Sparse Fourier Regularization for Modular Arithmetic Models

Amogh Dalal [*1] **Adityasinh Rathod** [*1] **Akshay Rangamani** [1]

Abstract

Modular arithmetic serves as a useful test bed for observing empirical phenomena in deep learning, including *grokking*. Prior work in mechanistic interpretability has shown that sequence models such as transformers and recurrent networks eventually converge to a Fourier multiplication strategy for solving these tasks. In this paper we introduce ℓ_1 regularization in the Fourier space of the (un)embedding layers to bypass grokking and train modular arithmetic models up to $3\times$ faster. We also study the embedding geometry of models trained on multiple arithmetic operations and show how models trained on multiple operations in the same group (like addition and subtraction) use the same Fourier spectrum, while models trained on multiple operations across different groups (like addition and multiplication) entangle their Fourier spectra in the same embedding dimensions - making targeted interventions harder. Here again ℓ_1 Fourier regularization applied to groups of embedding dimensions disentangles the Fourier spectra corresponding to different tasks.

1. Introduction

Modular arithmetic tasks serve as a useful test bed for studying how deep networks learn algorithmic tasks. Interest in these tasks was sparked by Power et al. (2022), who identified a phenomenon called *grokking* where models first memorize the training data and only generalize to unseen examples after prolonged additional training. Since then, a growing body of work has sought to explain grokking through the lens of circuit efficiency (Varma et al., 2023), feature emergence via margin maximization (Morwani et al., 2024), and training dynamics (Mohamadi et al., 2024; Mallinar et al., 2024).

[*]Equal contribution [1]Department of Data Science, New Jersey Institute of Technology, Newark, NJ, USA. Correspondence to: Akshay Rangamani <akshay.rangamani@njit.edu>.

Mechanistic Interpretability Workshop at the 43rd International Conference on Machine Learning, Seoul, South Korea, 2026.
Copyright 2026 by the author(s).

A key insight from mechanistic interpretability investigations of these tasks is that neural networks converge to a Fourier multiplication strategy. Nanda et al. (2023) showed that a one-layer transformer trained on modular addition encodes the inputs in a Fourier basis and computes the sum via trigonometric identities. Zhong et al. (2023) described two related algorithms — the "clock" and the "pizza" — that achieve modular addition in transformers via circular representations. These findings go beyond transformers to MLPs with quadratic activations (Gromov, 2023) and recurrent neural networks (Rangamani, 2025). This convergence of different architectures — transformers, MLPs, RNNs — to the same algorithmic strategy suggests that the Fourier multiplication circuit is a fundamental solution to modular arithmetic tasks rather than an artifact of any particular architecture.

A practical consequence of grokking is that training modular arithmetic models is slow — generalization occurs long after the training loss has saturated, often requiring orders of magnitude more training iterations. Several approaches have been proposed to accelerate this process. Lee et al. (2024) proposed Grokfast, which decomposes the gradient signal into fast-varying and slow-varying components and amplifies the latter, achieving significant speedups on grokking tasks. Xu et al. (2025) showed that initializing a model with Fourier embeddings — or transferring embeddings from a smaller grokked model — can bypass grokking entirely, since the model starts with a representation that already encodes the relevant periodic structure. Weight decay has also been identified as a dominant factor controlling the memorization-to-generalization transition (Liu et al., 2022). These approaches operate either at the level of optimizer dynamics, initialization, or generic regularization.

In this paper, we make the following contributions:

- We introduce ℓ_1 regularization in the Fourier domain of the embedding and unembedding layers to bypass grokking in modular arithmetic tasks. Unlike fixing the embeddings to a predetermined Fourier basis, our approach allows the model to learn its own representation while guiding it toward the known sparse structure. This achieves up to $3\times$ faster convergence when compared to training with weight decay.

- We study the embedding geometry of models trained on multiple modular arithmetic operations and show that operations within the same algebraic group (e.g., addition and subtraction modulo p) reuse the same Fourier spectrum, while operations across different groups (e.g., addition and multiplication) entangle their Fourier spectra within the same embedding dimensions, making targeted causal interventions difficult.
- We show that group-sparse ℓ_1 Fourier regularization applied to segments of the embedding dimensions can disentangle the spectra corresponding to different tasks, restoring the ability to perform targeted interventions.

