{u}_1(j)||\widehat{u}_2(j)||\widehat{u}_3(j)||\widehat{w}(j)| \leq (\frac{1}{4} \cdot (|\widehat{u}_1(j)|^2 + |\widehat{u}_2(j)|^2 + |\widehat{u}_3(j)|^2 + |\widehat{w}(j)|^2))^2. \quad (14)$$

where the first step is from inequality of quadratic and geometric means.

We define $z : \{1, \ldots, \frac{p-1}{2}\} \to \mathbb{R}$ as

$$z(j) := |\widehat{u}_1(j)|^2 + |\widehat{u}_2(j)|^2 + |\widehat{u}_3(j)|^2 + |\widehat{w}(j)|^2.$$

We need to have $\widehat{u}_1(0) = \widehat{u}_2(0) = \widehat{u}_3(0) = \widehat{w}(0) = 0$. Then, the upper-bound of Eq. equation 13 is given by

$$\begin{aligned}
&\frac{12}{(p-1)p^3} \cdot \max_{\|z\|_1 \leq \frac{p}{2}} \sum_{j=1}^{(p-1)/2} (\frac{z(j)}{4})^2 \\
&= \frac{3}{4(p-1)p^3} \cdot \max_{\|z\|_1 \leq \frac{p}{2}} \sum_{j=1}^{(p-1)/2} z(j)^2 \\
&= \frac{3}{4(p-1)p^3} \cdot \max_{\|z\|_1 \leq \frac{p}{2}} \|z\|_2^2 \\
&\leq \frac{3}{4(p-1)p^3} \cdot \frac{p^2}{4} \\
&= \frac{3}{16} \cdot \frac{1}{p(p-1)},
\end{aligned}$$

where the first step follows from simple algebra, the second step comes from the definition of L_2 norm, the third step follows from $\|z\|_2 \leq \|z\|_1 \leq \frac{p}{2}$, the last step comes from simple algebra.

For the inequality of quadratic and geometric means, Eq. equation 14 becomes equality when $|\widehat{u}_1(j)| = |\widehat{u}_2(j)| = |\widehat{u}_3(j)| = |\widehat{w}(j)|$. To achieve $\|z\|_2 = \frac{p}{2}$, all the mass must be placed on a single frequency. Hence, for some frequency $\zeta \in \{1, \ldots, \frac{p-1}{2}\}$, to achieve the upper bound, we have:

$$|\widehat{u}_1(j)| = |\widehat{u}_2(j)| = |\widehat{u}_3(j)| = |\widehat{w}(j)| = \begin{cases} \sqrt{p/8} & \text{if } j = \pm\zeta \\ 0 & \text{otherwise} \end{cases}, \quad (15)$$

In this case, Eq. equation 13 matches the upper bound.

$$\frac{12}{(p-1)p^3} \cdot (\frac{p}{8})^2 = \frac{3}{16} \cdot \frac{1}{p(p-1)},$$

where the first step is by simple algebra. Hence, the maximum-margin is $\frac{3}{16} \cdot \frac{1}{p(p-1)}$.

Let $\theta^*_{u_1} := \theta_{u_1}(\zeta)$. Combining all the results, up to scaling, it is established that all neurons which maximize the expected class-weighted margin conform to the form:

$$u_1(a_1) = \frac{1}{p} \sum_{j=0}^{p-1} \widehat{u}_1(j) \rho^{j a_1}$$

$$
\begin{aligned}
&= \frac{1}{p} \cdot (\widehat{u}_1(\zeta)\rho^{\zeta a_1} + \widehat{u}_1(-\zeta)\rho^{-\zeta a_1}) \\
&= \frac{1}{p} \cdot (\sqrt{\frac{p}{8}} \exp(\mathbf{i}\theta^*_{u_1})\rho^{\zeta a_1} + \sqrt{\frac{p}{8}} \exp(-\mathbf{i}\theta^*_{u_1})\rho^{-\zeta a_1}) \\
&= \sqrt{\frac{1}{2p}} \cos(\theta^*_{u_1} + 2\pi\zeta a_1/p),
\end{aligned}
$$

where the first step comes from the definition of $u_1(a)$, the second step and third step follow from Eq. equation 15, the last step follows from Euler's formula.

Similarly,

$$
\begin{aligned}
u_2(a_2) &= \sqrt{\frac{1}{2p}} \cos(\theta^*_{u_2} + 2\pi\zeta a_2/p) \\
u_3(a_3) &= \sqrt{\frac{1}{2p}} \cos(\theta^*_{u_3} + 2\pi\zeta a_3/p) \\
w(c) &= \sqrt{\frac{1}{2p}} \cos(\theta^*_{w} + 2\pi\zeta c/p),
\end{aligned}
$$

for some phase offsets $\theta^*_{u_1}, \theta^*_{u_2}, \theta^*_{u_3}, \theta^*_w \in \mathbb{R}$ satisfying $\theta^*_{u_1} + \theta^*_{u_2} + \theta^*_{u_3} = \theta^*_w$ and some $\zeta \in \mathbb{Z}_p \backslash \{0\}$, where u_1, u_2, u_3, and w shares the same ζ.

□

H.4 Transfer to Discrete Fourier Space for General k Version

Definition H.5. *Let*

$$
\eta_{u_1,\ldots,u_k,w}(\delta) := \mathop{\mathbb{E}}_{a_1,\ldots,a_k}[(u_1(a_1) + \cdots + u_k(a_k))^k w(a_1 + \cdots + a_k - \delta)].
$$

Definition H.6. *Provided the following conditions are met*

- *We denote $\mathcal{B}$ as the ball that $\|u_1\|^2 + \cdots + \|u_k\|^2 + \|w\|^2 \leq 1$.*

We define

$$
\Omega'^*_q = \mathop{\arg\max}_{u_1,\ldots,u_k,w \in \mathcal{B}} (\eta_{u_1,\ldots,u_k,w}(0) - \mathop{\mathbb{E}}_{\delta \neq 0}[\eta_{u_1,\ldots,u_k,w}(\delta)]).
$$

The goal of this section is to prove the following Lemma,

Lemma H.7. *Provided the following conditions are met*

- *Let $\mathcal{B}$ denote the ball that $\|u_1\|^2 + \cdots + \|u_k\|^2 + \|w\|^2 \leq 1$.*
- *We define Ω'^*_q in Definition H.6.*
- *We adopt the uniform class weighting: $\forall c' \neq a_1 + \cdots + a_k,\ \ \tau(a_1, \ldots, a_k)[c'] := 1/(p-1)$.*

We have the following

$$
\Omega'^*_q = \mathop{\arg\max}_{u_1,\ldots,u_k,w \in \mathcal{B}} \frac{k!}{(p-1)p^k} \sum_{j \neq 0} \widehat{w}(-j) \prod_{i=1}^{k} \widehat{u}_i(j).
$$

Proof. We have

$$
\eta_{u_1,\ldots,u_k,w}(\delta) = \mathop{\mathbb{E}}_{a_1,\ldots,a_k}[(u_1(a_1) + \cdots + u_k(a_k))^k w(a_1 + \cdots + a_k - \delta)].
$$

The goal is to solve the following mean margin maximization problem:

$$
\mathop{\arg\max}_{u_1,\ldots,u_k,w \in \mathcal{B}} (\eta_{u_1,\ldots,u_k,w}(0) - \mathop{\mathbb{E}}_{\delta \neq 0}[\eta_{u_1,\ldots,u_k,w}(\delta)])
$$

$$= \frac{p}{p-1}(\eta_{u_1,\dots,u_k,w}(0) - \mathop{\mathbb{E}}_{\delta}[\eta_{u_1,\dots,u_k,w}(\delta)]), \tag{16}$$

where the equation follows $\tau(a_1,\dots,a_k)[c'] := 1/(p-1)\ \ \forall c' \neq a_1+\cdots+a_k$ and $1-\frac{1}{p-1} = \frac{p}{p-1}$.

We note that all terms are zero rather than $w(\cdot)\cdot\prod_{i=1}^k u_i(a_i)$.

Hence, we can rewrite Eq. equation 16 as

$$\arg\max_{u_1,\dots,u_k,w\in\mathcal{B}} \frac{k!p}{p-1}(\widetilde{\eta}_{u_1,\dots,u_k,w}(0) - \mathop{\mathbb{E}}_{\delta}[\widetilde{\eta}_{u_1,\dots,u_k,w}(\delta)]),$$

where

$$\widetilde{\eta}_{u_1,\dots,u_k,w}(\delta) := \mathop{\mathbb{E}}_{a_1,\dots,a_k}[w(a_1+\cdots+a_k-\delta)\prod_{i=1}^k u_i(a_i)].$$

Let $\rho := e^{2\pi\mathbf{i}/p}$, and $\widehat{u}_1,\dots,\widehat{u}_k,\widehat{w}$ denote the discrete Fourier transforms of $u_1,\dots,u_k$, and w respectively. We have

$$\widetilde{\eta}_{u_1,\dots,u_k,w}(\delta) = \frac{1}{p^{k+1}}\sum_{j=0}^{p-1}\widehat{w}(-j)\rho^{j\delta}\prod_{i=1}^k\widehat{u}_i(j)$$

which comes from $\rho := e^{2\pi\mathbf{i}/p}$ and $\widehat{u}_1,\dots,\widehat{u}_k,\widehat{w}$ are the discrete Fourier transforms of $u_1,\dots,u_k,w$.

Hence, we need to maximize

$$\begin{aligned}
&\frac{k!p}{p-1}(\widetilde{\eta}_{u_1,\dots,u_k,w}(0) - \mathop{\mathbb{E}}_{\delta}[\widetilde{\eta}_{u_1,\dots,u_k,w}(\delta)]) \\
&= \frac{k!p}{p-1}\cdot\left(\frac{1}{p^{k+1}}\sum_{j=0}^{p-1}\widehat{w}(-j)\prod_{i=1}^k\widehat{u}_i(j) - \frac{1}{p^{k+1}}\sum_{j=0}^{p-1}\widehat{w}(-j)(\mathop{\mathbb{E}}_{\delta}[\rho^{j\delta}])\prod_{i=1}^k\widehat{u}_i(j)\right) \\
&= \frac{k!}{(p-1)p^k}\sum_{j\neq 0}\widehat{w}(-j)\prod_{i=1}^k\widehat{u}_i(j). \\
&= \frac{k!}{(p-1)p^k}\sum_{j\in[-(p-1)/2,+(p-1)/2]\setminus 0}\widehat{w}(-j)\prod_{i=1}^k\widehat{u}_i(j).
\end{aligned} \tag{17}$$

where the first step follows from the definition of $\widetilde{\eta}_{u_1,\dots,u_k,w}(\delta)$, the second step follows from $\mathbb{E}_\delta[\rho^{j\delta}] = 0$ when $j\neq 0$, the last step is from simple algebra.

□

H.5 Get Solution Set for General k Version

Lemma H.8 (Formal version of Lemma F.1)**.** *Provided the following conditions are met*

- *We denote $\mathcal{B}$ as the ball that $\|u_1\|^2+\cdots+\|u_k\|^2+\|w\|^2\leq 1$.*
- *Let $\Omega_q^{'*}$ be defined as Definition H.6.*
- *We adopt the uniform class weighting: $\forall c'\neq a_1+\cdots+a_k,\ \ \tau(a_1,\dots,a_k)[c'] := 1/(p-1)$.*
- *For any $\zeta\in\{1,\dots,\frac{p-1}{2}\}$, there exists a scaling constant $\beta\in\mathbb{R}$ and*

$$u_1(a_1) = \beta\cdot\cos(\theta^*_{u_1} + 2\pi\zeta a_1/p)$$

$$u_2(a_2) = \beta \cdot \cos(\theta^*_{u_2} + 2\pi\zeta a_2/p)$$
$$\cdots$$
$$u_k(a_k) = \beta \cdot \cos(\theta^*_{u_k} + 2\pi\zeta a_k/p)$$
$$w(c) = \beta \cdot \cos(\theta^*_w + 2\pi\zeta c/p)$$

where $\theta^*_{u_1}, \ldots, \theta^*_{u_k}, \theta^*_w \in \mathbb{R}$ *are some phase offsets satisfying* $\theta^*_{u_1} + \cdots + \theta^*_{u_k} = \theta^*_w$.

