Neural Networks Provably Learn Spectral Representations for Group Composition

Jianliang He

Department of Statistics and Data Science, Yale University

August 31, 2026

Yale University

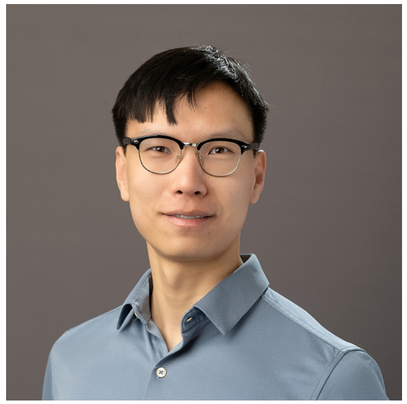
Introduction

- He, J., Wang, L., Zhang, F., Chen, S., & Yang, Z. (2026). Neural Networks Provably Learn Spectral Representations for Group Composition. *arXiv preprint arXiv:2606.02993*.
- He, J., Wang, L., Chen, S., & Yang, Z. (2026). On the mechanism and dynamics of modular addition: Fourier features, lottery ticket, and grokking. *arXiv preprint arXiv:2602.16849*.

Leda Wang
Yale

Siyu Chen
Yale

Fengzhuo Zhang
Yale



Zhuoran Yang
Yale

Background: Representation in DL

- Representation learning has been central since the revival of neural networks. The original sensational backpropagate paper [Rumelhart, Hinton, Williams ' 86] argued that hidden units learn task-relevant features

Letter | Published: 09 October 1986

Learning representations by back-propagating errors

David E. Rumelhart, Geoffrey E. Hinton & Ronald J. Williams

Nature **323**, 533–536 (1986) | Cite this article

Abstract

We describe a new learning procedure, back-propagation, for networks of neurone-like units. The procedure repeatedly adjusts the weights of the connections in the network so as to minimize a measure of the difference between the actual output vector of the net and the desired output vector. As a result of the weight adjustments, internal 'hidden' units which are not part of the input or output come to represent important features of the task domain, and the regularities in the task are captured by the interactions of these units. The ability to create useful new features distinguishes back-propagation from earlier, simpler methods such as the perceptron-convergence procedure[1].

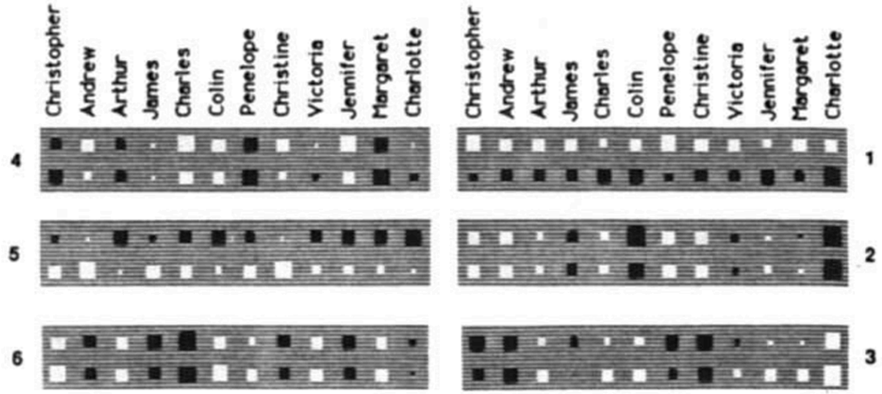


Fig. 4 The weights from the 24 input units that represent people to the 6 units in the second layer that learn distributed representations of people. White rectangles, excitatory weights; black rectangles, inhibitory weights; area of the rectangle encodes the magnitude of the weight. The weights from the 12 English people are in the top row of each unit. Unit 1 is primarily concerned with the distinction between English and Italian and most of the other units ignore this distinction. This means that the representation of an English person is very similar to the representation of their Italian equivalent. The network is making use of the isomorphism between

Background: Representation in DL

- **Transfer learning:** Reuse a representation learned on one task or dataset for a new task [Donahue et al., ’14, Huang et al., ’23, Ilharco et al., ’23]
- **Embeddings:** Represent words, images, or other objects as vectors whose geometry encodes useful similarity and relations. [Mikolov et al., ’13]

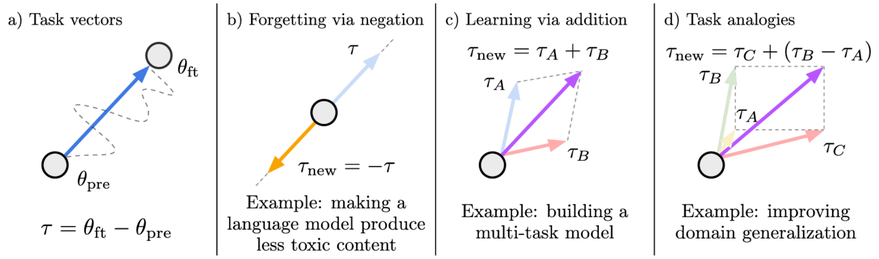


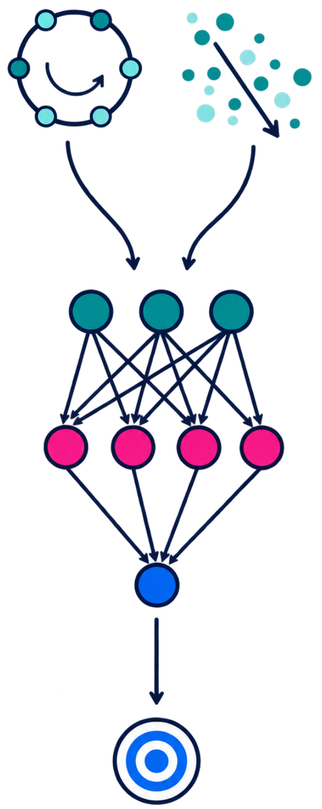
Figure 1: An illustration of task vectors and the arithmetic operations we study for editing models. (a) A task vector is obtained by subtracting the weights of a pre-trained model from the weights of the same model after fine-tuning (Section 2). (b) Negating a task vector degrades performance on the task, without substantial changes in control tasks (Section 3). (c) Adding task vectors together improves the performance of the pre-trained model on the tasks under consideration (Section 4). (d) When tasks form an analogy relationship such as supervised and unsupervised learning on two different data sources, it is possible to improve performance on a supervised target task using only vectors from the remaining three combinations of objectives and datasets (Section 5).

Ilharco, G., Ribeiro, M. T., Wortsman, M., Gururangan, S., Schmidt, L., Hajishirzi, H., & Farhadi, A.. Editing models with task arithmetic. ICLR 2023

Despite their empirical success, these methods usually treat a representation operationally—as a hidden feature vector, embedding. This yields question:

Can we formally define the representations learned by neural networks and characterize what makes a representation useful or correct?