We release a GitHub code repository for reproducibility.[1]

2. Embedding Geometry of Modular Arithmetic Sequence Models

In this section we characterize the Fourier structure of the embedding and unembedding layers of models trained on modular arithmetic tasks. While some of these results – especially on single operation models – are known in the literature, we present them here for the sake of completeness. We first show that the embedding and unembedding matrices learn a sparse set of Fourier frequencies across all four arithmetic operations and across both RNNs and Transformers. We then study Transformers trained on two operations simultaneously and show that the embedding geometry depends on the relationship between the operations: operations within the same group share a single set of frequencies, while operations across groups use separate frequency sets that become entangled in the same embedding dimensions.

2.1. Experimental Setup

Single-operation RNNs. We train single-layer RNNs on each of the four modular arithmetic operations modulo $p = 113$. The input sequence is of the form $|a|b| = |$ and the model must predict $c = a \circ b \pmod p$ for the appropriate operation $\circ$. For addition and subtraction, we generate all $p^2 = 12769$ pairs (a, b) with $a, b \in \{0, \ldots, p-1\}$. For multiplication and division, we exclude pairs where a or b is zero, yielding $112^2 = 12544$ pairs; for division we further restrict to pairs where b is coprime to p. We use a random 30% of the data for training and evaluate on the remaining 70%. The RNN has a hidden state of size $d_h = 256$ with the $\tanh$ nonlinearity. Input tokens are passed through a trainable embedding W_E and the final hidden state is normalized and passed through an unembedding W_{fc} to predict the output token. The embedding and unembedding are not tied. We train using Adam with full batches, a learning rate of 0.01, and weight decay $\lambda = 5 \times 10^{-5}$ for $20{,}000$ epochs. For multiplication and division we use weight decay $\lambda = 5 \times 10^{-4}$.

Single-operation Transformers. We train single-layer Transformers on modular subtraction and division modulo $p = 79$. The model uses an embedding dimension of $d_h = 256$, 16 attention heads, and an MLP with 512 hidden units using the $\tanh$ nonlinearity and learned positional embeddings. The outputs are normalized using LayerNorm and passed into an unembedding W_{fc}. The embedding and unembedding weights are tied. We train for $15{,}000$ epochs with a learning rate of 0.01.

Multi-operation Transformers. We train single-layer Transformers on pairs of modular arithmetic operations modulo $p = 113$, using the same sequence format $|a|\langle \text{op} \rangle|b| = |$ and operation-batch interleaving across all six operation pairs. Relative to the earlier setup, we use a larger model ($d_h = 512$, 8 heads, MLP width 1024), train for $10{,}000$ epochs with learning rate 10^{-3} and weight decay 1.0, and apply Fourier regularization without splitting the embedding space by operation group.

2.2. Sparse Fourier Spectra in Single-Operation Models

To analyze the Fourier structure of the embeddings, we construct a Fourier basis matrix $F_p \in \mathbb{R}^{p \times p}$ consisting of a constant row and $\{\cos(2\pi kn/p)\}_{n=1}^{p}, \{\sin(2\pi kn/p)\}_{n=1}^{p}$ for $k = 1, \ldots, \lfloor (p-1)/2 \rfloor$. For addition and subtraction, we compute the Fourier coefficients of the embedding and unembedding matrices by multiplying by F_p along the vocabulary dimension. For multiplication and division, we first permute the rows of W_E (and columns of W_{fc}) according to the discrete logarithm with respect to a generator g of $\mathbb{Z}_p^*$ (we use $g = 3$) to obtain W_E^{π}, W_{fc}^{π}, so that the token ordering reflects the cyclic group structure of multiplication. We then apply F_{p-1} as before. In both cases, the resulting spectra are normalized to unit ℓ_2 norm.

Across all four operations and both architectures, we find that the Fourier spectra of W_E and W_{fc} are sparse: only a small number of frequencies $\omega_k = 2\pi k/p$ carry significant energy, while the remaining coefficients are near zero. In this paper, we only show the spectra of two operation - model pairs: Figure 1 shows an RNN trained on subtraction, and Figure 2 shows a Transformer trained on division. For a model with K significant frequencies, the embedding maps each input token a into a $2K$-dimensional subspace spanned by $\{\cos(\omega_k a), \sin(\omega_k a)\}$ for the significant frequencies ω_k, similar to the observations made in Rangamani (2025).