Then, we have the following

$$\Omega'^*_q = \{(u_1, \ldots, u_k, w)\},$$

and

$$\max_{u_1,\ldots,u_k,w\in\mathcal{B}} (\eta_{u_1,\ldots,u_k,w}(0) - \mathop{\mathbb{E}}_{\delta\neq 0}[\eta_{u_1,\ldots,u_k,w}(\delta)]) = \frac{2(k!)}{(2k+2)^{(k+1)/2}(p-1)p^{(k-1)/2}}.$$

Proof. By Lemma H.7, we only need to maximize Equation equation 17. Thus, the mass of $\widehat{u}_1, \ldots, \widehat{u}_k$, and $\widehat{w}$ must be concentrated on the same frequencies. For all $j \in \mathbb{Z}_p$, we have

$$\widehat{u}_i(-j) = \overline{\widehat{u}_i(j)}, \quad \widehat{w}(-j) = \overline{\widehat{w}(j)} \tag{18}$$

as $u_1, \ldots, u_k, w$ are real-valued. For all $j \in \mathbb{Z}_p$ and for u_1, u_2, u_3, w, we denote $\theta_{u_1}, \ldots, \theta_{u_k}, \theta_w \in [0, 2\pi)^p$ as their phase, e.g.:

$$\widehat{u}_1(j) = |\widehat{u}_1(j)| \exp(\mathbf{i}\theta_{u_1}(j)). \tag{19}$$

Considering odd p, Equation equation 17 becomes:

$$\begin{aligned}
\textit{equation } 17 &= \frac{k!}{(p-1)p^k} \sum_{j\in[-(p-1)/2,+(p-1)/2]\setminus 0} \widehat{w}(-j) \prod_{i=1}^{k} \widehat{u}_i(j) \\
&= \frac{k!}{(p-1)p^k} \sum_{j=1}^{(p-1)/2} (\prod_{i=1}^{k} \widehat{u}_i(j)\overline{\widehat{w}(j)} + \widehat{w}(j) \prod_{i=1}^{k} \overline{\widehat{u}_i(j)}) \\
&= \frac{2(k!)}{(p-1)p^k} \sum_{j=1}^{(p-1)/2} |\widehat{w}(j)| \cos(\sum_{i=1}^{k} \theta_{u_i}(j) - \theta_w(j)) \prod_{i=1}^{k} |\widehat{u}_i(j)|.
\end{aligned}$$

where the first step follows from definition equation 17, the second step comes from Eq. equation 18, the last step follows from Eq. equation 19, i.e., Euler's formula.

Thus, we need to optimize:

$$\max_{u_1,\ldots,u_k,w\in\mathcal{B}} \frac{2(k!)}{(p-1)p^k} \sum_{j=1}^{(p-1)/2} |\widehat{w}(j)| \cos(\sum_{i=1}^{k} \theta_{u_i}(j) - \theta_w(j)) \prod_{i=1}^{k} |\widehat{u}_i(j)|. \tag{20}$$

We can transfer the norm constraint to

$$\|\widehat{u}_1\|^2 + \cdots + \|\widehat{u}_k\|^2 + \|\widehat{w}\|^2 \leq p$$

by using Plancherel's theorem.

Therefore, we need to select them in a such way that $\theta_{u_1}(j) + \cdots + \theta_{u_k}(j) = \theta_w(j)$, ensuring that, for each j, the expression $\cos(\theta_{u_1}(j) + \cdots + \theta_{u_k}(j) - \theta_w(j)) = 1$ is maximized, except in cases where the scalar of the j-th term is 0.

This further simplifies the problem to:

$$\max_{\|\widehat{u}_1\|^2+\cdots+\|\widehat{u}_k\|^2+\|\widehat{w}\|^2\leq p} \frac{2(k!)}{(p-1)p^k} \sum_{j=1}^{(p-1)/2} |\widehat{w}(j)| \prod_{i=1}^{k} |\widehat{u}_i(j)|. \tag{21}$$

Then, we have

$$|\widehat{w}(j)|\prod_{i=1}^{k}|\widehat{u}_i(j)| \leq (\frac{1}{k+1}\cdot(|\widehat{u}_1(j)|^2+\cdots+|\widehat{u}_k(j)|^2+|\widehat{w}(j)|^2))^{(k+1)/2}. \tag{22}$$

where the first step follows from inequality of quadratic and geometric means.

We define $z:\{1,\ldots,\frac{p-1}{2}\}\to\mathbb{R}$, where

$$z(j) := |\widehat{u}_1(j)|^2+\cdots+|\widehat{u}_k(j)|^2+|\widehat{w}(j)|^2.$$

We need to have $\widehat{u}_1(0)=\cdots=\widehat{u}_k(0)=\widehat{w}(0)=0$. Then, the upper-bound of Equation equation 21 is given by

$$\begin{aligned}
&\frac{2(k!)}{(p-1)p^k}\cdot\max_{\|z\|_1\leq\frac{p}{2}}\sum_{j=1}^{(p-1)/2}(\frac{z(j)}{k+1})^{(k+1)/2}\\
&=\frac{2(k!)}{(k+1)^{(k+1)/2}(p-1)p^k}\cdot\max_{\|z\|_1\leq\frac{p}{2}}\sum_{j=1}^{(p-1)/2}z(j)^{(k+1)/2}\\
&\leq\frac{2(k!)}{(k+1)^{(k+1)/2}(p-1)p^k}\cdot(p/2)^{(k+1)/2}\\
&=\frac{2(k!)}{(2k+2)^{(k+1)/2}(p-1)p^{(k-1)/2}},
\end{aligned}$$

where the first step follows from simple algebra, the second step comes from the definition of L_2 norm, the third step follows from $\|z\|_2\leq\|z\|_1\leq\frac{p}{2}$, the last step follows from simple algebra.

For the inequality of quadratic and geometric means, Eq. equation 22 becomes equality when $|\widehat{u}_1(j)|=\cdots=|\widehat{u}_k(j)|=|\widehat{w}(j)|$. To achieve $\|z\|_2=\frac{p}{2}$, all the mass must be placed on a single frequency. Hence, for some frequency $\zeta\in\{1,\ldots,\frac{p-1}{2}\}$, to achieve the upper bound, we have:

$$|\widehat{u}_1(j)|=\cdots=|\widehat{u}_k(j)|=|\widehat{w}(j)|=\begin{cases}\sqrt{\frac{p}{2(k+1)}}, & \text{if } j=\pm\zeta;\\ 0, & \text{otherwise.}\end{cases} \tag{23}$$

In this case, Equation equation 21 matches the upper bound. Hence, this is the maximum-margin.

Let $\theta^*_{u_1}:=\theta_{u_1}(\zeta)$. Combining all the results, up to scaling, it is established that all neurons which maximize the expected class-weighted margin conform to the form:

$$\begin{aligned}
u_1(a_1)&=\frac{1}{p}\sum_{j=0}^{p-1}\widehat{u}_1(j)\rho^{ja_1}\\
&=\frac{1}{p}\cdot(\widehat{u}_1(\zeta)\rho^{\zeta a_1}+\widehat{u}_1(-\zeta)\rho^{-\zeta a_1})\\
&=\frac{1}{p}\cdot(\sqrt{\frac{p}{2(k+1)}}\exp(\mathbf{i}\theta^*_{u_1})\rho^{\zeta a_1}+\sqrt{\frac{p}{2(k+1)}}\exp(-\mathbf{i}\theta^*_{u_1})\rho^{-\zeta a_1})\\
&=\sqrt{\frac{2}{(k+1)p}}\cos(\theta^*_{u_1}+2\pi\zeta a_1/p),
\end{aligned}$$

where the first step comes from the definition of $u_1(a)$, the second step and third step follow from Eq. equation 23, the last step follows from Eq. equation 19 i.e., Euler's formula.

We have similar results for other neurons where $\theta^*_{u_1},\ldots,\theta^*_{u_k},\theta^*_w\in\mathbb{R}$ satisfying $\theta^*_{u_1}+\cdots+\theta^*_{u_k}=\theta^*_w$ and some $\zeta\in\mathbb{Z}_p\backslash\{0\}$, where $u_1,\ldots,u_k$, and w shares the same ζ.

□

I CONSTRUCT MAX MARGIN SOLUTION

Section I.1 proposed the sum-to-product identities for k inputs. Section I.2 shows how we construct θ^* when $k = 3$. Section I.3 gives the constructions for θ^* for general k version.

I.1 SUM-TO-PRODUCT IDENTITIES

Lemma I.1 (Sum-to-product Identities)**.** *If the following conditions hold*

- *Let $a_1, \ldots, a_k$ denote any k real numbers*

We have

- **Part 1.**

$$2^2 \cdot 2! \cdot a_1 a_2 = (a_1 + a_2)^2 - (a_1 - a_2)^2 - (-a_1 + a_2)^2 + (-a_1 - a_2)^2$$

- **Part 2.**

$$\begin{aligned} 2^3 \cdot 3! \cdot a_1 a_2 a_3 = {} & (a_1 + a_2 + a_3)^3 - (a_1 + a_2 - a_3)^3 - (a_1 - a_2 + a_3)^3 - (-a_1 + a_2 + a_3)^3 \\ & + (a_1 - a_2 - a_3)^3 + (-a_1 + a_2 - a_3)^3 + (-a_1 - a_2 + a_3)^3 - (-a_1 - a_2 - a_3)^3 \end{aligned}$$

- **Part 3.**

$$\begin{aligned} & 2^4 \cdot 4! \cdot a_1 a_2 a_3 a_4 = \\ & (a_1 + a_2 + a_3 + a_4)^4 \\ & - (a_1 + a_2 + a_3 - a_4)^4 - (a_1 + a_2 - a_3 + a_4)^4 - (a_1 - a_2 + a_3 + a_4)^4 - (-a_1 + a_2 + a_3 + a_4)^4 \\ & + (a_1 + a_2 - a_3 - a_4)^4 + (a_1 - a_2 + a_3 - a_4)^4 + (a_1 - a_2 - a_3 + a_4)^4 \\ & + (-a_1 - a_2 + a_3 + a_4)^4 + (-a_1 + a_2 - a_3 + a_4)^4 + (-a_1 + a_2 + a_3 - a_4)^4 \\ & - (-a_1 - a_2 - a_3 + a_4)^4 - (-a_1 - a_2 + a_3 - a_4)^4 - (-a_1 + a_2 - a_3 - a_4)^4 - (a_1 - a_2 - a_3 - a_4)^4 \\ & + (-a_1 - a_2 - a_3 - a_4)^4 \end{aligned}$$

- **Part 4.**

$$2^k \cdot k! \cdot \prod_{i=1}^{k} a_i = \sum_{c \in \{-1,+1\}^k} (-1)^{(k - \sum_{i=1}^k c_i)/2} (\sum_{j=1}^{k} c_j a_j)^k.$$

Proof. **Proof of Part 1.**

We define A_1, A_2, A_3, A_4 as follows

$$A_1 := (a_1 + a_2)^2, A_2 := (a_1 - a_2)^2, A_3 := (-a_1 + a_2)^2, A_4 := (-a_1 - a_2)^2,$$

For the first term, we have

$$A_1 = a_1^2 + a_2^2 + 2a_1 a_2.$$

For the second term, we have

$$A_2 = a_1^2 + a_2^2 - 2a_1 a_2.$$

For the third term, we have

$$A_3 = a_1^2 + a_2^2 - 2a_1 a_2.$$

For the fourth term, we have

$$A_4 = a_1^2 + a_2^2 + 2a_1 a_2.$$

Putting things together, we have

$$\begin{aligned}
&(a_1+a_2)^2-(a_1-a_2)^2-(-a_1+a_2)^2+(-a_1-a_2)^2\\
=&A_1-A_2-A_3+A_4\\
=&8a_1a_2\\
=&2^3a_1a_2
\end{aligned}$$

Proof of Part 2.

We define $B_1, B_2, B_3, B_4, B_5, B_6, B_7, B_8$ as follows

$$\begin{aligned}
B_1 &:= (a_1+a_2+a_3)^3,\\
B_2 &:= (a_1+a_2-a_3)^3,\\
B_3 &:= (a_1-a_2+a_3)^3,\\
B_4 &:= (-a_1+a_2+a_3)^3,\\
B_5 &:= (a_1-a_2-a_3)^3,\\
B_6 &:= (-a_1+a_2-a_3)^3,\\
B_7 &:= (-a_1-a_2+a_3)^3,\\
B_8 &:= (-a_1-a_2-a_3)^3,
\end{aligned}$$

For the first term, we have

$$B_1 = a_1^3+a_2^3+a_3^3+3a_2a_3^2+3a_2a_1^2+3a_1a_3^2+3a_1a_2^2+3a_3a_1^2+3a_3a_2^2+6a_1a_2a_3.$$

For the second term, we have

$$B_2 = a_1^3+a_2^3-a_3^3+3a_2a_3^2+3a_2a_1^2+3a_1a_3^2+3a_1a_2^2-3a_3a_1^2-3a_3a_2^2-6a_1a_2a_3.$$

For the third term, we have

$$B_3 = a_1^3-a_2^3+a_3^3-3a_2a_3^2-3a_2a_1^2+3a_1a_3^2+3a_1a_2^2+3a_3a_1^2+3a_3a_2^2-6a_1a_2a_3.$$

For the fourth term, we have

$$B_4 = -a_1^3+a_2^3+a_3^3+3a_2a_3^2+3a_2a_1^2-3a_1a_3^2-3a_1a_2^2+3a_3a_1^2+3a_3a_2^2-6a_1a_2a_3.$$

For the fifth term, we have

$$B_5 = a_1^3-a_2^3-a_3^3-3a_2a_3^2-3a_2a_1^2+3a_1a_3^2+3a_1a_2^2-3a_3a_1^2-3a_3a_2^2+6a_1a_2a_3.$$

For the sixth term, we have

$$B_6 = -a_1^3+a_2^3-a_3^3+3a_2a_3^2+3a_2a_1^2-3a_1a_3^2-3a_1a_2^2-3a_3a_1^2-3a_3a_2^2+6a_1a_2a_3.$$

For the seventh term, we have

$$B_7 = -a_1^3-a_2^3+a_3^3-3a_2a_3^2-3a_2a_1^2-3a_1a_3^2-3a_1a_2^2+3a_3a_1^2+3a_3a_2^2+6a_1a_2a_3.$$

For the eighth term, we have

$$B_8 = -a_1^3-a_2^3-a_3^3-3a_2a_3^2-3a_2a_1^2-3a_1a_3^2-3a_1a_2^2-3a_3a_1^2-3a_3a_2^2-6a_1a_2a_3.$$

Putting things together, we have

$$\begin{aligned}
&(a_1+a_2+a_3)^3-(a_1+a_2-a_3)^3-(a_1-a_2+a_3)^3-(-a_1+a_2+a_3)^3\\
&+(a_1-a_2-a_3)^3+(-a_1+a_2-a_3)^3+(-a_1-a_2+a_3)^3-(-a_1-a_2-a_3)^3\\
=&B_1-B_2-B_3-B_4+B_5+B_6+B_7-B_8\\
=&48a_1a_2a_3\\
=&3\cdot 2^4a_1a_2a_3
\end{aligned}$$

Proof of Part 3.