Define Representation: Data and Model



- **Data**: $(x_i, y_i) \overset{\text{i.i.d}}{\sim} \mathsf{P}$ for all $i \in [n]$
  - E.x.1: single index model $y_i = g(\langle x_i, \theta_* \rangle) \in \mathbb{R}$ with $\theta_* \in \mathbb{R}^d$, $x_i \sim \mathsf{P}_x \in \mathbb{P}(\mathbb{R}^d)$
  - E.x.2: modular addition $\mathscr{D}_p = \{(x_1, x_2, y) : y = (x_1 + x_2) \bmod p,\ x_1, x_2 \in \mathbb{Z}_p\}$

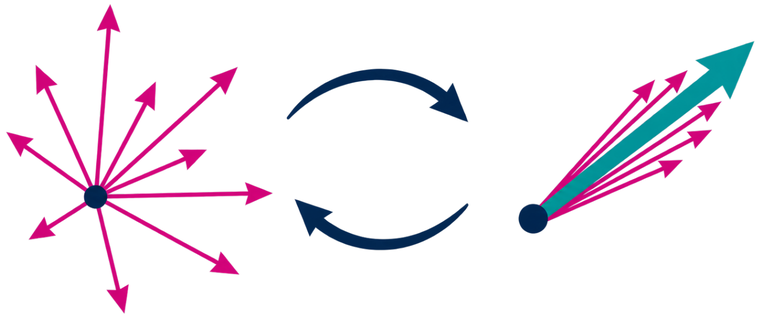
    where $\mathbb{Z}_p = \{0,1,...,p-1\}$, $\mathsf{P} = \mathrm{Unif}(\mathscr{D}_p)$
- **Model**: function class $\mathscr{F}_\theta$ parameterized by $\theta \in \mathbb{R}^d$
  - E.x.: Two-layer Neural Network $f_\theta(x) = \sum_{m=1}^{M} \xi_m \cdot \sigma(\langle \theta_m, x \rangle),\ \theta = \{(\theta_m, \xi_m)\}_{m=1}^{M}$

$\Longrightarrow$ **Two different representation:**

(Implicit) Data Representation + (Learned) Model Representation (DL context)

Alignment of Data and Model Representation

- **Data**: $(x_i, y_i) \overset{\text{i.i.d}}{\sim} \mathsf{P}$ for all $i \in [n]$
  - E.x.: single index model $y_i = g(\langle x_i, \theta_* \rangle) \in \mathbb{R}$ with $\theta_* \in \mathbb{R}^d$
- **Model**: function class $\mathscr{F}_\theta$ parameterized by $\theta \in \mathbb{R}^d$
  - E.x.: Two-layer Neural Network $f_\theta(x) = \sum_{m=1}^{M} \xi_m \cdot \sigma(\langle \theta_m, x \rangle + b_m)$



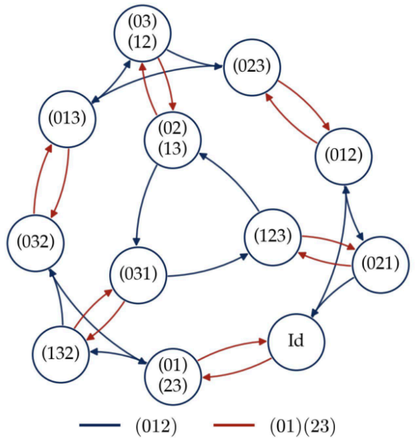
Example of Representation Alignment: Previous work show that $\langle \theta_m, \theta_* \rangle$ grow from 0 and try approaching 1 for all m during the training until some attains $\langle \theta_m, \theta_* \rangle \approx 1$.

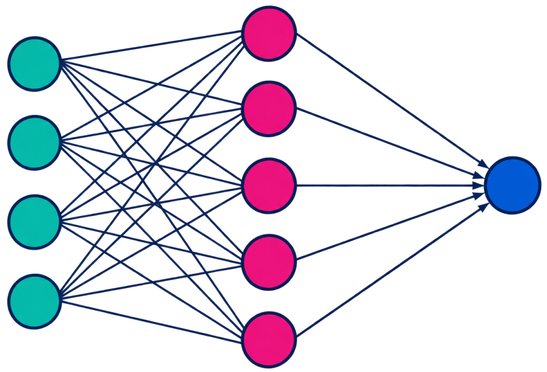
Representation Alignment: Narrowly, we call a model representation "correct" when its learned weights align with the data's representation (feature).

Bietti, A., Bruna, J., Sanford, C., & Song, M. J. (2022). Learning single-index models with shallow neural networks. *Advances in neural information processing systems*, *35*, 9768-9783.

Questions We Address in This Work

- **Representation Alignment:** Can this representation-alignment perspective extend beyond re-specified linear features, e.g., index model, to more natural data and tasks?
- **Identification:** Can we reverse-engineer a trained network to identify the representation encoded?
  - **Mechanism:** Once identified, can we explain how the network uses this representation to compute the correct output?
  - **Training dynamics:** How do candidate representations emerge during training?





*To this end, we study how **Two-Layer Neural Network** learn to perform **Group Composition Task.***