[1] GitHub repository: https://github.com/adirathodd/fourier_regularization

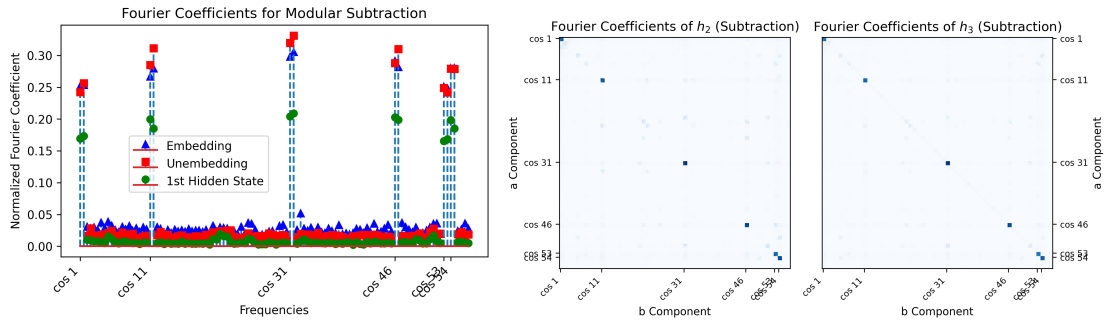


Figure 1. Recurrent Network trained on Modular Subtraction. We observe consistent sparse Fourier spectra in the embedding, unembedding layers, as well as the first hidden state h_1. On the right we plot the 2D Fourier coefficients of the second and third hidden states h_2, h_3. The same frequencies appear there as well.

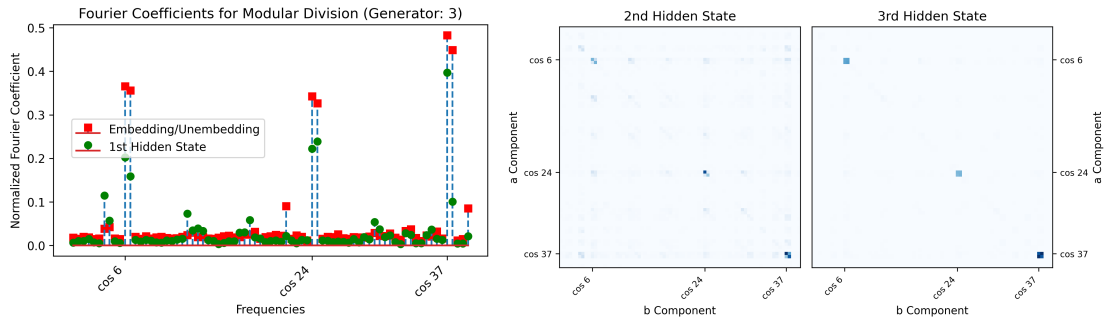


Figure 2. Transformer trained on Modular Division. We observe consistent sparse Fourier spectra in the embedding, unembedding layers, as well as the first hidden state h_1. On the right we plot the 2D Fourier coefficients of the second and third hidden states h_2, h_3. The sparse Fourier spectra appear when all quantities are permuted according to their discrete logarithm order with respect to the generator $g = 3$.

2.3. Sparse Fourier Spectra in Multi-Operation Models

We train single-layer Transformers with the same architecture on all six pairwise combinations of the four operations with $p = 113$, and achieve near-perfect generalization in each case (Table 1).

We then analyze the Fourier spectra of the embedding and unembedding layers and find that the structure depends on whether the two operations are performed on the same group or different groups.

2.3.1. Operations on the Same Group: Shared Fourier Spectra

For pairs of operations within the same algebraic group – addition/subtraction (inverses in $(\mathbb{Z}_p, +)$) and multiplication/division (inverses in $(\mathbb{Z}_p^*, \times)$) – the model learns a single shared set of K Fourier frequencies. The Fourier spectra of W_E and W_{fc} are sparse and identical to what a single-operation model would learn (Figure 3).

SVD truncation of the embedding matrix confirms this: retaining the top $2K + 1$ singular vectors of the embedding matrix W_E is sufficient to maintain near-perfect accuracy on both operations. Here $2K$ vectors are used for the number token embeddings, while the $2K+1^{\text{th}}$ additional component corresponds to the direction associated with the operation tokens. The operation tokens are usually orthogonal to the number tokens, and nearly anti-parallel to each other.