We define $C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8, C_9, C_{10}, C_{11}, C_{12}, C_{13}, C_{14}, C_{15}, C_{16}$ as follows

$$\begin{aligned}
C_1 &:= (a_1 + a_2 + a_3 + a_4)^4, \\
C_2 &:= (a_1 + a_2 + a_3 - a_4)^4, \\
C_3 &:= (a_1 + a_2 - a_3 + a_4)^4, \\
C_4 &:= (a_1 - a_2 + a_3 + a_4)^4, \\
C_5 &:= (-a_1 + a_2 + a_3 + a_4)^4, \\
C_6 &:= (a_1 + a_2 - a_3 - a_4)^4, \\
C_7 &:= (a_1 - a_2 + a_3 - a_4)^4, \\
C_8 &:= (a_1 - a_2 - a_3 + a_4)^4, \\
C_9 &:= (-a_1 - a_2 + a_3 + a_4)^4, \\
C_{10} &:= (-a_1 + a_2 - a_3 + a_4)^4, \\
C_{11} &:= (-a_1 + a_2 + a_3 - a_4)^4, \\
C_{12} &:= (-a_1 - a_2 - a_3 + a_4)^4, \\
C_{13} &:= (-a_1 - a_2 + a_3 - a_4)^4, \\
C_{14} &:= (-a_1 + a_2 - a_3 - a_4)^4, \\
C_{15} &:= (a_1 - a_2 - a_3 - a_4)^4, \\
C_{16} &:= (-a_1 - a_2 - a_3 - a_4)^4,
\end{aligned}$$

For the first term, we have

$$\begin{aligned}
C_1 = {} & a_4^4 + 4a_1a_4^3 + 4a_2a_4^3 + 4a_3a_4^3 + 12a_1a_2a_4^2 + 12a_1a_3a_4^2 + 12a_2a_3a_4^2 + 24a_1a_2a_3a_4 \\
& + 12a_1a_4a_2^2 + 12a_1a_4a_3^2 + 12a_2a_4a_1^2 + 12a_2a_4a_3^2 + 12a_3a_4a_1^2 + 12a_3a_4a_2^2 \\
& + 4a_4a_1^3 + 4a_4a_2^3 + 4a_4a_3^3 + 6(a_1a_2)^2 + 6(a_1a_3)^2 + 6(a_1a_4)^2 + 6(a_2a_3)^2 + 6(a_2a_4)^2 + 6(a_3a_4)^2 \\
& + 12a_1a_2a_3^2 + 12a_1a_3a_2^2 + 12a_2a_3a_1^2 + 4a_1a_2^3 + 4a_1a_3^3 + 4a_2a_1^3 + 4a_2a_3^3 + 4a_3a_1^3 + 4a_3a_2^3 \\
& + a_1^4 + a_2^4 + a_3^4.
\end{aligned}$$

For the second term, we have

$$\begin{aligned}
C_2 = {} & a_4^4 - 4a_1a_4^3 - 4a_2a_4^3 - 4a_3a_4^3 + 12a_1a_2a_4^2 + 12a_1a_3a_4^2 + 12a_2a_3a_4^2 - 24a_1a_2a_3a_4 \\
& - 12a_1a_4a_2^2 - 12a_1a_4a_3^2 - 12a_2a_4a_1^2 - 12a_2a_4a_3^2 - 12a_3a_4a_1^2 - 12a_3a_4a_2^2 \\
& - 4a_4a_1^3 - 4a_4a_2^3 - 4a_4a_3^3 + 6(a_1a_2)^2 + 6(a_1a_3)^2 + 6(a_1a_4)^2 + 6(a_2a_3)^2 + 6(a_2a_4)^2 + 6(a_3a_4)^2 \\
& + 12a_1a_2a_3^2 + 12a_1a_3a_2^2 + 12a_2a_3a_1^2 + 4a_1a_2^3 + 4a_1a_3^3 + 4a_2a_1^3 + 4a_2a_3^3 + 4a_3a_1^3 + 4a_3a_2^3 \\
& + a_1^4 + a_2^4 + a_3^4.
\end{aligned}$$

For the third term, we have

$$\begin{aligned}
C_3 = {} & a_4^4 + 4a_1a_4^3 + 4a_2a_4^3 - 4a_3a_4^3 + 12a_1a_2a_4^2 - 12a_1a_3a_4^2 - 12a_2a_3a_4^2 - 24a_1a_2a_3a_4 \\
& + 12a_1a_4a_2^2 + 12a_1a_4a_3^2 + 12a_2a_4a_1^2 + 12a_2a_4a_3^2 - 12a_3a_4a_1^2 - 12a_3a_4a_2^2 \\
& + 4a_4a_1^3 + 4a_4a_2^3 - 4a_4a_3^3 + 6(a_1a_2)^2 + 6(a_1a_3)^2 + 6(a_1a_4)^2 + 6(a_2a_3)^2 + 6(a_2a_4)^2 + 6(a_3a_4)^2 \\
& + 12a_1a_2a_3^2 - 12a_1a_3a_2^2 - 12a_2a_3a_1^2 + 4a_1a_2^3 - 4a_1a_3^3 + 4a_2a_1^3 - 4a_2a_3^3 - 4a_3a_1^3 - 4a_3a_2^3 \\
& + a_1^4 + a_2^4 + a_3^4.
\end{aligned}$$

For the fourth term, we have

$$\begin{aligned}
C_4 = {} & a_4^4 + 4a_1a_4^3 - 4a_2a_4^3 + 4a_3a_4^3 - 12a_1a_2a_4^2 + 12a_1a_3a_4^2 - 12a_2a_3a_4^2 - 24a_1a_2a_3a_4 \\
& + 12a_1a_4a_2^2 + 12a_1a_4a_3^2 - 12a_2a_4a_1^2 - 12a_2a_4a_3^2 + 12a_3a_4a_1^2 + 12a_3a_4a_2^2
\end{aligned}$$

$$+ 4a_4a_1^3 - 4a_4a_2^3 + 4a_4a_3^3 + 6(a_1a_2)^2 + 6(a_1a_3)^2 + 6(a_1a_4)^2 + 6(a_2a_3)^2 + 6(a_2a_4)^2 + 6(a_3a_4)^2$$
$$- 12a_1a_2a_3^2 + 12a_1a_3a_2^2 - 12a_2a_3a_1^2 - 4a_1a_2^3 + 4a_1a_3^3 - 4a_2a_1^3 - 4a_2a_3^3 + 4a_3a_1^3 - 4a_3a_2^3$$
$$+ a_1^4 + a_2^4 + a_3^4.$$

For the fifth term, we have

$$C_5 = a_4^4 - 4a_1a_4^3 + 4a_2a_4^3 + 4a_3a_4^3 - 12a_1a_2a_4^2 - 12a_1a_3a_4^2 + 12a_2a_3a_4^2 - 24a_1a_2a_3a_4$$
$$- 12a_1a_4a_2^2 - 12a_1a_4a_3^2 + 12a_2a_4a_1^2 + 12a_2a_4a_3^2 + 12a_3a_4a_1^2 + 12a_3a_4a_2^2$$
$$- 4a_4a_1^3 + 4a_4a_2^3 + 4a_4a_3^3 + 6(a_1a_2)^2 + 6(a_1a_3)^2 + 6(a_1a_4)^2 + 6(a_2a_3)^2 + 6(a_2a_4)^2 + 6(a_3a_4)^2$$
$$- 12a_1a_2a_3^2 - 12a_1a_3a_2^2 + 12a_2a_3a_1^2 - 4a_1a_2^3 - 4a_1a_3^3 - 4a_2a_1^3 + 4a_2a_3^3 - 4a_3a_1^3 + 4a_3a_2^3$$
$$+ a_1^4 + a_2^4 + a_3^4.$$

For the sixth term, we have

$$C_6 = a_4^4 - 4a_1a_4^3 - 4a_2a_4^3 + 4a_3a_4^3 + 12a_1a_2a_4^2 - 12a_1a_3a_4^2 - 12a_2a_3a_4^2 + 24a_1a_2a_3a_4$$
$$- 12a_1a_4a_2^2 - 12a_1a_4a_3^2 - 12a_2a_4a_1^2 - 12a_2a_4a_3^2 + 12a_3a_4a_1^2 + 12a_3a_4a_2^2$$
$$- 4a_4a_1^3 - 4a_4a_2^3 + 4a_4a_3^3 + 6(a_1a_2)^2 + 6(a_1a_3)^2 + 6(a_1a_4)^2 + 6(a_2a_3)^2 + 6(a_2a_4)^2 + 6(a_3a_4)^2$$
$$+ 12a_1a_2a_3^2 - 12a_1a_3a_2^2 - 12a_2a_3a_1^2 + 4a_1a_2^3 - 4a_1a_3^3 + 4a_2a_1^3 - 4a_2a_3^3 - 4a_3a_1^3 - 4a_3a_2^3$$
$$+ a_1^4 + a_2^4 + a_3^4.$$

For the seventh term, we have

$$C_7 = a_4^4 - 4a_1a_4^3 + 4a_2a_4^3 - 4a_3a_4^3 - 12a_1a_2a_4^2 + 12a_1a_3a_4^2 - 12a_2a_3a_4^2 + 24a_1a_2a_3a_4$$
$$- 12a_1a_4a_2^2 - 12a_1a_4a_3^2 + 12a_2a_4a_1^2 + 12a_2a_4a_3^2 - 12a_3a_4a_1^2 - 12a_3a_4a_2^2$$
$$- 4a_4a_1^3 + 4a_4a_2^3 - 4a_4a_3^3 + 6(a_1a_2)^2 + 6(a_1a_3)^2 + 6(a_1a_4)^2 + 6(a_2a_3)^2 + 6(a_2a_4)^2 + 6(a_3a_4)^2$$
$$- 12a_1a_2a_3^2 + 12a_1a_3a_2^2 - 12a_2a_3a_1^2 - 4a_1a_2^3 + 4a_1a_3^3 - 4a_2a_1^3 - 4a_2a_3^3 + 4a_3a_1^3 - 4a_3a_2^3$$
$$+ a_1^4 + a_2^4 + a_3^4.$$

For the eighth term, we have

$$C_8 = a_4^4 + 4a_1a_4^3 - 4a_2a_4^3 - 4a_3a_4^3 - 12a_1a_2a_4^2 - 12a_1a_3a_4^2 + 12a_2a_3a_4^2 + 24a_1a_2a_3a_4$$
$$+ 12a_1a_4a_2^2 + 12a_1a_4a_3^2 - 12a_2a_4a_1^2 - 12a_2a_4a_3^2 - 12a_3a_4a_1^2 - 12a_3a_4a_2^2$$
$$+ 4a_4a_1^3 - 4a_4a_2^3 - 4a_4a_3^3 + 6(a_1a_2)^2 + 6(a_1a_3)^2 + 6(a_1a_4)^2 + 6(a_2a_3)^2 + 6(a_2a_4)^2 + 6(a_3a_4)^2$$
$$- 12a_1a_2a_3^2 - 12a_1a_3a_2^2 + 12a_2a_3a_1^2 - 4a_1a_2^3 - 4a_1a_3^3 - 4a_2a_1^3 + 4a_2a_3^3 - 4a_3a_1^3 + 4a_3a_2^3$$
$$+ a_1^4 + a_2^4 + a_3^4.$$

For the ninth term, we have

$$C_9 = a_4^4 - 4a_1a_4^3 - 4a_2a_4^3 + 4a_3a_4^3 + 12a_1a_2a_4^2 - 12a_1a_3a_4^2 - 12a_2a_3a_4^2 + 24a_1a_2a_3a_4$$
$$- 12a_1a_4a_2^2 - 12a_1a_4a_3^2 - 12a_2a_4a_1^2 - 12a_2a_4a_3^2 + 12a_3a_4a_1^2 + 12a_3a_4a_2^2$$
$$- 4a_4a_1^3 - 4a_4a_2^3 + 4a_4a_3^3 + 6(a_1a_2)^2 + 6(a_1a_3)^2 + 6(a_1a_4)^2 + 6(a_2a_3)^2 + 6(a_2a_4)^2 + 6(a_3a_4)^2$$
$$+ 12a_1a_2a_3^2 - 12a_1a_3a_2^2 - 12a_2a_3a_1^2 + 4a_1a_2^3 - 4a_1a_3^3 + 4a_2a_1^3 - 4a_2a_3^3 - 4a_3a_1^3 - 4a_3a_2^3$$
$$+ a_1^4 + a_2^4 + a_3^4.$$