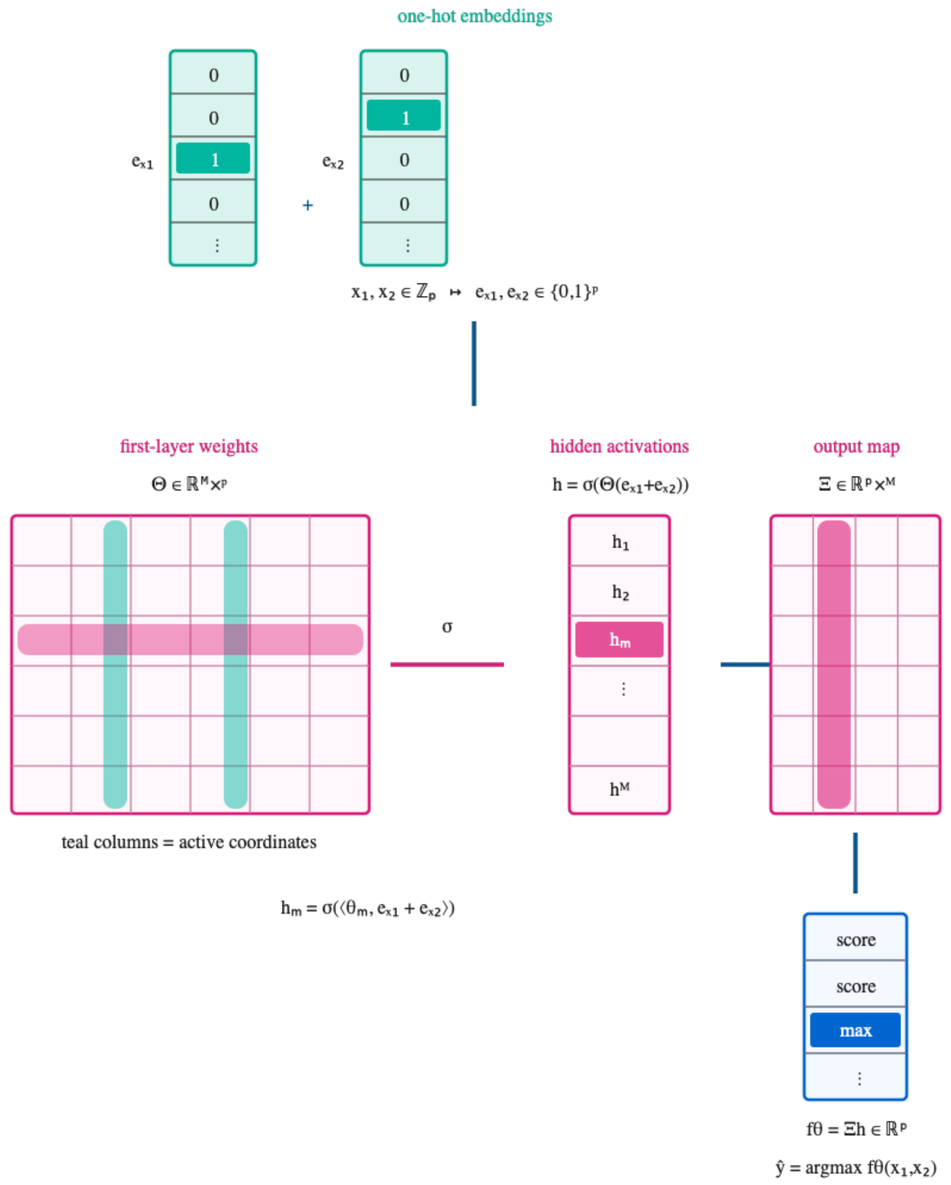
Case Study: Modular Addition

- **Data**: $\mathscr{D}_p = \{(x_1, x_2, y) : y = (x_1 + x_2) \bmod p,\ x_1, x_2 \in \mathbb{Z}_p\}$ where

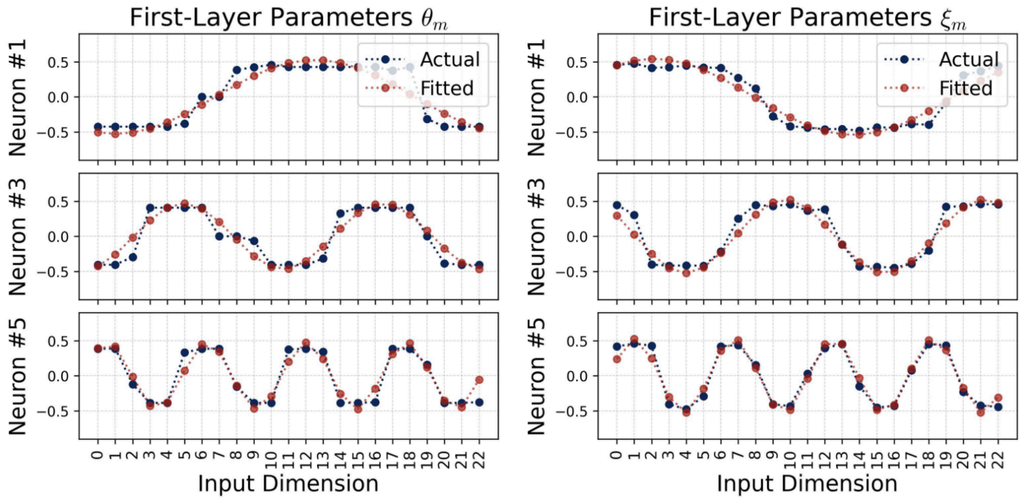
  $\mathbb{Z}_p = \{0,1,...,p-1\}$ and p is prime, e.g., $(7 + 12) \bmod 13 = 6$

- **Model**: Two-layer NN $f_\theta(x_1, x_2) = \sum_{m=1}^{M} \xi_m \cdot \sigma(\langle \theta_m, e_{x_1} + e_{x_2} \rangle) \in \mathbb{R}^p$

  - **Parameters**: $\theta = \{(\theta_m, \xi_m)\}_{m=1}^{M},\ \theta_m, \xi_m \in \mathbb{R}^p$
  - **Input/Embedding**: one-hot $x \in \mathbb{Z}_p \mapsto e_x \in \{0,1\}^p$, $\theta_m[j]$ denotes the logit contribution of neuron m for input j at 1st layer
  - **Activation**: $\max\{\cdot, 0\}$ for experiment and $(\cdot)^2$ for theory
  - **Output**: $\hat{y} = \text{argmax}(\text{smax}(f_\theta(x_1, x_2)))$ where $\text{smax}(x)_i = \exp(x[i]) / \sum_j \exp(x[j])$. $\xi_m[j]$ denotes the logit contribution of neuron m for output/prediction at j
  - **Training:** CE loss + GD/AdamW



Neural Network Learn Cosine Waves



- After training, each neuron's weights follow a clean cosine pattern at shared **single frequency**

$$\theta_m[j] = \alpha_m \cdot \cos(\omega_k j + \phi_m), \quad \xi_m[j] = \beta_m \cdot \cos(\omega_k j + \psi_m)$$

where $w_k = 2\pi k/p$ for all $k \in \{1,\ldots,(p-1)/2\}$

  - Magnitude: $\alpha_m, \beta_m \in \mathbb{R}_+$
  - Phase: $\phi_m, \psi_m \in (0, 2\pi]$

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

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

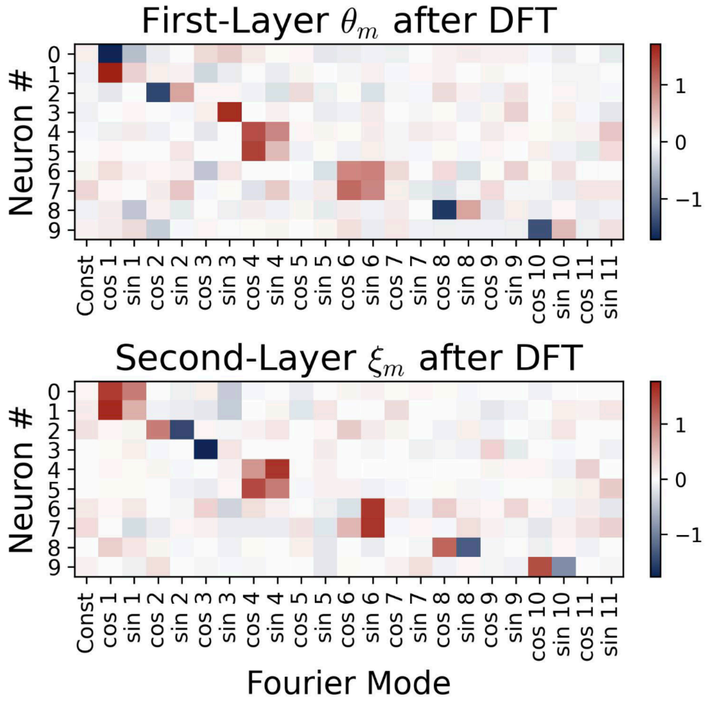
Finding 1: NN Learn Fourier Representation