Ablation experiments that remove singular vectors from the embedding layers of multi-op models one by one further confirm that both operations share the same set of frequencies shared. Since both operations require a large fraction of the frequencies to compute the answers accurately, removing any subset of the important singular vectors of W_E degrades performance on both operations, not just one. The model does not partition its embedding into separate subspaces (or Fourier frequencies) for each operation – it reuses the same representation for both.

2.3.2. Operations on Different Groups: Separate but Entangled Fourier Spectra

For pairs of operations across different groups — such as addition and multiplication — the embedding must represent two distinct sets of Fourier frequencies: K_1 for the additive operation and K_2 for the multiplicative one. The total number of significant singular components in the embedding is $2K_1 + 2K_2 + 1$, consistent with the model representing two

Ops Pair	Exact Accuracy on both operations (%)
Addition + Division	100.0000
Addition + Multiplication	99.9802
Addition + Subtraction	100.0000
Division + Multiplication	100.0000
Division + Subtraction	100.0000
Multiplication + Subtraction	100.0000

Table 1. Exact full-dataset accuracy for 2-term multi-op pairwise transformer models.

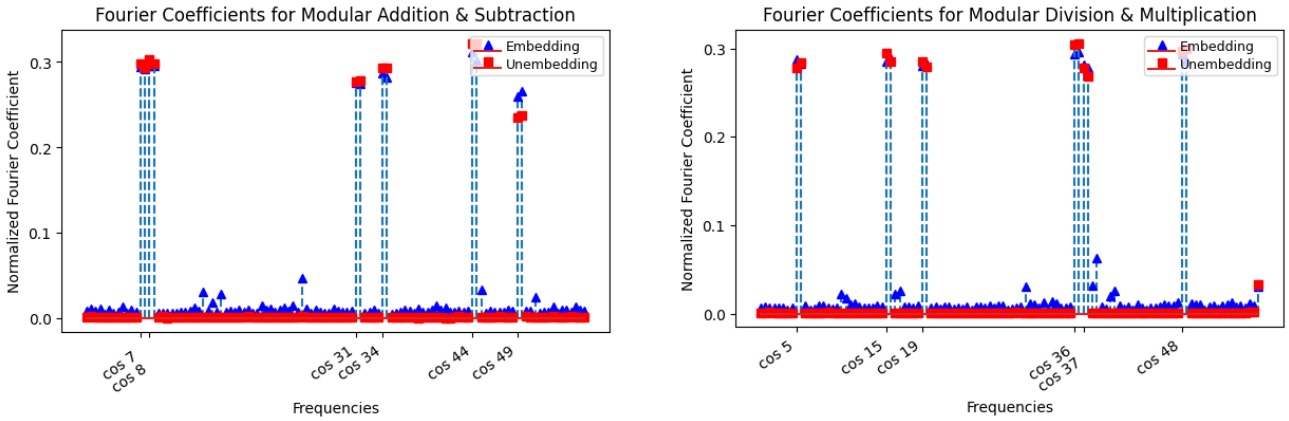


Figure 3. Fourier spectra for same-group operation pairs (addition/subtraction and multiplication/division). Both W_E and W_{fc} concentrate on the same sparse set of frequencies, similar to the case of single-operation training.

independent Fourier bases (and including the component for the op tokens). However, these two representations are not cleanly separable. They are entangled in the embedding dimensions, as well as the singular vectors. The Fourier spectra directly show the entanglement in embedding dimensions. In Figure 4 we see that computing $F_p W_E$ in the natural token ordering reveals the additive frequencies as sparse peaks, but the multiplicative frequencies — which are sparse only after permuting by the discrete logarithm order according to the generator $g = 3$ — appear as a noisy floor. Conversely, permuting the embedding rows by the discrete logarithm order W_E^{π} and then applying F_{p-1} reveals the multiplicative frequencies as sparse peaks, while the additive frequencies appear as a noisy floor. Neither ordering produces a sparse Fourier spectrum (Figure 4).

We can show that the Fourier spectra of two operations on two different groups are also entangled in the singular vector space of the embedding W_E. SVD ablation experiments confirm this. Again in Figure 5 we can see that removing singular components of W_E affects accuracy on both operations, even though the operations use different frequencies. Thus we can conclude that the model has learned to represent both frequency sets in overlapping embedding dimensions.