For the tenth term, we have

$$C_{10} = a_4^4 - 4a_1a_4^3 + 4a_2a_4^3 - 4a_3a_4^3 - 12a_1a_2a_4^2 + 12a_1a_3a_4^2 - 12a_2a_3a_4^2 + 24a_1a_2a_3a_4$$
$$- 12a_1a_4a_2^2 - 12a_1a_4a_3^2 + 12a_2a_4a_1^2 + 12a_2a_4a_3^2 - 12a_3a_4a_1^2 - 12a_3a_4a_2^2$$
$$+ 4a_4a_1^3 - 4a_4a_2^3 + 4a_4a_3^3 + 6(a_1a_2)^2 + 6(a_1a_3)^2 + 6(a_1a_4)^2 + 6(a_2a_3)^2 + 6(a_2a_4)^2 + 6(a_3a_4)^2$$
$$- 12a_1a_2a_3^2 + 12a_1a_3a_2^2 - 12a_2a_3a_1^2 - 4a_1a_2^3 + 4a_1a_3^3 - 4a_2a_1^3 - 4a_2a_3^3 + 4a_3a_1^3 - 4a_3a_2^3$$

$$+ a_1^4 + a_2^4 + a_3^4.$$

For the eleventh term, we have

$$\begin{aligned}C_{11} = {} & a_4^4 + 4a_1a_4^3 - 4a_2a_4^3 - 4a_3a_4^3 - 12a_1a_2a_4^2 - 12a_1a_3a_4^2 + 12a_2a_3a_4^2 + 24a_1a_2a_3a_4 \\ & + 12a_1a_4a_2^2 + 12a_1a_4a_3^2 - 12a_2a_4a_1^2 - 12a_2a_4a_3^2 - 12a_3a_4a_1^2 - 12a_3a_4a_2^2 \\ & + 4a_4a_1^3 - 4a_4a_2^3 - 4a_4a_3^3 + 6(a_1a_2)^2 + 6(a_1a_3)^2 + 6(a_1a_4)^2 + 6(a_2a_3)^2 + 6(a_2a_4)^2 + 6(a_3a_4)^2 \\ & - 12a_1a_2a_3^2 - 12a_1a_3a_2^2 + 12a_2a_3a_1^2 - 4a_1a_2^3 - 4a_1a_3^3 - 4a_2a_1^3 + 4a_2a_3^3 - 4a_3a_1^3 + 4a_3a_2^3 \\ & + a_1^4 + a_2^4 + a_3^4.\end{aligned}$$

For the twelfth term, we have

$$\begin{aligned}C_{12} = {} & a_4^4 - 4a_1a_4^3 - 4a_2a_4^3 - 4a_3a_4^3 + 12a_1a_2a_4^2 + 12a_1a_3a_4^2 + 12a_2a_3a_4^2 - 24a_1a_2a_3a_4 \\ & - 12a_1a_4a_2^2 - 12a_1a_4a_3^2 - 12a_2a_4a_1^2 - 12a_2a_4a_3^2 - 12a_3a_4a_1^2 - 12a_3a_4a_2^2 \\ & - 4a_4a_1^3 - 4a_4a_2^3 - 4a_4a_3^3 + 6(a_1a_2)^2 + 6(a_1a_3)^2 + 6(a_1a_4)^2 + 6(a_2a_3)^2 + 6(a_2a_4)^2 + 6(a_3a_4)^2 \\ & + 12a_1a_2a_3^2 + 12a_1a_3a_2^2 + 12a_2a_3a_1^2 + 4a_1a_2^3 + 4a_1a_3^3 + 4a_2a_1^3 + 4a_2a_3^3 + 4a_3a_1^3 + 4a_3a_2^3 \\ & + a_1^4 + a_2^4 + a_3^4.\end{aligned}$$

For the thirteenth term, we have

$$\begin{aligned}C_{13} = {} & a_4^4 + 4a_1a_4^3 + 4a_2a_4^3 - 4a_3a_4^3 + 12a_1a_2a_4^2 - 12a_1a_3a_4^2 - 12a_2a_3a_4^2 - 24a_1a_2a_3a_4 \\ & + 12a_1a_4a_2^2 + 12a_1a_4a_3^2 + 12a_2a_4a_1^2 + 12a_2a_4a_3^2 - 12a_3a_4a_1^2 - 12a_3a_4a_2^2 \\ & + 4a_4a_1^3 + 4a_4a_2^3 - 4a_4a_3^3 + 6(a_1a_2)^2 + 6(a_1a_3)^2 + 6(a_1a_4)^2 + 6(a_2a_3)^2 + 6(a_2a_4)^2 + 6(a_3a_4)^2 \\ & + 12a_1a_2a_3^2 - 12a_1a_3a_2^2 - 12a_2a_3a_1^2 + 4a_1a_2^3 - 4a_1a_3^3 + 4a_2a_1^3 - 4a_2a_3^3 - 4a_3a_1^3 - 4a_3a_2^3 \\ & + a_1^4 + a_2^4 + a_3^4.\end{aligned}$$

For the fourteenth term, we have

$$\begin{aligned}C_{14} = {} & a_4^4 + 4a_1a_4^3 - 4a_2a_4^3 + 4a_3a_4^3 - 12a_1a_2a_4^2 + 12a_1a_3a_4^2 - 12a_2a_3a_4^2 - 24a_1a_2a_3a_4 \\ & + 12a_1a_4a_2^2 + 12a_1a_4a_3^2 - 12a_2a_4a_1^2 - 12a_2a_4a_3^2 + 12a_3a_4a_1^2 + 12a_3a_4a_2^2 \\ & + 4a_4a_1^3 - 4a_4a_2^3 + 4a_4a_3^3 + 6(a_1a_2)^2 + 6(a_1a_3)^2 + 6(a_1a_4)^2 + 6(a_2a_3)^2 + 6(a_2a_4)^2 + 6(a_3a_4)^2 \\ & - 12a_1a_2a_3^2 + 12a_1a_3a_2^2 - 12a_2a_3a_1^2 - 4a_1a_2^3 + 4a_1a_3^3 - 4a_2a_1^3 - 4a_2a_3^3 + 4a_3a_1^3 - 4a_3a_2^3 \\ & + a_1^4 + a_2^4 + a_3^4.\end{aligned}$$

For the fifteenth term, we have

$$\begin{aligned}C_{15} = {} & a_4^4 - 4a_1a_4^3 + 4a_2a_4^3 + 4a_3a_4^3 - 12a_1a_2a_4^2 - 12a_1a_3a_4^2 + 12a_2a_3a_4^2 - 24a_1a_2a_3a_4 \\ & - 12a_1a_4a_2^2 - 12a_1a_4a_3^2 + 12a_2a_4a_1^2 + 12a_2a_4a_3^2 + 12a_3a_4a_1^2 + 12a_3a_4a_2^2 \\ & - 4a_4a_1^3 + 4a_4a_2^3 + 4a_4a_3^3 + 6(a_1a_2)^2 + 6(a_1a_3)^2 + 6(a_1a_4)^2 + 6(a_2a_3)^2 + 6(a_2a_4)^2 + 6(a_3a_4)^2 \\ & - 12a_1a_2a_3^2 - 12a_1a_3a_2^2 + 12a_2a_3a_1^2 - 4a_1a_2^3 - 4a_1a_3^3 - 4a_2a_1^3 + 4a_2a_3^3 - 4a_3a_1^3 + 4a_3a_2^3 \\ & + a_1^4 + a_2^4 + a_3^4.\end{aligned}$$

For the sixteenth term, we have

$$\begin{aligned}C_{16} = {} & a_4^4 + 4a_1a_4^3 + 4a_2a_4^3 + 4a_3a_4^3 + 12a_1a_2a_4^2 + 12a_1a_3a_4^2 + 12a_2a_3a_4^2 + 24a_1a_2a_3a_4 \\ & + 12a_1a_4a_2^2 + 12a_1a_4a_3^2 + 12a_2a_4a_1^2 + 12a_2a_4a_3^2 + 12a_3a_4a_1^2 + 12a_3a_4a_2^2 \\ & + 4a_4a_1^3 + 4a_4a_2^3 + 4a_4a_3^3 + 6(a_1a_2)^2 + 6(a_1a_3)^2 + 6(a_1a_4)^2 + 6(a_2a_3)^2 + 6(a_2a_4)^2 + 6(a_3a_4)^2 \\ & + 12a_1a_2a_3^2 + 12a_1a_3a_2^2 + 12a_2a_3a_1^2 + 4a_1a_2^3 + 4a_1a_3^3 + 4a_2a_1^3 + 4a_2a_3^3 + 4a_3a_1^3 + 4a_3a_2^3 \\ & + a_1^4 + a_2^4 + a_3^4.\end{aligned}$$

Putting things together, we have

$$
\begin{aligned}
&(a_1+a_2+a_3+a_4)^4 \\
&-(a_1+a_2+a_3-a_4)^4-(a_1+a_2-a_3+a_4)^4-(a_1-a_2+a_3+a_4)^4-(-a_1+a_2+a_3+a_4)^4 \\
&+(a_1+a_2-a_3-a_4)^4+(a_1-a_2+a_3-a_4)^4+(a_1-a_2-a_3+a_4)^4 \\
&+(-a_1-a_2+a_3+a_4)^4+(-a_1+a_2-a_3+a_4)^4+(-a_1+a_2+a_3-a_4)^4 \\
&-(-a_1-a_2-a_3+a_4)^4-(-a_1-a_2+a_3-a_4)^4-(-a_1+a_2-a_3-a_4)^4-(a_1-a_2-a_3-a_4)^4 \\
&+(-a_1-a_2-a_3-a_4)^4 \\
=&\, C_1-C_2-C_3-C_4-C_5+C_6+C_7+C_8+C_9+C_{10}+C_{11}-C_{12}-C_{13}-C_{14}-C_{15}+C_{16} \\
=&\, 384a_1a_2a_3 \\
=&\, 3\cdot 2^7 a_1a_2a_3a_4
\end{aligned}
$$

Proof of Part 4.

$$2^k\cdot k!\cdot\prod_{i=1}^k a_i=\sum_{c\in\{-1,+1\}^k}(-1)^{(k-\sum_{i=1}^k c_i)/2}(\sum_{j=1}^k c_j a_j)^k.$$

We first let $a_1=0$. Then each term on RHS can find a corresponding negative copy of this term. In detail, let c_1 change sign and we have, $(-1)^{(k-c_1-\sum_{i=2}^k c_i)/2}(c_1\cdot 0+\sum_{j=2}^k c_j a_j)^k = -(-1)^{(k+c_1-\sum_{i=2}^k c_i)/2}(-c_1\cdot 0+\sum_{j=2}^k c_j a_j)^k$. We can find this mapping is always one-to-one and onto mapping with each other. Thus, we have RHS is constant 0 regardless of $a_2,\dots,a_k$. Thus, a_1 is a factor of RHS. By symmetry, $a_2,\dots,a_k$ also are factors of RHS. Since RHS is k-th order, we have RHS$=\alpha\prod_{i=1}^k a_i$ where α is a constant. Take $a_1=\cdots=a_k=1$, we have $\alpha=2^k\cdot k!$ =RHS. Thus, we finish the proof. □

I.2 Constructions for θ^*

Lemma I.2. *When $k=3$, provided the following conditions are met*

- *We denote $\mathcal{B}$ as the ball that $\|u_1\|^2+\|u_2\|^2+\|u_3\|^2+\|w\|^2\le 1$.*
- *We define $\Omega_q^{'*}$ in Definition H.2.*
- *We adopt the uniform class weighting: $\forall c'\neq a_1+a_2+a_3,\ \ \tau(a_1,a_2,a_3)[c']:=1/(p-1)$.*
- *Let $\cos_\zeta(x)$ denote $\cos(2\pi\zeta x/p)$*
- *Let $\sin_\zeta(x)$ denote $\sin(2\pi\zeta x/p)$*

Then, we have

- *The maximum $L_{2,4}$-margin solution θ^* will consist of $16(p-1)$ neurons $\theta_i^*\in\Omega_q^{'*}$ to simulate $\frac{p-1}{2}$ type of cosine computation, each cosine computation is uniquely determined a $\zeta\in\{1,\dots,\frac{p-1}{2}\}$. In particular, for each ζ the cosine computation is $\cos_\zeta(a_1+a_2+a_3-c),\forall a_1,a_2,a_3,c\in\mathbb{Z}_p$.*