- **Representation Perspective**. Cosine waves indexed by k are **Fourier representation** of $\mathbb{Z}_p$. Mathematically, DFT gives

$$\{\mathbf{1}_p, \cos(w_1\cdot), \sin(w_1\cdot), \ldots \cos(w_{(p-1)/2}\cdot)), \sin(w_{(p-1)/2}\cdot))\}$$

forms an orthogonal basis such that for any $\nu \in \mathbb{R}^p$, we can write

$$\begin{aligned}\nu[j] &= \alpha^0 + \sum_{k=1}^{(p-1)/2} \alpha_1^k \cdot \cos(\omega_k j) + \sum_{k=1}^{(p-1)/2} \alpha_2^k \cdot \sin(\omega_k j) \\ &= \alpha^0 + \sum_{k=1}^{(p-1)/2} \alpha^k \cdot \cos(\omega_k j + \phi^k), \qquad \forall j \in [p],\end{aligned}$$



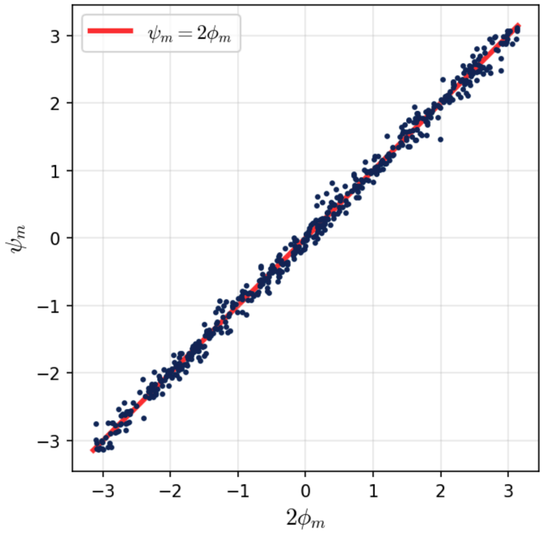
Finding 1:

- Each neuron's DFT has only *one non-zero frequency (representation)* that dominates, no zero frequency
- Both layers share *the same dominant frequency* for each neuron

Finding 2: Phase Alignment

Finding 2: the input phase ϕ_m of θ_m and output ψ_m of ξ_m are aligned:

$$\psi_m = 2\phi_m \bmod 2\pi$$



- **A Short Explanation of** $2\phi_m$. Consider the output of first layer:

$$\sigma(\theta_m[x_1] + \theta_m[x_2]) \propto (\cos(w_k x_1 + \phi_m) + \cos(w_k x_1 + \phi_m))^2$$
$$= \cos(\omega_k(x-y)/2)^2 \cdot \{1 + \cos(\omega_k(x+y) + 2\phi_m)\},$$

which indicates an *alignment of weight within the representation space*!

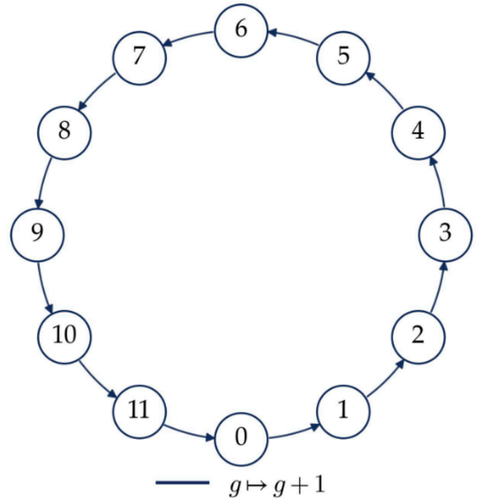
Q: *Does this phenomenon (1) each neuron learning one shared representation across layers, and (2) first- and second-layer weights aligning within the representation space—hold for general groups?*

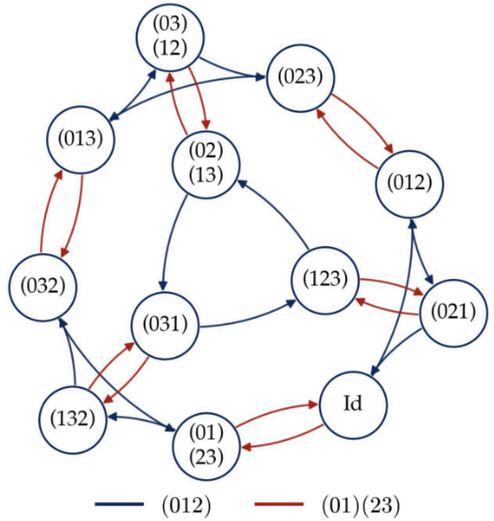
Setup: Group Composition

- **Group**: Let $(G, \star)$ denote a group, which is defined as a set G equipped with a binary operation $\star : G \times G \to G$ satisfying
  - **Associativity.** $(a \star b) \star c = a \star (b \star c)$
  - **Identity.** $a \star \mathrm{Id} = \mathrm{Id} \star a = a$
  - **Inversion.** $a \star a^{-1} = a^{-1} \star a = \mathrm{Id}$

  Example: $(G, \star) = (\mathbb{Z}_p, +)$ for modular addition, general group example includes symmetric group S_n and alternating group A_n
- **Data**: $\mathscr{D}_p = \{(g_1, g_2, y) : y = g_1 \star g_2,\ g_1, g_2 \in G\}$
- **Model**: Two-layer Neural Network

$$f_\theta(x) = \sum_{m=1}^{M} a_m \cdot \xi_m \cdot \sigma(\langle \theta_m^1, e_{g_1} \rangle + \langle \theta_m^2, e_{g_1} \rangle)$$





Group Representation

- **Group Representation**: Let $\mathrm{Irr}(G)$ denote the set of representation. Each $\rho \in \mathrm{Irr}(G)$ has dimension d_ρ s.t. $\rho(\cdot) : G \mapsto \mathbb{C}^{d_\rho \times d_\rho}$, satisfying
  - $\sum_{\rho \in \mathrm{Irr}(G)} d_\rho^2 = |G|$
  - $\rho(g_1 \star g_2) = \rho(g_1) \cdot \rho(g_2), \qquad \forall g_1, g_2 \in G$
  - There exists $\rho_{\mathrm{triv}}(g) = 1$ for all $g \in G$
  - For any ρ, there exists a dual representation $\rho^\vee(g) = \overline{\rho(g)}^\top$
  - $\left\{\sqrt{d_\rho}\, \rho_{ij}(\cdot) \in \mathbb{C}^{|G|} : \rho \in \mathrm{Irr}(G),\, i, j = 1, \ldots, d_\rho\right\}$

- **E.x., $\mathbb{Z}_p$**: Representation indexed by frequency
  - $\rho_k(g) = \exp(2\pi i k/p \cdot g) \in \mathbb{C},\, d_{\rho_k} = 1$
  - $k \in \{0, \ldots, p-1\}$ s.t. $\sum_{\rho \in \mathrm{Irr}(\mathbb{Z}_p)} d_\rho^2 = p$
  - $\rho(g_1 \star g_2) = \rho(g_1) \cdot \rho(g_2)$,
  - $\rho_0(g) = 1$ for all $g \in G$
  - dual representation $\rho_k^\vee(g) = \rho_{(p-k) \bmod p}(g)$
  - $\left\{\rho_k(\cdot) \in \mathbb{C}^{|G|} : \rho \in \mathrm{Irr}(\mathbb{Z}_p)\right\}$
  - Orthogonality verified ✅