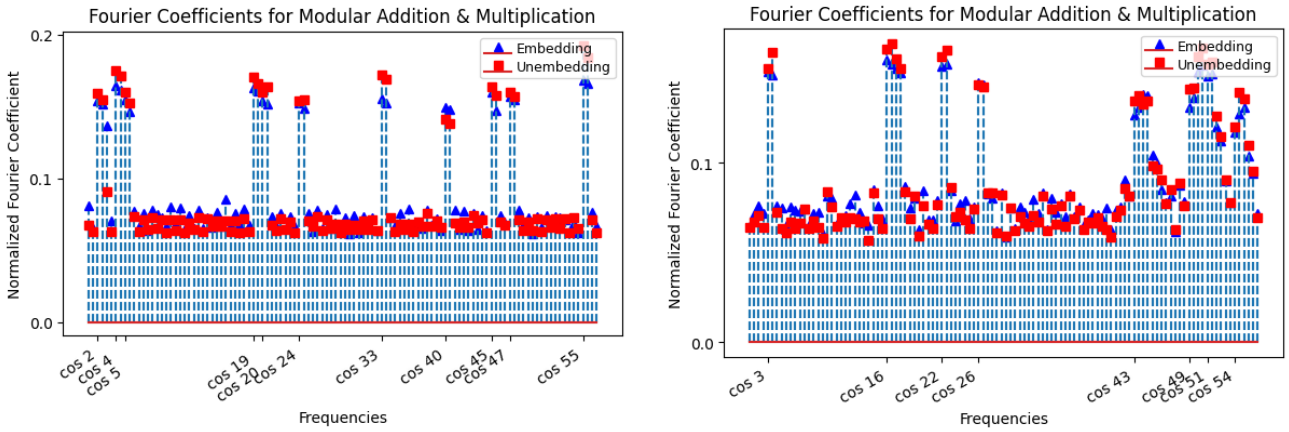
3. ℓ_1 Fourier Regularization for Modular Arithmetic Models

In the previous section we showed that generalizing solutions to modular arithmetic tasks are characterized by sparse Fourier spectra in the embedding and unembedding layers. We now exploit this structure by introducing an ℓ_1 penalty on the Fourier coefficients of these layers during training. We show that this regularization bypasses the grokking phase and accelerates generalization. We then extend this idea to multi-operation models, where applying the regularization to groups of embedding dimensions disentangles the representations of incompatible operations.

3.1. Bypassing Grokking in Single Operation Models

It is well established that weight decay is necessary for models to generalize on modular arithmetic tasks, but models trained with weight decay alone undergo a prolonged grokking phase: the training loss converges quickly while test accuracy remains at chance for many additional epochs. Since we know that the generalizing solution is sparse in the Fourier domain, we can directly encourage this structure by adding an ℓ_1 penalty on the Fourier coefficients of the embedding and unembedding layers.

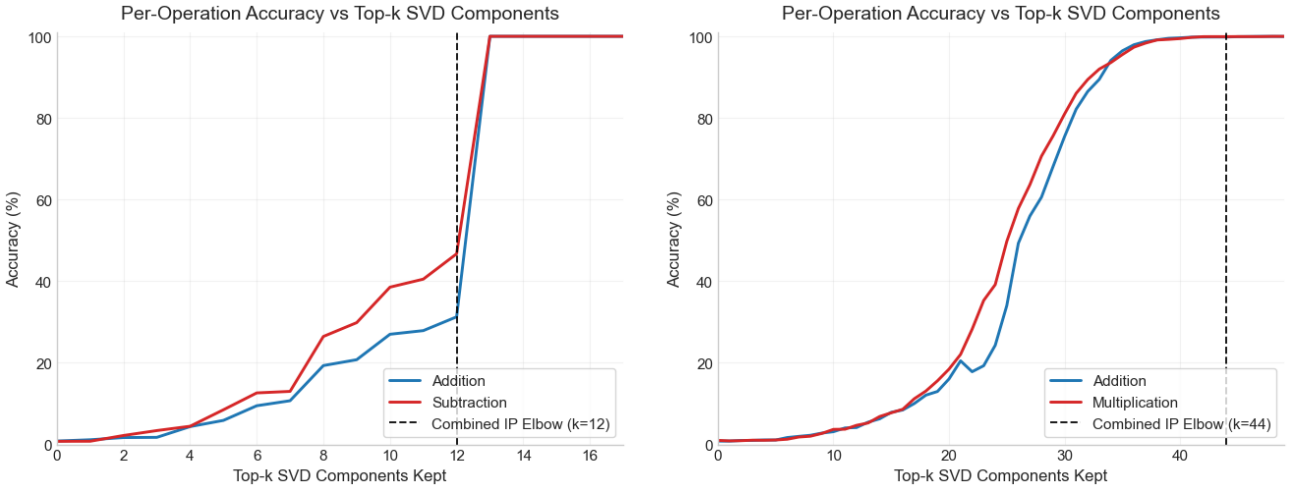
Specifically, let $F_p \in \mathbb{R}^{p \times p}$ be the Fourier basis matrix



(a) Natural token ordering.