Proof. Referencing Lemma H.4, we can identify elements within $\Omega_q^{'}$. Our set θ^* will be composed of $16(p-1)$ neurons, including 32 neurons dedicated to each frequency in the range $1,\dots,\frac{p-1}{2}$. Focusing on a specific frequency ζ, for the sake of simplicity, let us use $\cos_\zeta(x)$ to represent $\cos(2\pi\zeta x/p)$ and $\sin_\zeta(x)$ likewise. We note:

$$\cos_\zeta(a_1+a_2+a_3-c)=\cos_\zeta(a_1+a_2+a_3)\cos_\zeta(c)+\sin_\zeta(a_1+a_2+a_3)\sin_\zeta(c)$$

$$
\begin{aligned}
&= \cos_\zeta(a_1+a_2)\cos_\zeta(a_3)\cos_\zeta(c) - \sin_\zeta(a_1+a_2)\sin_\zeta(a_3)\cos_\zeta(c) \\
&\quad + \sin_\zeta(a_1+a_2)\cos_\zeta(a_3)\sin_\zeta(c) + \cos_\zeta(a_1+a_2)\sin_\zeta(a_3)\sin_\zeta(c) \\
&= (\cos_\zeta(a_1)\cos_\zeta(a_2) - \sin_\zeta(a_1)\sin_\zeta(a_2))\cos_\zeta(a_3)\cos_\zeta(c) \\
&\quad - (\sin_\zeta(a_1)\cos_\zeta(a_2) + \cos_\zeta(a_1)\sin_\zeta(a_2))\sin_\zeta(a_3)\cos_\zeta(c) \\
&\quad + (\sin_\zeta(a_1)\cos_\zeta(a_2) + \cos_\zeta(a_1)\sin_\zeta(a_2))\cos_\zeta(a_3)\sin_\zeta(c) \\
&\quad + ((\cos_\zeta(a_1)\cos_\zeta(a_2) - \sin_\zeta(a_1)\sin_\zeta(a_2)))\sin_\zeta(a_3)\sin_\zeta(c) \\
&= \cos_\zeta(a_1)\cos_\zeta(a_2)\cos_\zeta(a_3)\cos_\zeta(c) - \sin_\zeta(a_1)\sin_\zeta(a_2)\cos_\zeta(a_3)\cos_\zeta(c) \\
&\quad - \sin_\zeta(a_1)\cos_\zeta(a_2)\sin_\zeta(a_3)\cos_\zeta(c) - \cos_\zeta(a_1)\sin_\zeta(a_2)\sin_\zeta(a_3)\cos_\zeta(c) \\
&\quad + \sin_\zeta(a_1)\cos_\zeta(a_2)\cos_\zeta(a_3)\sin_\zeta(c) + \cos_\zeta(a_1)\sin_\zeta(a_2)\cos_\zeta(a_3)\sin_\zeta(c) \\
&\quad + \cos_\zeta(a_1)\cos_\zeta(a_2)\sin_\zeta(a_3)\sin_\zeta(c) - \sin_\zeta(a_1)\sin_\zeta(a_2)\sin_\zeta(a_3)\sin_\zeta(c)
\end{aligned} \tag{24}
$$

where all steps comes from trigonometric function.

Each of these 8 terms can be implemented by 4 neurons $\phi_1, \phi_2, \cdots, \phi_4$. Consider the first term, $\cos_\zeta(a_1)\cos_\zeta(a_2)\cos_\zeta(a_3)\cos_\zeta(c)$.

For the i-th neuron, we have

$$\phi_i = (u_{i,1}(a_1) + u_{i,2}(a_2) + u_{i,3}(a_3))^3 \cdot w_i(c).$$

By changing $(\theta_{i,j})^*$, we can change the constant factor of $\cos_\zeta(\cdot)$ to be $+\beta$ or $-\beta$. Hence, we can view $u_{i,j}(\cdot), w_i(\cdot)$ as the following:

$$
\begin{aligned}
u_{i,1}(\cdot) &:= p_{i,1} \cdot \cos_\zeta(\cdot), \\
u_{i,2}(\cdot) &:= p_{i,2} \cdot \cos_\zeta(\cdot), \\
u_{i,3}(\cdot) &:= p_{i,3} \cdot \cos_\zeta(\cdot), \\
w_i(\cdot) &:= p_{i,4} \cdot \cos_\zeta(\cdot)
\end{aligned}
$$

where $p_{i,j} \in \{-1, 1\}$.

For simplicity, let d_i denote $\cos_\zeta(a_i)$.

We set $(\theta^*_{u_1}, \theta^*_{u_2}, \theta^*_{u_3}, \theta^*_w) = (0, 0, 0, 0)$, then

$$p_{1,1}, p_{1,2}, p_{1,3}, p_{1,4} = 1,$$

then we have

$$\phi_1 = (d_1 + d_2 + d_3)^3 \cos_\zeta(c).$$

We set $(\theta^*_{u_1}, \theta^*_{u_2}, \theta^*_{u_3}, \theta^*_w) = (0, 0, \pi, \pi)$, then $p_{2,1}, p_{2,2} = 1$ and $p_{2,3}, p_{2,4} = -1$, then we have

$$\phi_2 = -(d_1 + d_2 - d_3)^3 \cos_\zeta(c).$$

We set $(\theta^*_{u_1}, \theta^*_{u_2}, \theta^*_{u_3}, \theta^*_w) = (0, \pi, 0, \pi)$, then $p_{3,1}, p_{3,3} = 1$ and $p_{3,2}, p_{3,4} = -1$, then we have

$$\phi_3 = -(d_1 - d_2 + d_3)^3 \cos_\zeta(c).$$

We set $(\theta^*_{u_1}, \theta^*_{u_2}, \theta^*_{u_3}, \theta^*_w) = (\pi, 0, 0, \pi)$, then $p_{4,1}, p_{4,4} = -1$ and $p_{2,2}, p_{2,3} = 1$, then we have

$$\phi_4 = -(-d_1 + d_2 + d_3)^3 \cos_\zeta(c).$$

Putting them together, we have

$$
\begin{aligned}
&\sum_{i=1}^4 \phi_i(a_1, a_2, a_3) \\
&= \sum_{i=1}^4 (u_{i,1}(a_1) + u_{i,2}(a_2) + u_{i,3}(a_3))^3 w_i(c)
\end{aligned}
$$

$$
\begin{aligned}
&= \sum_{i=1}^{4} (p_{i,1} \cos_\zeta(a_1) + p_{i,2} \cos_\zeta(a_2) + p_{i,3} \cos_\zeta(a_3))^3 w_i(c) \\
&= [(d_1 + d_2 + d_3)^3 - (d_1 + d_2 - d_3)^3 - (d_1 - d_2 + d_3)^3 - (-d_1 + d_2 + d_3)^3] \cos_\zeta(c) \\
&= 24 d_1 d_2 d_3 \cos_\zeta(c) \\
&= 24 \cos_\zeta(a_1) \cos_\zeta(a_2) \cos_\zeta(a_3) \cos_\zeta(c)
\end{aligned} \tag{25}
$$

where the first step comes from the definition of ϕ_i, the second step comes from the definition of $u_{i,j}$, the third step comes from $d_i = \cos_\zeta(a_i)$, the fourth step comes from simple algebra, the last step comes from $d_i = \cos_\zeta(a_i)$.

Similarly, consider $-\sin_\zeta(a_1) \sin_\zeta(a_2) \cos_\zeta(a_3) \cos_\zeta(c)$.

We set $(\theta^*_{u_1}, \theta^*_{u_2}, \theta^*_{u_3}, \theta^*_w) = (\pi/2, \pi/2, 0, \pi)$, then we have

$$\phi_1 = -(\sin_\zeta(a_1) + \sin_\zeta(a_2) + \cos_\zeta(a_3))^3 \cos_\zeta(c).$$

We set $(\theta^*_{u_1}, \theta^*_{u_2}, \theta^*_{u_3}, \theta^*_w) = (\pi/2, \pi/2, -\pi, 0)$, then we have

$$\phi_2 = (\sin_\zeta(a_1) + \sin_\zeta(a_2) - \cos_\zeta(a_3))^3 \cos_\zeta(c).$$

We set $(\theta^*_{u_1}, \theta^*_{u_2}, \theta^*_{u_3}, \theta^*_w) = (\pi/2, -\pi/2, 0, 0)$, then we have

$$\phi_3 = (\sin_\zeta(a_1) - \sin_\zeta(a_2) + \cos_\zeta(a_3))^3 \cos_\zeta(c).$$

We set $(\theta^*_{u_1}, \theta^*_{u_2}, \theta^*_{u_3}, \theta^*_w) = (-\pi/2, \pi/2, 0, 0)$, then we have

$$\phi_4 = (-\sin_\zeta(a_1) + \sin_\zeta(a_2) + \cos_\zeta(a_3))^3 \cos_\zeta(c).$$

Putting them together, we have

$$
\begin{aligned}
&\sum_{i=1}^{4} \phi_i(a_1, a_2, a_3) \\
=& -24 \sin_\zeta(a_1) \sin_\zeta(a_2) \cos_\zeta(a_3) \cos_\zeta(c)
\end{aligned} \tag{26}
$$

Similarly, all other six terms in Eq. equation 24 can be composed by four neurons with different $(\theta^*_{u_1}, \theta^*_{u_2}, \theta^*_{u_3}, \theta^*_w)$.

When we include such 56 neurons for all frequencies $\zeta \in \{1, \ldots, \frac{p-1}{2}\}$, we have that the network will calculate the following function

$$
\begin{aligned}
f(a_1, a_2, a_3, c) &= \sum_{\zeta=1}^{(p-1)/2} \cos_\zeta(a_1 + a_2 + a_3 - c) \\
&= \sum_{\zeta=1}^{p-1} \frac{1}{2} \cdot \exp(2\pi \mathbf{i} \zeta (a_1 + a_2 + a_3 - c)/p) \\
&= \begin{cases} \frac{p-1}{2} & \text{if } a_1 + a_2 + a_3 = c \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
$$

where the first step comes from the definition of $f(a_1, a_2, a_3, c)$, the second step comes from Euler's formula, the last step comes from the properties of discrete Fourier transform.

The scaling factor β for each neuron can be selected such that the entire network maintains an $L_{2,4}$-norm of 1. In this setup, every data point lies exactly on the margin, meaning $q = \text{unif}(\mathbb{Z}_p)$ uniformly covers points on the margin, thus meeting the criteria for q^* as outlined in Definition C.7. Furthermore, for any input (a_1, a_2, a_3), the function f yields an identical result across all incorrect labels c', adhering to Condition C.10. □

I.3 CONSTRUCTIONS FOR θ^* FOR GENERAL k VERSION

Lemma I.3 (Formal version of Lemma F.2)**.** *Provided the following conditions are met*

- *We denote $\mathcal{B}$ as the ball that $\|u_1\|^2 + \cdots + \|u_k\|^2 + \|w\|^2 \leq 1$.*
- *We define $\Omega_q^{'*}$ in Definition H.2.*
- *We adopt the uniform class weighting: $\forall c' \neq a_1 + \cdots + a_k,\ \ \tau(a_1, \ldots, a_k)[c'] := 1/(p-1)$.*
- *Let $\cos_\zeta(x)$ denote $\cos(2\pi\zeta x/p)$*
- *Let $\sin_\zeta(x)$ denote $\sin(2\pi\zeta x/p)$*

Then, we have

- *The maximum $L_{2,k+1}$-margin solution θ^* will consist of $2^{2k-1} \cdot \frac{p-1}{2}$ neurons $\theta_i^* \in \Omega_q^{'*}$ to simulate $\frac{p-1}{2}$ type of cosine computation, each cosine computation is uniquely determined a $\zeta \in \{1, \ldots, \frac{p-1}{2}\}$. In particular, for each ζ the cosine computation is $\cos_\zeta(a_1 + \cdots + a_k - c), \forall a_1, \ldots, a_k, c \in \mathbb{Z}_p$.*

Proof. By Lemma H.4, we can get elements of $\Omega_q^{'*}$. Our set θ^* will be composed of $2^{2k-1} \cdot \frac{p-1}{2}$ neurons, including 2^{2k-1} neurons dedicated to each frequency in the range $1, \ldots, \frac{p-1}{2}$. Focusing on a specific frequency ζ, for the sake of simplicity, let us use $\cos_\zeta(x)$ to represent $\cos(2\pi\zeta x/p)$ and $\sin_\zeta(x)$ likewise.

We define

$$a_{[k]} := \sum_{i=1}^{k} a_k$$

and we also define

$$a_{k+1} := -c.$$

For easy of writing, we will write $\cos_\zeta$ as $\cos$ and $\sin_\zeta$ as $\sin$. We have the following.

$$\begin{aligned}
& \cos_\zeta(\sum_{i=1}^{k} a_i - c) \\
= \ & \cos(\sum_{i=1}^{k} a_i - c) \\
= \ & \cos(a_{[k+1]}) \\
= \ & \cos(a_{[k]})\cos(a_{k+1}) - \sin(a_{[k]})\sin(a_{k+1}) \\
= \ & \cos(a_{[k-1]} + a_k)\cos(a_{k+1}) - \sin(a_{[k-1]} + a_k)\sin(a_{k+1}) \\
= \ & \cos(a_{[k-1]})\cos(a_k)\cos(a_{k+1}) - \sin(a_{[k-1]})\sin(a_k)\cos(a_{k+1}) \\
& - \sin(a_{[k-1]})\cos(a_k)\sin(a_{k+1}) - \cos(a_{[k-1]})\sin(a_k)\sin(a_{k+1}) \\
= \ & \sum_{b \in \{0,1\}^{k+1}} \prod_{i=1}^{k+1} \cos^{1-b_i}(a_i) \cdot \sin^{b_i}(a_i) \cdot \mathbf{1}[\sum_{i=1}^{k+1} b_i \% 2 = 0] \cdot (-1)^{\mathbf{1}[\sum_{i=1}^{k+1} b_i \% 4 = 2]}, \qquad (27)
\end{aligned}$$

where the first step comes from the simplicity of writing, the second step comes from the definition of $a_{[k+1]}$ and a_{k+1}, the third step comes from the trigonometric function, the fourth step also follows trigonometric function, and the last step comes from the below two observations:

- First, we observe that $\cos(a+b) = \cos(a)\cos(b) - \sin(a)\sin(b)$ and $\sin(a+b) = \sin(a)\cos(b) + \cos(a)\sin(b)$. When we split $\cos$ once, we will remove one $\cos$ product and we may add zero or two $\sin$ products. When we split $\sin$ once, we may remove one

$\sin$ product and we will add one $\sin$ product as well. Thus, we can observe that the number of $\sin$ products in each term is always even.