Abelian Group: Extension to Modular Addition

- **Abelian Group**: commutative group $g_1 \star g_2 = g_2 \star g_1$
- **Fundamental Theorem**: For any abelian group G, we have

$$G \simeq \mathbb{Z}_{n_1} \oplus \ldots \oplus \mathbb{Z}_{n_d}, \qquad (g \star h)_\ell = (g_\ell + h_\ell) \bmod n_\ell,$$

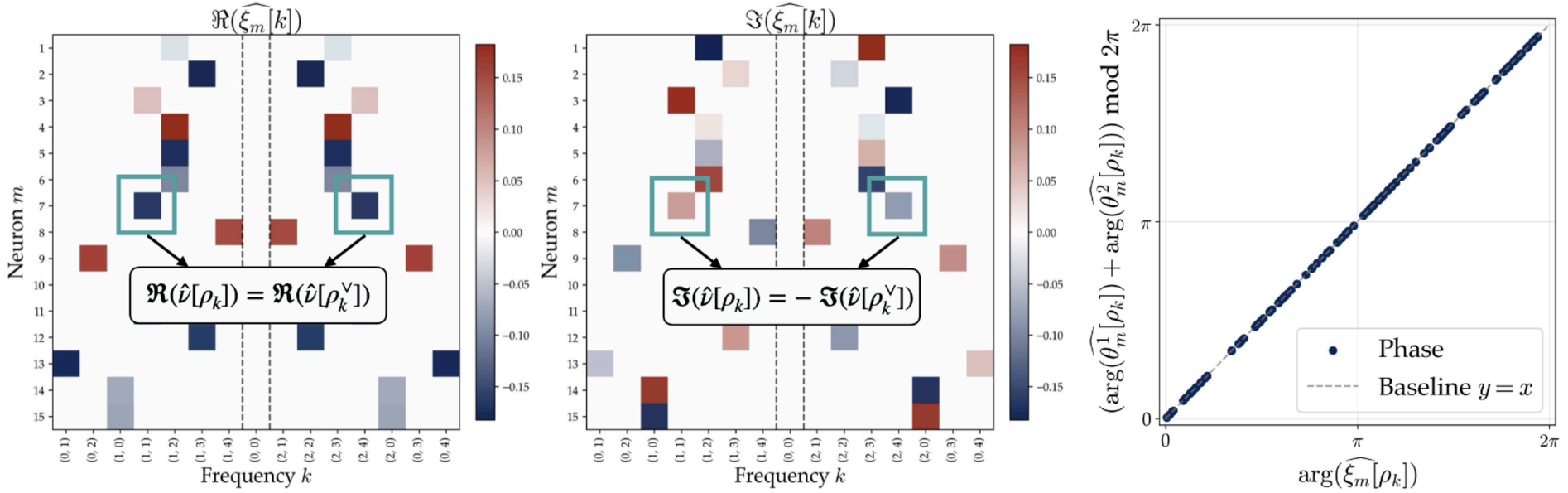
E.x., $G = \mathbb{Z}_3 \oplus \mathbb{Z}_5$ with $|G| = 15$

- **Representation**: $\rho_k(g) = \prod_{j=1}^{d} \exp\left(\frac{2\pi i k_j}{n_j} \cdot g_j\right) \in \mathbb{C}$ indexed by $k = (k_1, \ldots, k_d)$, all $d_\rho = 1$

- **Group DFT:** $\nu(g) = \sum_{k \in G} \hat{\nu}[\rho_k] \cdot \rho_k(g)$, where the Fourier coefficient $\hat{\nu}[\rho_k] = \frac{1}{|G|} \sum_{g \in G} \nu(g) \cdot \overline{\rho_k(g)} \in \mathbb{C}$

Fourier Coeff → Magnitude & Phase: $\hat{\nu}[\rho] = \alpha[\rho] \cdot \exp(i \cdot \phi[\rho])$, i.e., $\arg(\hat{\nu}[\rho]) := \phi[\rho]$

Abelian Group: Single Representation & Phase Alignment



(a) Visualization of the Single-Frequency Structure.

(b) Phase Alignment.

Findings:

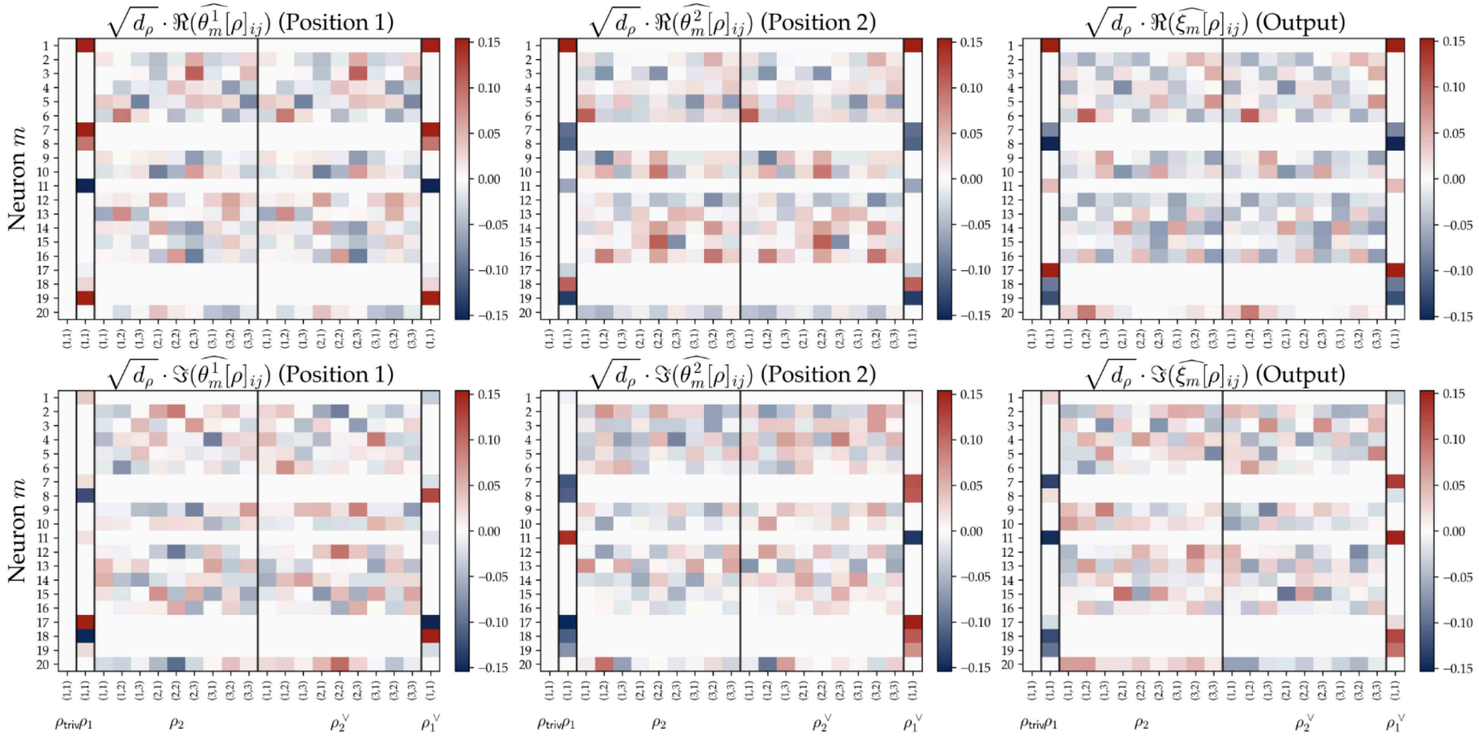
- Each neuron's DFT has only *one non-zero representation* (and its dual) that dominates, no zero frequency
- Phase aligned w.r.t. input and output: $\arg(\hat{\xi}_m[\rho_m]) = \left\{\arg(\hat{\theta}^1_m[\rho_m]) + \arg(\hat{\theta}^2_m[\rho_m])\right\} \bmod 2\pi$