(b) Discrete-log permuted ordering.

Figure 4. Natural ordering reveals sparsity in the Fourier spectrum of the Add embedding components, while discrete-log permutation reveals sparsity in the Fourier spectrum of the Multiplication embeddings; no single ordering is perfectly sparse indicating the two bases are entangled in the embedding dimensions.



(a) Same-Group operation pair.

(b) Different-Group operation pair.

Figure 5. SVD ablation curves for same-group vs. different-group operation pairs. Removing singular components of W_E degrades accuracy on both operations, indicating that the model encodes both frequency sets in overlapping embedding dimensions. For same-group operations this indicates that both operations use the same set of Fourier frequencies. For different-group operations this indicates the entanglement between both operations in every singular vector.

defined in Section 2.1. We add the penalty term

$$\mathcal{L}_F = \lambda_F \left(\|F_p W_E\|_1 + \|F_p W_{fc}\|_1 \right) \tag{1}$$

to the training loss, where λ_F controls the strength of the regularization. This penalty encourages the embedding to map tokens into a sparse set of Fourier modes, and the unembedding to decode from the same sparse set.

We compare models trained with weight decay alone against models trained with Fourier regularization $\lambda_F = 0.01$ on the modular addition task ($p = 113$) using 10 identical RNN configurations, each with a different random seed. Models trained with weight decay generalize (reaching $>$ 99.9% test accuracy) at an average of 7,032 epochs with a standard deviation of 1,345. Models trained with Fourier regularization generalize at an average of 1,857 epochs with a standard deviation of 499 — approximately 3$\times$ faster, with substantially less variance across runs. We show the loss curves in Figure 6, and present the results Table 2.

This result indicates that by regularizing towards the right embedding geometry, we can bypass the grokking stage in training sequence models on modular arithmetic tasks. The regularization term accelerates convergence to a sparse Fourier spectrum, rather than waiting for weight decay to implicitly find this solution through grokking.

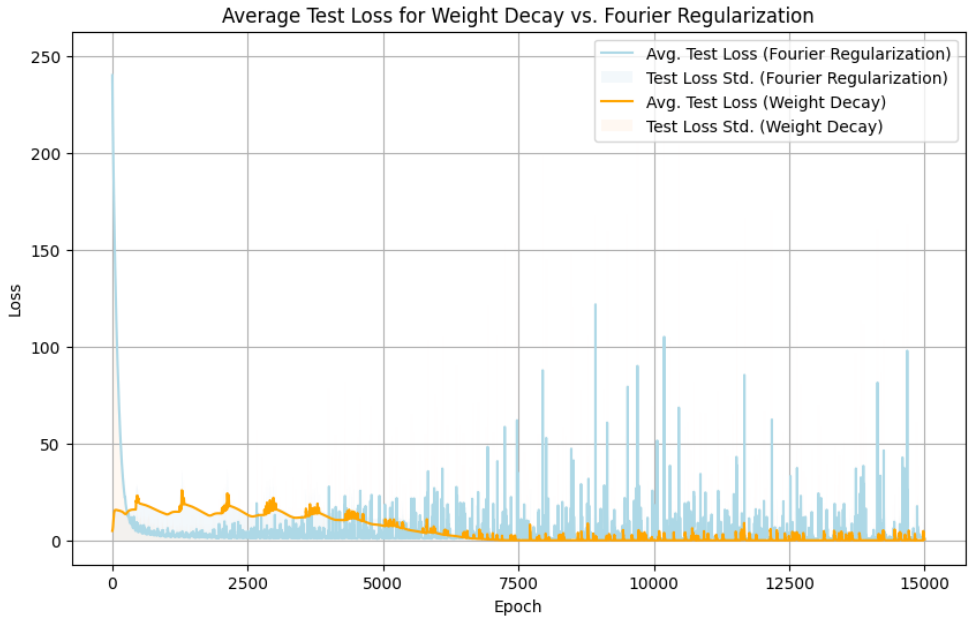


Figure 6. The average train and test loss using weight decay (orange) and fourier regularization (blue), across 10 different random seeds. Models trained using fourier regularization consistently generalize without grokking.