- Second, we observe only when we split $\cos$ and add two $\sin$ products will introduce a -1 is this term. Thus, when the number of $\sin$ products $\%4 = 2$, the sign of this term will be -1. Otherwise, it will be $+1$.

Note that we have 2^k non-zero term in Eq. equation 27. Each of these 2^k terms can be implemented by 2^{k-1} neurons $\phi_1, \cdots, \phi_{2^{k-1}}$.

For the i-th neuron, we have

$$\phi_i = (\sum_{j=1}^{k} u_{i,j}(a_j))^k \cdot w_i(c).$$

By changing $(\theta_{i,j})^*$, we can change the $u_{i,j}(a_j)$ from $\cos_\zeta(\cdot)$ to be $-\cos_\zeta(\cdot)$ or $\sin_\zeta(\cdot)$ or $-\sin_\zeta(\cdot)$. Denote $\theta^*_{u_i}$ as $(\theta_{i,a_i})^*$.

For simplicity, let d_i denote the i-th product in one term of Eq. equation 27. By fact that

$$2^k \cdot k! \cdot \prod_{i=1}^{k} d_i = \sum_{c \in \{-1,+1\}^k} (-1)^{(k-\sum_{i=1}^k c_i)/2} (\sum_{j=1}^{k} c_j d_j)^k,$$

each term can be constructed by 2^{k-1} neurons (note that there is a symmetric effect so we only need half terms). Based on Eq. equation 27 and the above fact with carefully check, we can see that $\theta^*_{u_1} + \cdots + \theta^*_{u_k} = \theta^*_w$. Thus, we need $2^k \cdot 2^{k-1} \cdot \frac{p-1}{2}$ neurons in total.

When we include such $2^k \cdot 2^{k-1}$ neurons for all frequencies $\zeta \in \{1, \ldots, \frac{p-1}{2}\}$, we have the network will calculate the following function

$$\begin{aligned} f(a_1, \ldots, a_k, c) &= \sum_{\zeta=1}^{(p-1)/2} \cos_\zeta(\sum_{i=1}^{k} a_i - c) \\ &= \sum_{\zeta=1}^{p-1} \frac{1}{2} \cdot \exp(2\pi \mathbf{i} \zeta (\sum_{i=1}^{k} a_i - c)/p) \\ &= \begin{cases} \frac{p-1}{2} & \text{if } \sum_{i=1}^k a_i = c \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

where the first step comes from the definition of $f(a_1, \ldots, a_k, c)$, the second step comes from Euler's formula, the last step comes from the properties of discrete Fourier transform.

The scaling parameter β for each neuron can be adjusted to ensure that the network possesses an $L_{2,k+1}$-norm of 1. For this network, all data points are positioned on the margin, which implies that $q = \text{unif}(\mathbb{Z}_p)$ naturally supports points along the margin, aligning with the requirements for q^* presented in Definition C.7. Additionally, for every input $(a_1, \ldots, a_k)$, the function f assigns the same outcome to all incorrect labels c', thereby fulfilling Condition C.10. □

J Check Fourier Frequencies

Section J.1 proves all frequencies are used. Section J.2 proves all frequencies are used for general k version.

J.1 All Frequencies are Used

Let $f : \mathbb{Z}_p^4 \to \mathbb{C}$. Its multi-dimensional discrete Fourier transform is defined as:

$$\begin{aligned}&\widehat{f}(j_1,j_2,j_3,j_4)\\ :=&\sum_{a_1\in\mathbb{Z}_p} e^{-2\pi\mathbf{i}\cdot j_1a_1/p}(\sum_{a_2\in\mathbb{Z}_p} e^{-2\pi\mathbf{i}\cdot j_2a_2/p}(\sum_{a_3\in\mathbb{Z}_p} e^{-2\pi\mathbf{i}\cdot j_3a_3/p}(\sum_{c\in\mathbb{Z}_p} e^{-2\pi\mathbf{i}\cdot j_4c/p}f(a_1,a_2,a_3,c))).\end{aligned}$$

Lemma J.1. *When $k=3$, if the following conditions hold*

- *We adopt the uniform class weighting: $\forall c'\neq a_1+a_2+a_3,\ \ \tau(a_1,a_2,a_3)[c']:=1/(p-1)$.*
- *f is the maximum $L_{2,4}$-margin solution.*

Then, for any $j_1=j_2=j_3=-j_4\neq 0$, we have $\widehat{f}(j_1,j_2,j_3,j_4)>0$.

Proof. In this proof, let $j_1,j_2,j_3,j_4\in\mathbb{Z}$, and $\theta_u=\theta_u^*\cdot\frac{p}{2\pi}$ to simplify the notation. By Lemma H.4,

$$u_1(a_1)=\sqrt{\frac{1}{2p}}\cos_p(\theta_{u_1}+\zeta a_1). \tag{28}$$

Let

$$\begin{aligned}f(a_1,a_2,a_3,c)&=\sum_{h=1}^{H}\phi_h(a_1,a_2,a_3,c)\\ &=\sum_{h=1}^{H}(u_{h,1}(a_1)+u_{h,2}(a_2)+u_{h,3}(a_3))^3w_h(c)\\ &=(\frac{1}{2p})^2\sum_{h=1}^{H}(\cos_p(\theta_{u_{h,1}}+\zeta_h a_1)+\cos_p(\theta_{u_{h,2}}+\zeta_h a_2)+\cos_p(\theta_{u_{h,3}}+\zeta_h a_3))^3\cos_p(\theta_{w_h}+\zeta_h c)\end{aligned}$$

where each neuron conforms to the previously established form, and the width H function is an arbitrary margin-maximizing network. The first step is from the definition of $f(a_1,a_2,a_3,c)$, the subsequent step on the definition of $\phi_h(a_1,a_2,a_3,c)$, and the final step is justified by Eq. equation 28.

We can divide each ϕ into ten terms:

$$\begin{aligned}&\phi(a_1,a_2,a_3,c)\\ =&\phi^{(1)}(a_1,a_2,a_3,c)+\cdots+\phi^{(10)}(a_1,a_2,a_3,c)\\ =&\big(u_1(a_1)^3+u_2(a_2)^3+u_3(a_3)^3+3u_1(a_1)^2u_2(a_2)+3u_1(a_1)^2u_3(a_3)+3u_2(a_2)^2u_1(a_1)\\ &+3u_2(a_2)^2u_3(a_3)+3u_3(a_3)^2u_1(a_1)+3u_3(a_3)^2u_2(a_2)+6u_1(a_1)u_2(a_2)u_3(a_3)\big)w(c).\end{aligned}$$

Note, $\rho=e^{2\pi\mathbf{i}/p}$. $\widehat{\phi}_1(j_1,j_2,j_3,j_4)$ is nonzero only for $j_1=0$, and $\widehat{\phi}_4(j_1,j_2,j_3,j_4)$ is nonzero only for $j_1=j_2=0$. Similar to other terms. For the tenth term, we have

$$\begin{aligned}\widehat{\phi}_{10}(j_1,j_2,j_3,j_4)&=6\sum_{a_1,a_2,a_3,c\in\mathbb{Z}_p}u_1(a_1)u_2(a_2)u_3(a_3)w(c)\rho^{-(j_1a_1+j_2a_2+j_3a_3+j_4c)}\\ &=6\widehat{u}_1(j_1)\widehat{u}_2(j_2)\widehat{u}_3(j_3)\widehat{w}(j_4).\end{aligned}$$

In particular,

$$\begin{aligned}\widehat{u}_1(j_1)&=\sum_{a_1\in\mathbb{Z}_p}\sqrt{\frac{1}{2p}}\cos_p(\theta_{u_1}+\zeta a_1)\rho^{-j_1a_1}\\ &=(8p)^{-1/2}\sum_{a_1\in\mathbb{Z}_p}(\rho^{\theta_{u_1}+\zeta a_1}+\rho^{-(\theta_{u_1}+\zeta a_1)})\rho^{-j_1a_1}\end{aligned}$$

$$= (8p)^{-1/2}(\rho^{\theta_{u_1}} \sum_{a_1 \in \mathbb{Z}_p} \rho^{(\zeta - j_1)a_1} + \rho^{-\theta_{u_1}} \sum_{a_1 \in \mathbb{Z}_p} \rho^{-(\zeta + j_1)a_1})$$
$$= \begin{cases} \sqrt{p/8} \cdot \rho^{\theta_{u_1}} & \text{if } j_1 = +\zeta \\ \sqrt{p/8} \cdot \rho^{-\theta_{u_1}} & \text{if } j_1 = -\zeta \\ 0 & \text{otherwise} \end{cases}$$

where the first step comes from $\widehat{u}_1(j_1)$ definition, the second step comes from Euler's formula, the third step comes from simple algebra, the last step comes from the properties of discrete Fourier transform. Similarly for $\widehat{u}_2, \widehat{u}_3$ and $\widehat{w}$. As we consider ζ to be nonzero, we ignore the $\zeta = 0$ case. Hence, $\widehat{\phi}_{10}(j_1, j_2, j_3, j_4)$ is nonzero only when j_1, j_2, j_3, j_4 are all $\pm\zeta$. We can summarize that $\widehat{\phi}(j_1, j_2, j_3, j_4)$ can only be nonzero if one of the following satisfies:

- $j_1 \cdot j_2 \cdot j_3 = 0$
- $j_1, j_2, j_3, j_4 = \pm\zeta$.

Setting aside the previously discussed points, it's established in Lemma G.2 that the function f maintains a consistent margin for various inputs as well as over different classes, i.e., f can be broken down as

$$f(a_1, a_2, a_3, c) = f_1(a_1, a_2, a_3, c) + f_2(a_1, a_2, a_3, c)$$

where

$$f_1(a_1, a_2, a_3, c) = F(a_1, a_2, a_3)$$

for some $F : \mathbb{Z}_p \times \mathbb{Z}_p \times \mathbb{Z}_p \to \mathbb{R}$, and

$$f_2(a_1, a_2, a_3, c) = \lambda \cdot \mathbf{1}_{a_1 + a_2 + a_3 = c}$$

where $\lambda > 0$ is the margin of f. Then, we have the DFT of f_1 and f_2 are

$$\widehat{f_1}(j_1, j_2, j_3, j_4) = \begin{cases} \widehat{F}(j_1, j_2, j_3) & \text{if } j_4 = 0 \\ 0 & \text{otherwise} \end{cases}$$

and

$$\widehat{f_2}(j_1, j_2, j_3, j_4) = \begin{cases} \lambda p^3 & \text{if } j_1 = j_2 = j_3 = -j_4 \\ 0 & \text{otherwise} \end{cases}.$$

Hence, when $j_1 = j_2 = j_3 = -j_4 \neq 0$, we must have $\widehat{f}(j_1, j_2, j_3, j_4) > 0$. □

J.2 All Frequencies are Used for General k Version

Let $f : \mathbb{Z}_p^{k+1} \to \mathbb{C}$. Its multi-dimensional discrete Fourier transform is defined as:

$$\widehat{f}(j_1, \ldots, j_{k+1})$$
$$:= \sum_{a_1 \in \mathbb{Z}_p} e^{-2\pi \mathbf{i} \cdot j_1 a_1 / p} (\ldots (\sum_{a_k \in \mathbb{Z}_p} e^{-2\pi \mathbf{i} \cdot j_k a_k / p} (\sum_{c \in \mathbb{Z}_p} e^{-2\pi \mathbf{i} \cdot j_{k+1} c / p} f(a_1, \ldots, a_k, c))).$$

Lemma J.2. *If the following conditions hold*

- *We adopt the uniform class weighting:* $\forall c' \neq a_1 + \cdots + a_k,\ \tau(a_1, \ldots, a_k)[c'] := 1/(p-1)$.

- *f is the maximum $L_{2,k+1}$-margin solution.*

Then, for any $j_1 = \cdots = j_k = -j_{k+1} \neq 0$, we have $\widehat{f}(j_1, \ldots, j_{k+1}) > 0$.