Modular Addition v.s. Abelian Group

	Modular Addition on $\mathbb{Z}_p$	**Finite Abelian Group G**
Harmonic basis	Scalar frequency $k \in \mathbb{Z}_p$	Representation ρ_k, indexed by $k = (k_1, \ldots, k_d)$
Learned feature	$\cos\left(\frac{2\pi k g}{p} + \phi_m\right)$	$\Re\left(\hat{\nu}[\rho_k] \cdot \rho_k(g)\right)$
Fourier coefficient	Scalar $\widehat{\nu}[\rho_k] \in \mathbb{C}$	Scalar $\widehat{\nu}[\rho_k] \in \mathbb{C}$
Obs. 1: Sparsity	Single frequency k	Single representation $\rho_{\widetilde{k}}$
Obs. 2: Alignment	Doubled phase: $\psi_m = 2\phi_m \pmod{2\pi}$	$\arg\left(\widehat{\xi}_m[\rho_k]\right) = \arg\left(\widehat{\theta}^1_m[\rho_k]\right) + \arg\left(\widehat{\theta}^2_m[\rho_k]\right)$

General Group: Single Representation

- **Group DFT:** $\nu(g) = \sum_{\rho \in \mathrm{Irr}(G)} d_\rho \cdot \mathrm{tr}\big(\hat{\nu}[\rho]\rho(g)\big) = \sum_{\rho \in \mathrm{Irr}(G)} \sum_{i,j=1}^{d_\rho} d_\rho \cdot (\hat{\nu}[\rho])_{ij} \cdot \rho_{ji}(g) \in \mathbb{R}, \qquad \forall g \in G.$



- $G \simeq C_7 \rtimes C_3$ with $|G| = 21$
- **Irreps.** $\rho_{\mathrm{triv}}, \rho_1, \rho_1^\vee, \rho_2, \rho_2^\vee$ with dimension 1,1,1,3,3
- **Single Representation:** Only one non-trivial representation block active.

General Group: Rank-one Rotational Alignment

- **Rotational Alignment:** Note $\arg(\hat{\xi}_m[\rho_m]) = \left\{\arg(\hat{\theta}^1_m[\rho_m]) + \arg(\hat{\theta}^2_m[\rho_m])\right\} \bmod 2\pi$ is equivalent to

$$\hat{\xi}_m[\rho] \propto_+ \hat{\theta}^2_m[\rho] \cdot \hat{\theta}^1_m[\rho]$$

for scalar, i.e., Abelian cases. What about the general form? Answer is

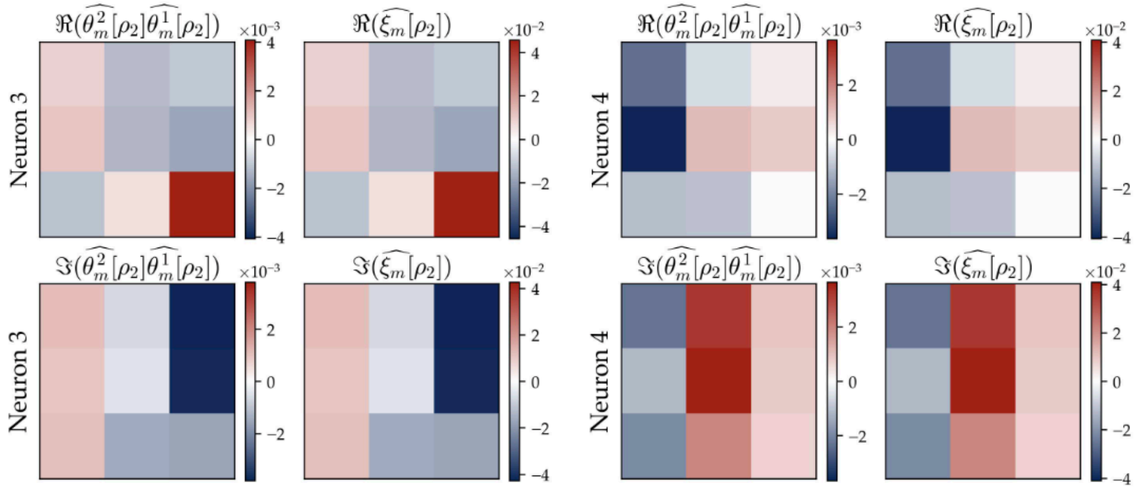
$$\hat{\xi}_m[\rho] \propto_+ \hat{\theta}^2_m[\rho]\, \hat{\theta}^1_m[\rho], \qquad \hat{\theta}^1_m[\rho] \propto_+ \left(\hat{\theta}^2_m[\rho]\right)^* \hat{\xi}_m[\rho], \qquad \hat{\theta}^2_m[\rho] \propto_+ \hat{\xi}_m[\rho]\left(\hat{\theta}^1_m[\rho]\right)^*.$$