	Average Epoch of Generalization	Std. Dev.
Group 1 (Weight Decay)	7,031.5	1,344.62
Group 2 (Fourier Regularization)	1,857.2	498.67

Table 2. Comparison of Models trained using Weight Decay vs. Fourier Regularization

3.2. Disentangling Multi-Operation Embeddings

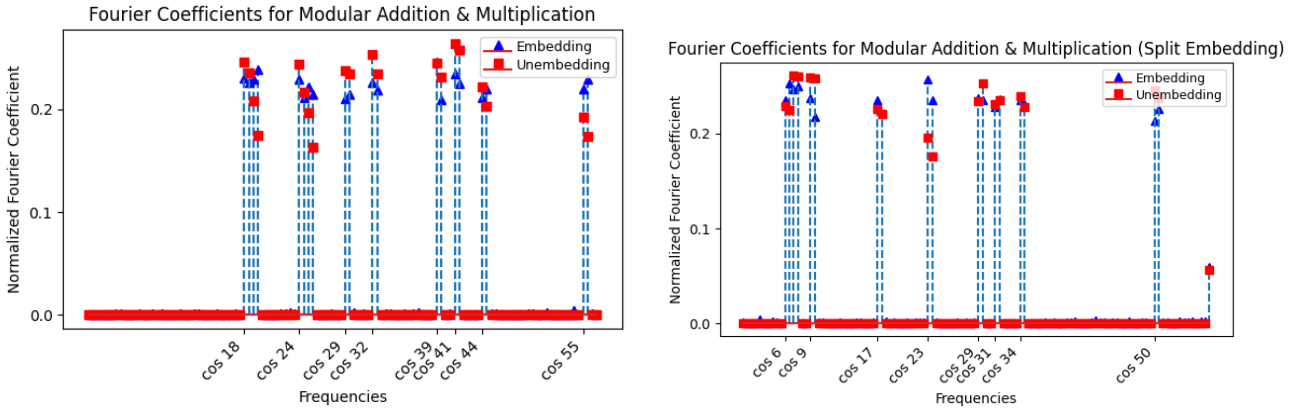
In Section 2.3 we showed that models trained on different-group operations (e.g., addition and multiplication) entangle the Fourier representations of both operations in the same embedding dimensions. We now show that a variant of Fourier regularization can disentangle these representations by encouraging different groups of embedding dimensions to encode different operations.

We partition the embedding dimensions of W_E into two groups of equal size: the first $d_h/2$ dimensions and the last $d_h/2$ dimensions. We apply the Fourier regularization penalty separately to each group, using the appropriate Fourier basis for each operation. For the group assigned to the additive operation, we penalize $\|F_p W_E^{(1)}\|_1$ where $W_E^{(1)}$ denotes the first $d_h/2$ columns of W_E. For the group assigned to the multiplicative operation, we first permute the rows of $W_E^{(2)}$ by the discrete logarithm order for generator $g = 3$ and then penalize $\|F_{p-1} W_E^{(2),\pi}\|_1$. The same grouping is applied to W_{fc}.This grouped ℓ_1 regularization encourages the model to represent the additive Fourier frequencies in one set of embedding dimensions and the multiplicative Fourier frequencies in the other.

We verify disentanglement in two ways. First, we compute the Fourier spectra of each half of the embedding separately: $F_p W_E^{(1)}$ yields sparse additive frequencies with no noisy floor from the multiplicative representation, and $F_{p-1} W_E^{(2),\pi}$ yields sparse multiplicative frequencies with no noisy floor from the additive representation. This indicates the two halves of the embedding dimensions separate the encoding of the two representations.

Second, we perform embedding ablations in Table 3. We zero out each half of the embedding dimensions and compute the accuracy of the model in computing both operations. We find that masking the embedding dimensions corresponding to an operation degrades performance on that operation, while accuracy on the other operation is nearly untouched.

Without regularization, as shown in Section 2.3, neither verification is possible — the Fourier spectra of any subset of embedding dimensions contain a noisy floor from the other operation, and removing embedding dimensions degrades both operations.



(a) Additive group ($W_E^{(1)}$).

(b) Multiplicative group ($W_E^{(2),\pi}$).