Proof. For this proof, for all $j_1, \ldots, j_{k+1} \in \mathbb{Z}$, to simplify the notation, let $\theta_u = \theta_u^* \cdot \frac{p}{2\pi}$, by Lemma H.8, so

$$u_1(a_1) = \sqrt{\frac{2}{(k+1)p}} \cos_p(\theta_{u_1} + \zeta a_1). \tag{29}$$

Let

$$\begin{aligned} f(a_1, \ldots, a_k, c) &= \sum_{h=1}^{H} \phi_h(a_1, \ldots, a_k, c) \\ &= \sum_{h=1}^{H} (u_{h,1}(a_1) + \cdots + u_{h,k}(a_k))^k w_h(c) \\ &= (\frac{2}{(k+1)p})^{(k+1)/2} \sum_{h=1}^{H} (\cos_p(\theta_{u_{h,1}} + \zeta_h a_1) + \cdots + \cos_p(\theta_{u_{h,k}} + \zeta_h a_k))^k \cos_p(\theta_{w_h} + \zeta_h c) \end{aligned}$$

where each neuron conforms to the previously established form, and the width H function is an arbitrary margin-maximizing network. The first step is based on the definition of $f(a_1, \ldots, a_k, c)$, the subsequent step on the definition of $\phi_h(a_1, \ldots, a_k, c)$, and the final step is justified by Eq. equation 29.

Each neuron ϕ we have

$$\begin{aligned} \widehat{\phi}(j_1, \ldots, j_k, j_{k+1}) &= k! \sum_{a_1, \ldots, a_k, c \in \mathbb{Z}_p} w(c) \rho^{-(j_1 a_1 + \cdots + j_k a_k + j_{k+1} c)} \prod_{i=1}^{k} u_i(a_i) \\ &= k! \widehat{w}(j_{k+1}) \prod_{i=1}^{k} \widehat{u}_i(j_i). \end{aligned}$$

In particular,

$$\begin{aligned} \widehat{u}_1(j_1) &= \sum_{a_1 \in \mathbb{Z}_p} \sqrt{\frac{2}{(k+1)p}} \cos_p(\theta_{u_1} + \zeta a_1) \rho^{-j_1 a_1} \\ &= \sqrt{\frac{1}{2(k+1)p}} \sum_{a_1 \in \mathbb{Z}_p} (\rho^{\theta_{u_1} + \zeta a_1} + \rho^{-(\theta_{u_1} + \zeta a_1)}) \rho^{-j_1 a_1} \\ &= \sqrt{\frac{1}{2(k+1)p}} (\rho^{\theta_{u_1}} \sum_{a_1 \in \mathbb{Z}_p} \rho^{(\zeta - j_1) a_1} + \rho^{-\theta_{u_1}} \sum_{a_1 \in \mathbb{Z}_p} \rho^{-(\zeta + j_1) a_1}) \\ &= \begin{cases} \sqrt{\frac{p}{2(k+1)}} \cdot \rho^{\theta_{u_1}} & \text{if } j_1 = +\zeta \\ \sqrt{\frac{p}{2(k+1)}} \cdot \rho^{-\theta_{u_1}} & \text{if } j_1 = -\zeta \\ 0 & \text{otherwise,} \end{cases} \end{aligned}$$

where the first step comes from $\widehat{u}_1(j_1)$ definition, the second step comes from Euler's formula, the third step comes from simple algebra, the last step comes from the properties of discrete Fourier transform. Similarly for $\widehat{u}_i$ and $\widehat{w}$. We consider ζ to be nonzero, so we ignore the $\zeta = 0$ case. Hence, $\widehat{\phi}(j_1, \ldots, j_k, j_{k+1})$ is nonzero only when $j_1, \ldots, j_k, j_{k+1}$ are all $\pm\zeta$. We can summarize that $\widehat{\phi}(j_1, \ldots, j_k, j_{k+1})$ can only be nonzero if one of the below conditions satisfies:

- $\prod_{i=1}^{k} j_i = 0$
- $j_1, \ldots, j_k, j_{k+1} = \pm\zeta$.

Setting aside the previously discussed points, it's established in Lemma G.2 that the function f maintains a consistent margin for various inputs as well as over different classes, i.e., f can be broken down as

$$f(a_1, \ldots, a_k, c) = f_1(a_1, \ldots, a_k, c) + f_2(a_1, \ldots, a_k, c)$$

where

$$f_1(a_1, \ldots, a_k, c) = F(a_1, \ldots, a_k)$$

for some $F : \mathbb{Z}_p^k \to \mathbb{R}$, and

$$f_2(a_1, \ldots, a_k, c) = \lambda \cdot \mathbf{1}_{a_1 + \cdots + a_k = c}$$

where $\lambda > 0$ is the margin of f. Then, we have the DFT of f_1 and f_2 are

$$\widehat{f}_1(j_1, \ldots, j_k, j_{k+1}) = \begin{cases} \widehat{F}(j_1, \ldots, j_k) & \text{if } j_{k+1} = 0 \\ 0 & \text{otherwise} \end{cases}$$

and

$$\widehat{f}_2(j_1, \ldots, j_k, j_{k+1}) = \begin{cases} \lambda p^k & \text{if } j_1 = \cdots = j_k = -j_{k+1} \\ 0 & \text{otherwise} \end{cases}.$$

Hence, when $j_1 = \cdots = j_k = -j_{k+1} \neq 0$, we must have $\widehat{f}(j_1, \ldots, j_k, j_{k+1}) > 0$. □

K Main Result

Section K.1 proves the main result for $k = 3$. Section K.2 proves the general k version of our main result.

K.1 Main result for $k = 3$

Theorem K.1. *When $k = 3$, let $f(\theta, x)$ be the one-hidden layer networks defined in Section C. If the following conditions hold*

- *We adopt the uniform class weighting: $\forall c' \neq a_1 + a_2 + a_3$, $\tau(a_1, a_2, a_3)[c'] := 1/(p-1)$.*
- *$m \geq 16(p-1)$ neurons.*

Then we have the maximum $L_{2,4}$-margin network satisfying:

- *The maximum $L_{2,4}$-margin for a given dataset D_p is:*

$$\gamma^* = \frac{3}{16} \cdot \frac{1}{p(p-1)}.$$

- *For each neuron $\phi(\{u_1, u_2, u_3, w\}; a_1, a_2, a_3)$, there is a constant scalar $\beta \in \mathbb{R}$ and a frequency $\zeta \in \{1, \ldots, \frac{p-1}{2}\}$ satisfying*

$$u_1(a_1) = \beta \cdot \cos(\theta^*_{u_1} + 2\pi\zeta a_1/p)$$

$$u_2(a_2) = \beta \cdot \cos(\theta^*_{u_2} + 2\pi\zeta a_2/p)$$
$$u_3(a_3) = \beta \cdot \cos(\theta^*_{u_3} + 2\pi\zeta a_3/p)$$
$$w(c) = \beta \cdot \cos(\theta^*_{w} + 2\pi\zeta c/p)$$

where $\theta^*_{u_1}, \theta^*_{u_2}, \theta^*_{u_3}, \theta^*_w \in \mathbb{R}$ *are some phase offsets satisfying* $\theta^*_{u_1} + \theta^*_{u_2} + \theta^*_{u_3} = \theta^*_w$.

- *For each frequency* $\zeta \in \{1, \dots, \frac{p-1}{2}\}$*, there exists one neuron using this frequency only.*

Proof. By Lemma H.4, we get the single neuron class-weighted margin solution set $\Omega_q^{'*}$ satisfying Condition C.10 and γ^*.

By Lemma I.2 and Lemma G.1, we can construct network θ^* which uses neurons in $\Omega_q^{'*}$ and satisfies Condition C.10 and Definition C.7 with respect to $q = \mathrm{unif}(\mathbb{Z}_p)$. By Lemma G.2, we know it is the maximum-margin solution.

By Lemma J.1, when $j_1 = j_2 = j_3 = -j_4 \neq 0$, we must have $\widehat{f}(j_1, j_2, j_3, j_4) > 0$. However, as discrete Fourier transform $\widehat{\phi}$ of each neuron is nonzero, for each frequency, we must have that there exists one neuron using it.

□

K.2 Main Result for General k Version

Theorem K.2 (Formal version of Theorem D.1)**.** *Let* $f(\theta, x)$ *be the one-hidden layer networks defined in Section C. If the following conditions hold*

- *We adopt the uniform class weighting:* $\forall c' \neq a_1 + \cdots + a_k,\ \ \tau(a_1, \dots, a_k)[c'] := 1/(p-1)$.
- $m \geq 2^{2k-1} \cdot \frac{p-1}{2}$ *neurons.*

Then we have the maximum $L_{2,k+1}$*-margin network satisfying:*

- *The maximum* $L_{2,k+1}$*-margin for a given dataset* D_p *is:*

$$\gamma^* = \frac{2(k!)}{(2k+2)^{(k+1)/2}(p-1)p^{(k-1)/2}}.$$

- *For each neuron* $\phi(\{u_1, \dots, u_k, w\}; a_1, \dots, a_k)$ *there is a constant scalar* $\beta \in \mathbb{R}$ *and a frequency* $\zeta \in \{1, \dots, \frac{p-1}{2}\}$ *satisfying*

$$u_1(a_1) = \beta \cdot \cos(\theta^*_{u_1} + 2\pi\zeta a_1/p)$$
$$\dots$$
$$u_k(a_k) = \beta \cdot \cos(\theta^*_{u_k} + 2\pi\zeta a_k/p)$$
$$w(c) = \beta \cdot \cos(\theta^*_{w} + 2\pi\zeta c/p)$$

where $\theta^*_{u_1}, \dots, \theta^*_{u_k}, \theta^*_w \in \mathbb{R}$ *are some phase offsets satisfying* $\theta^*_{u_1} + \cdots + \theta^*_{u_k} = \theta^*_w$.

- *For every frequency* $\zeta \in \{1, \dots, \frac{p-1}{2}\}$*, there exists one neuron using this frequency only.*

Proof. Follow the same proof sketch as Theorem K.1 by Lemma H.8, Condition C.10, Lemma I.3, Lemma G.1, Definition C.7, Lemma G.2, Lemma J.2.

□

L More Experiments

L.1 One-hidden Layer Neural Network

In Figure 5 and Figure 6, we use SGD to train a two-layer network with $m = 1536 = 2^{2k-2} \cdot (p-1)$ neurons, i.e., Eq. equation 3, on $k = 3$-sum mod-$p = 97$ addition dataset, i.e., Eq. equation 1. In

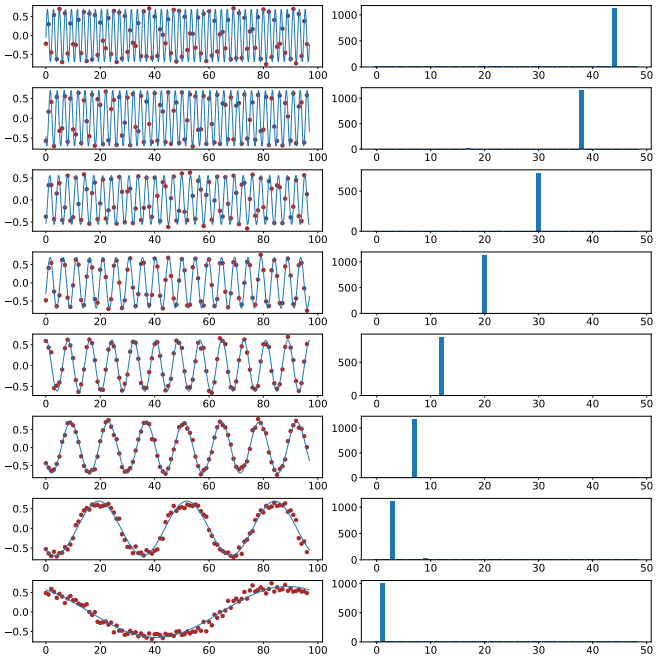


Figure 5: Cosine shape of the trained embeddings (hidden layer weights) and corresponding power of Fourier spectrum. The two-layer network with $m = 1536$ neurons is trained on $k = 3$-sum mod-$p = 97$ addition dataset. We even split the whole datasets ($p^k = 97^3$ data points) into the training and test datasets. Every row represents a random neuron from the network. The left figure shows the final trained embeddings, with red dots indicating the true weight values, and the pale blue interpolation is achieved by identifying the function that shares the same Fourier spectrum. The right figure shows their Fourier power spectrum. The results in these figures are consistent with our analysis statements in Lemma F.1.

Figure 7 and Figure 8, we use SGD to train a two-layer network with $m = 5632 = 2^{2k-2} \cdot (p-1)$ neurons, i.e., Eq. equation 3, on $k = 5$-sum mod-$p = 23$ addition dataset, i.e., Eq. equation 1.

Figure 5 and Figure 7 show that the networks trained with stochastic gradient descent have single-frequency hidden neurons, which support our analysis in Lemma F.1. Furthermore, Figure 6 and Figure 8 demonstrate that the network will learn all frequencies in the Fourier spectrum which is consistent with our analysis in Lemma F.2. Together, they verify our main results in Theorem D.1 and show that the network trained by SGD prefers to learn Fourier-based circuits.