- **Some intuitions of the nice form**
  - Phases satisfies the composition rule

    1) matrix multiplication ⇔ group action,

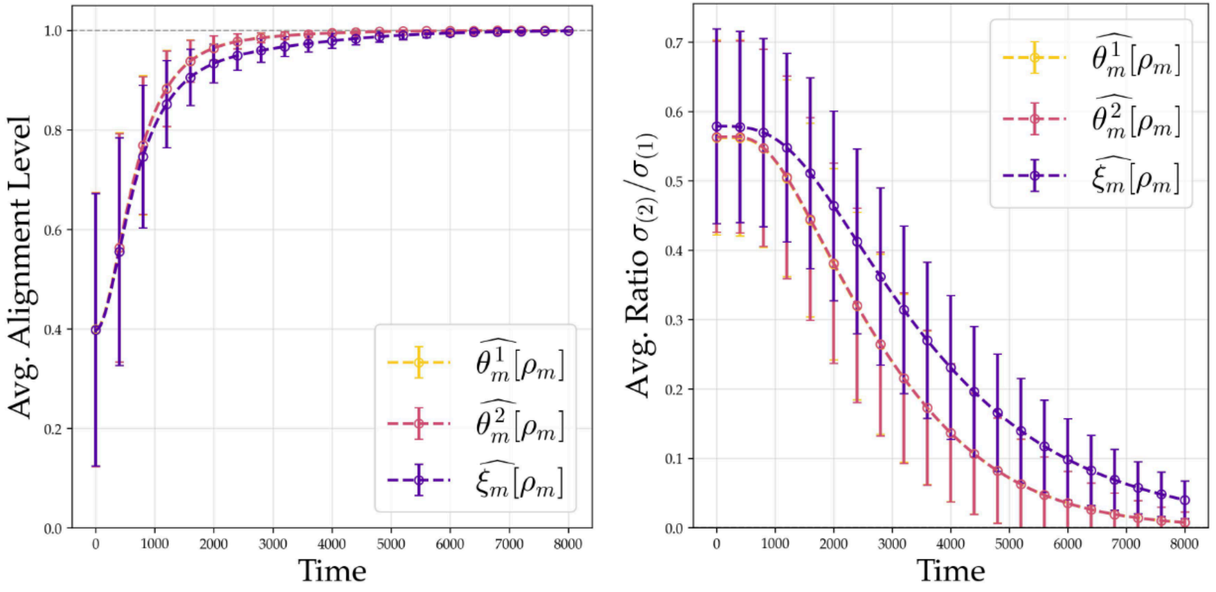
    2) Invertibility: Hermitian conjugate

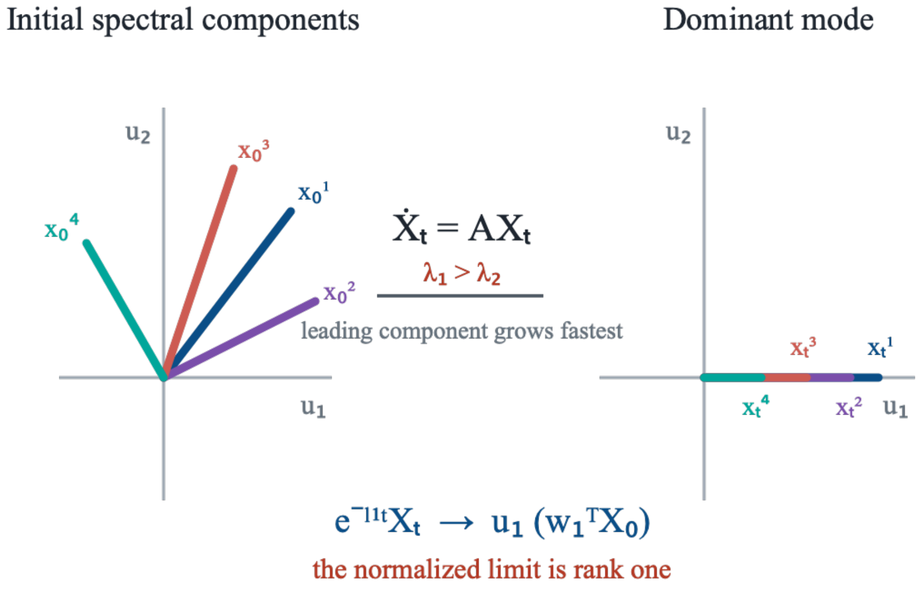
    3) Associability: 3 equations



General Group: Rank-one Rotational Alignment

- **Rank-one Fourier Matrix:** Despite the alignment, the model learns a **low-rank structure** in **representation space**, i.e., $\mathrm{rank}(\hat{\theta}^1_m[\rho]) = \mathrm{rank}(\hat{\theta}^2_m[\rho]) = \mathrm{rank}(\hat{\xi}_m[\rho]) = 1$.





$$\mathrm{dist}_{\mathrm{al}}(C_1, C_2) = |\langle \mathrm{vec}(C_1), \mathrm{vec}(C_2)\rangle_{\mathbb{C}}| / (\|C_1\|_{\mathrm{F}} \cdot \|C_2\|_{\mathrm{F}})$$

General Group: Summary

Informal Theorem. *Consider the gradient flow under the population risk. We write* $\nu \in \{\theta_m^1, \theta_m^2, \xi_m\}$ *and view each* ν *as a function* $G \to \mathbb{R}$*. Then, for every neuron* m*, with probability one over the continuous random initialization, there exists a non-trivial irreducible representation* $\check{\rho}_m$ *such that, as* $t \to \infty$*, the gradient flow converges to a state satisfying the following:*

(i) ***(Single Representation).*** *In the Fourier domain, let* $\widehat{\nu}[\rho]$ *denote the Fourier coefficient that measures the component of* ν *along the representation* ρ*. Each neuron keeps only* $\check{\rho}_m$ *and its conjugate* $\check{\rho}_m^\vee$*:*

$$\widehat{\nu}[\rho] \to \mathbf{0}_{d_\rho \times d_\rho}, \qquad \forall \rho \notin \{\check{\rho}_m, \check{\rho}_m^\vee\},\ \nu \in \{\theta_m^1, \theta_m^2, \xi_m\},$$

where $d_\rho \in \mathbb{N}^+$ *represents the dimension of the corresponding representation.*

(ii) ***(Rank-one Rotational Alignment).*** *On the surviving representation* $\rho \in \{\check{\rho}_m, \check{\rho}_m^\vee\}$*, the three Fourier coefficients (as a matrix in general) become rank-one:* $\operatorname{rank}(\widehat{\theta_m^1}[\rho]) = \operatorname{rank}(\widehat{\theta_m^2}[\rho]) = \operatorname{rank}(\widehat{\xi_m}[\rho]) = 1$*. Moreover, these coefficients have some alignment across layers, i.e., there exists* $\lambda > 0$ *such that*

$$\widehat{\xi_m}[\rho] = \lambda \cdot \widehat{\theta_m^2}[\rho]\, \widehat{\theta_m^1}[\rho], \qquad \widehat{\theta_m^1}[\rho] = \lambda \cdot \big(\widehat{\theta_m^2}[\rho]\big)^* \widehat{\xi_m}[\rho], \qquad \widehat{\theta_m^2}[\rho] = \lambda \cdot \widehat{\xi_m}[\rho]\, \big(\widehat{\theta_m^1}[\rho]\big)^*,$$

where $(\cdot)^*$ *denotes the Hermitian adjoint. In other words, each of* $\widehat{\theta_m^1}[\rho]$*,* $\widehat{\theta_m^2}[\rho]$ *and* $\widehat{\xi_m}[\rho]$ *is positively proportional to the product formed by the other two coefficients in a rotational order.*

General Group: Theoretical Insights

- **Minimizing CE Loss v.s Energy Maximization.** Under small initialization and some regularity conditions, the training is implicitly maximizing an **energy functional**

$$\Omega_m = \sum_{\rho \in \mathrm{Irr}(G)_{\neq 1}} d_\rho \cdot \mathrm{tr}\left((\widehat{\xi_m}[\rho])^* \widehat{\theta_m^2}[\rho] \widehat{\theta_m^1}[\rho] \right) \in \mathbb{R}$$

The learned patterns, i.e., single representation + alignment, are *non-measure-zero* local minima.