Figure 7. Fourier spectra after grouped regularization. The first half of embedding dimensions exhibits sparse additive frequencies, while the second half (after discrete-log permutation) exhibits sparse multiplicative frequencies, with no cross-contamination between groups.

Operation	Baseline Accuracy (%)	Mask Addition Embeddings	Mask Multiplication Embeddings
Addition	100.00	1.25	96.78
Multiplication	99.98	95.48	0.71

Table 3. Ablation accuracy after zeroing each group of dimensions.

4. Discussion & Conclusion

In this paper we studied the embedding geometry of sequence models trained on modular arithmetic tasks. We showed that across all four arithmetic operations and across both RNNs and Transformers, the embedding and unembedding layers learn sparse Fourier representations. We then showed that single-layer Transformers trained on pairs of operations exhibit two distinct regimes: compatible operations (addition/subtraction, multiplication/division) share a single set of Fourier frequencies, while incompatible operations (e.g., addition and multiplication) learn separate frequency sets that become entangled in the same embedding dimensions. We introduced ℓ_1 Fourier regularization on the embedding and unembedding layers and showed that it bypasses grokking, achieving generalization approximately $3\times$ faster than weight decay alone. We also showed that applying this regularization to groups of embedding dimensions disentangles the representations of incompatible operations.

We observe two broader points from these findings. First, the embedding controls the learned solution: once the embedding encodes tokens in a sparse Fourier basis, the remaining layers implement the corresponding trigonometric circuit. When we know the structure of the right embedding, we can regularize towards it and accelerate training. Second, when training on multiple tasks, the representations of different tasks get entangled in the same embedding dimensions by default. To obtain models whose components are more easily editable and composable, we may want to induce structure in the embedding — and Fourier regularization provides one way to do so.

These findings are specific to modular arithmetic, and it remains to be seen whether the same approach — identifying the right embedding geometry and regularizing towards it — can accelerate training on other algorithmic tasks. We are also interested in whether the multi-task entanglement we observe here occurs in language models, where the embedding must support many tasks simultaneously, and whether designing the embedding geometry can disentangle them.

Acknowledgements

Amogh Dalal and Adityasinh Rathod were supported by an NJIT Grace Hopper AI Research Institute Seed Grant Award.

References

Gromov, A. (2023). Grokking modular arithmetic. *arXiv preprint arXiv:2301.02679*.

Lee, J., Kang, B. G., Kim, K., and Lee, K. M. (2024). Grokfast: Accelerated grokking by amplifying slow gradients. *arXiv preprint arXiv:2405.20233*.

Liu, Z., Michaud, E. J., and Tegmark, M. (2022). Omnigrok: Grokking beyond algorithmic data. *arXiv preprint arXiv:2210.01117*.

Mallinar, N., Beaglehole, D., Zhu, L., Radhakrishnan, A., Pandit, P., and Belkin, M. (2024). Emergence in non-neural models: grokking modular arithmetic via average gradient outer product. *arXiv preprint arXiv:2407.20199*.

Mohamadi, M. A., Li, Z., Wu, L., and Sutherland, D. J. (2024). Why do you grok? a theoretical analysis of grokking modular addition. *arXiv preprint arXiv:2407.12332*.

Morwani, D., Edelman, B. L., Oncescu, C.-A., Zhao, R., and Kakade, S. M. (2024). Feature emergence via margin maximization: case studies in algebraic tasks. In *The Twelfth International Conference on Learning Representations*.

Nanda, N., Chan, L., Lieberum, T., Smith, J., and Steinhardt, J. (2023). Progress measures for grokking via mechanistic interpretability. *arXiv preprint arXiv:2301.05217*.

Power, A., Burda, Y., Edwards, H., Babuschkin, I., and Misra, V. (2022). Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*.

Rangamani, A. (2025). Low rank and sparse fourier structure in recurrent networks trained on modular addition. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Varma, V., Shah, R., Kenton, Z., Kramár, J., and Kumar, R. (2023). Explaining grokking through circuit efficiency. *arXiv preprint arXiv:2309.02390*.

Xu, Z., Ni, Z., Wang, Y., and Hu, W. (2025). Let me grok for you: Accelerating grokking via embedding transfer from a weaker model. *arXiv preprint arXiv:2504.13292*.

Zhong, Z., Liu, Z., Tegmark, M., and Andreas, J. (2023). The clock and the pizza: Two stories in mechanistic explanation of neural networks. *Advances in Neural Information Processing Systems*, 36.