L.2 One-layer Transformer

In Figure 9 , we train a one-layer transformer with $m = 160$ heads attention, i.e., Eq. equation 2, on $k = 3$-sum mod-$p = 61$ addition dataset, i.e., Eq. equation 1. In Figure 10 , we train a one-layer transformer with $m = 160$ heads attention, i.e., Eq. equation 2, on $k = 5$-sum mod-$p = 17$ addition dataset, i.e., Eq. equation 1.

Figure 9 and Figure 10 show that the one-layer transformer trained with stochastic gradient descent learns 2-dim cosine shape attention matrices, which is similar to one-hidden layer neural networks in Figure 5 and Figure 7. It means that the attention layer has a similar learning mechanism as neural networks in the modular arithmetic task, where it prefers to learn Fourier-based circuits when trained by SGD.

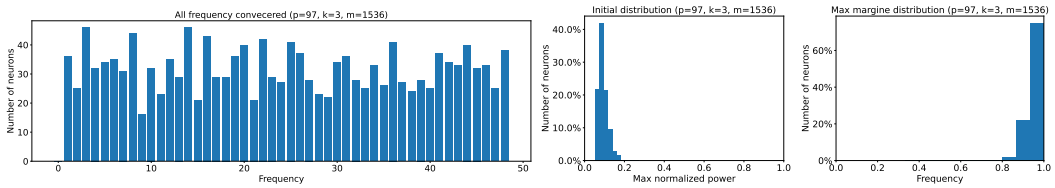


Figure 6: All Fourier spectrum frequencies being covered and the maximum normalized power of the embeddings (hidden layer weights). The one-hidden layer network with $m = 1536$ neurons is trained on $k = 3$-sum mod-$p = 97$ addition dataset. We denote $\widehat{u}[i]$ as the Fourier transform of $u[i]$. Let $\max_i |\widehat{u}[i]|^2/(\sum |\widehat{u}[j]|^2)$ be the maximum normalized power. Mapping each neuron to its maximum normalized power frequency, (a) shows the final frequency distribution of the embeddings. Similar to our construction analysis in Lemma F.2, we have an almost uniform distribution over all frequencies. (b) shows the maximum normalized power of the neural network with random initialization. (c) shows, in frequency space, the embeddings of the final trained network are one-sparse, i.e., maximum normalized power being almost 1 for all neurons. This is consistent with our maximum-margin analysis results in Lemma F.2.

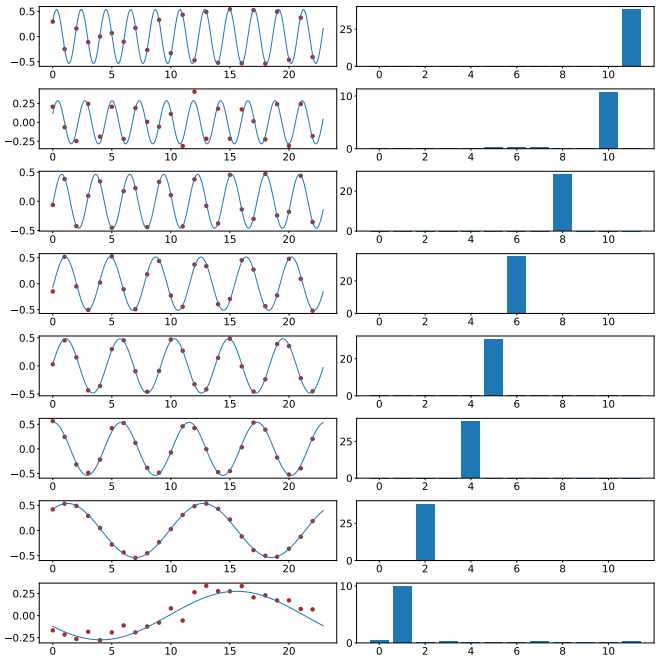


Figure 7: Cosine shape of the trained embeddings (hidden layer weights) and corresponding power of Fourier spectrum. The two-layer network with $m = 5632$ neurons is trained on $k = 5$-sum mod-$p = 23$ addition dataset. We even split the whole datasets ($p^k = 23^5$ data points) into the training and test datasets. Every row represents a random neuron from the network. The left figure shows the final trained embeddings, with red dots indicating the true weight values, and the pale blue interpolation is achieved by identifying the function that shares the same Fourier spectrum. The right figure shows their Fourier power spectrum. The results in these figures are consistent with our analysis statements in Lemma F.1.

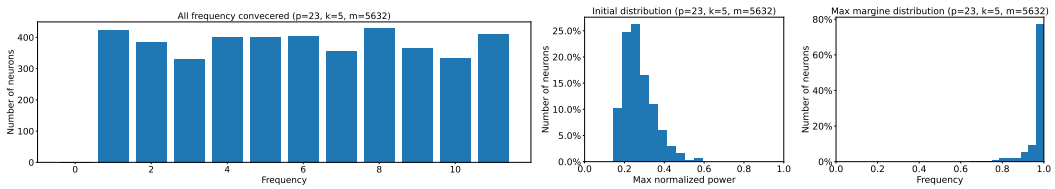


Figure 8: All Fourier spectrum frequencies being covered and the maximum normalized power of the embeddings (hidden layer weights). The one-hidden layer network with $m = 5632$ neurons is trained on $k = 5$-sum mod-$p = 23$ addition dataset. We denote $\widehat{u}[i]$ as the Fourier transform of $u[i]$. Let $\max_i |\widehat{u}[i]|^2/(\sum |\widehat{u}[j]|^2)$ be the maximum normalized power. Mapping each neuron to its maximum normalized power frequency, (a) shows the final frequency distribution of the embeddings. Similar to our construction analysis in Lemma F.2, we have an almost uniform distribution over all frequencies. (b) shows the maximum normalized power of the neural network with random initialization. (c) shows, in frequency space, the embeddings of the final trained network are one-sparse, i.e., maximum normalized power being almost 1 for all neurons. This is consistent with our maximum-margin analysis results in Lemma F.2.

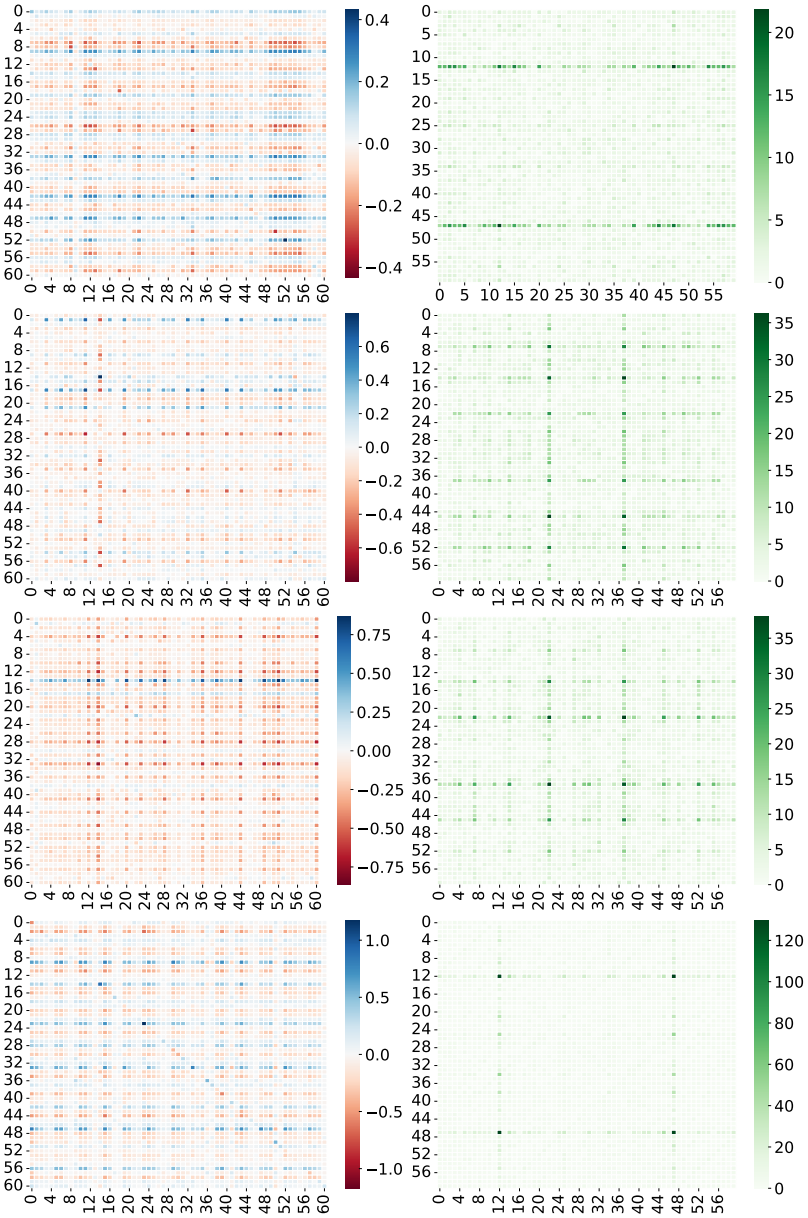


Figure 9: 2-dimension cosine shape of the trained W^{KQ} (attention weights) and their Fourier power spectrum. The one-layer transformer with attention heads $m = 160$ is trained on $k = 3$-sum mod-$p = 61$ addition dataset. We even split the whole datasets ($p^k = 61^3$ data points) into training and test datasets. Every row represents a random attention head from the transformer. The left figure shows the final trained attention weights being an apparent 2-dim cosine shape. The right figure shows their 2-dim Fourier power spectrum. The results in these figures are consistent with Figure 5.

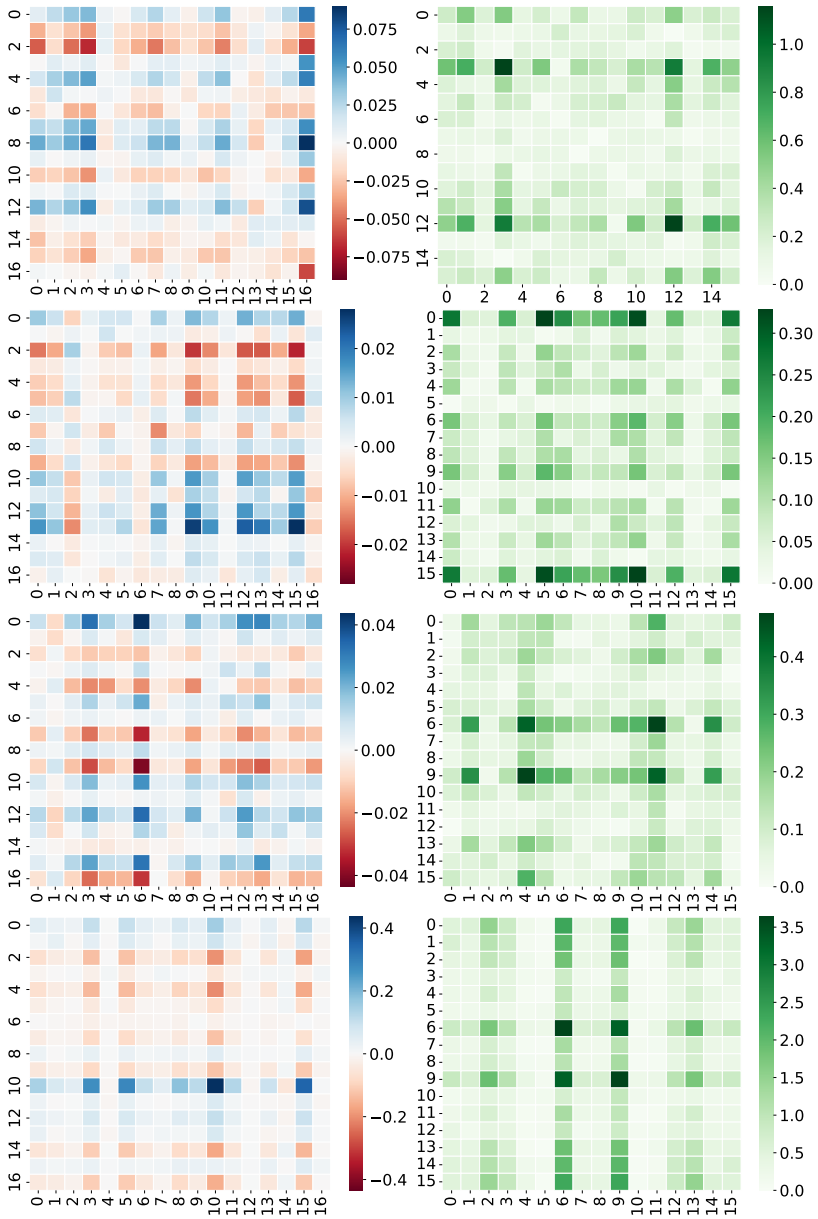


Figure 10: 2-dimension cosine shape of the trained W^{KQ} (attention weights) and their Fourier power spectrum. The one-layer transformer with attention heads $m = 160$ is trained on $k = 5$-sum mod-$p = 17$ addition dataset. We even split the whole datasets ($p^k = 17^5$ data points) into training and test datasets. Every row represents a random attention head from the transformer. The left figure shows the final trained attention weights being an apparent 2-dim cosine shape. The right figure shows their 2-dim Fourier power spectrum. The results in these figures are consistent with Figure 7.