- **Dynamics.** Under (projected) gradient flow, the **training dynamics** can be captured by

$$\partial_t \hat{\theta}_m^1[\rho] = \frac{2a|G|}{M} \cdot (\hat{\theta}_m^2[\rho])^* \hat{\xi}_m[\rho] - \frac{2a|G|^2}{M} \cdot \Omega_m \cdot \hat{\theta}_m^1[\rho]$$

The evolution equations for $\hat{\theta}_m^2[\rho]$ and $\hat{\xi}_m[\rho]$ have the same structure.

Takeaways on Representation Learning

- Neural network **can** learn the representation of group composition naturally from gradient training instead of *interpolation-based memorization*. (Important for generalization in train-test split case)
- Model learns exactly the data representation in this case!
  - **Single Representation:** Each neuron's weights encode a single, specific group representation.
  - **Layer-wise Alignment:** Within each neuron, weights from different layers are rotationally aligned in the shared representation space following the composition rule.
  - **Rank-one in Representation Space:** the Fourier coefficient of learned representation is of rank one regardless of the dimension of representation. (Related to intrinsic dimension of model)

Mechanism Sketch: Majority Vote

- For modular addition, single neuron computes

$$f_m(x,y)[j] \propto \cos(w_k(x-y)/2) \cdot \left(\cos(w_k(x+y-j)) + \text{noise}_{k,\phi}\right)$$

The red term is the main signal term and the blue one is biased noise term.

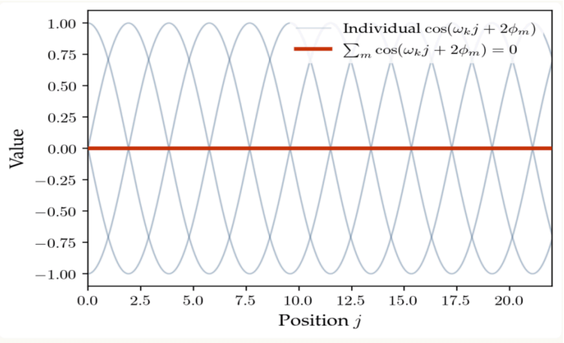
- Under standard *random initialization*, e.g., gaussian, the model learns a **diversified solution** (proved by mean-field analysis) s.t.

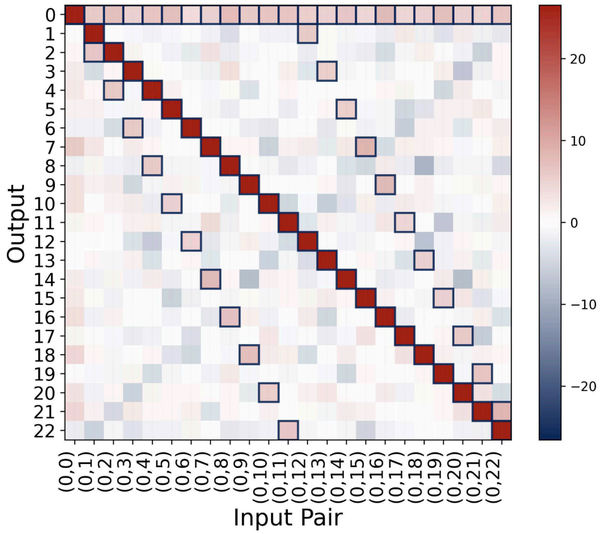
$$\phi_m, \psi_m \sim \text{Unif}(0,2\pi), \qquad k \sim \text{Unif}(\{1,\ldots,(p-1)/2\})$$

and thus

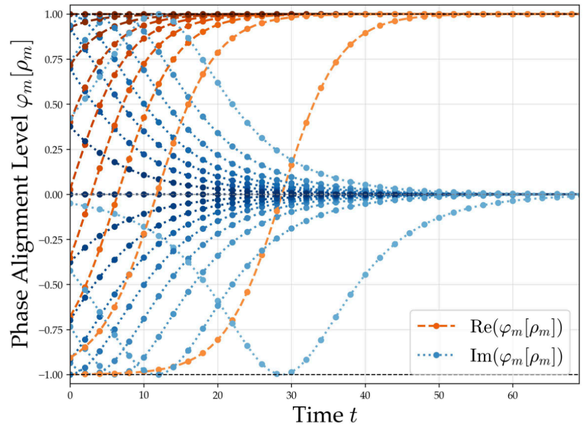
$$f(x,y)[j] \propto -1 + p/2 \cdot \mathbf{1}(x+y \bmod p = j) + p/4 \cdot \sum_{z \in \{x,y\}} \mathbf{1}(z \bmod p = j)$$

By **voting across neurons**, bias are eliminated and we get the correct answer!

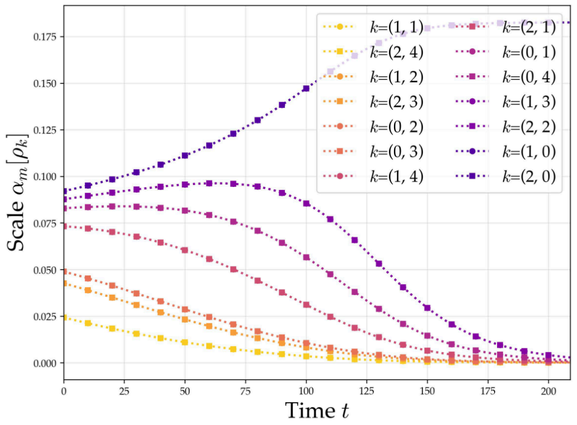




Dynamics Sketch: Lottery Ticket Mechanism



(a) Training Dynamics of Phase Alignment.



(b) Representation Competition within Neuron.

The learning dynamics is governed by

$$\alpha_\theta[\rho], \quad \alpha_\xi[\rho], \quad \varphi[\rho] = \overline{\phi_\xi[\rho]} \cdot \phi_\theta[\rho]^2$$

which corresponds to the magnitudes and the phase alignment level:

$$\partial_t \alpha_\xi[\rho] = \frac{2a\,|G|}{M} \cdot \alpha_\theta[\rho]^2 \cdot \Re\big(\varphi_m[\rho]\big) - \frac{2a\,|G|^2}{M} \cdot \Omega_m \cdot \alpha_\xi[\rho]$$

and

$$\partial_t \varphi_m[\rho] = -\frac{2a\,|G|}{M} \cdot \left(2\alpha_{\xi,m}[\rho] + \frac{\alpha_{\theta,m}[\rho]^2}{\alpha_{\xi,m}[\rho]}\right) \cdot \Im\big(\varphi_m[\rho]\big) \cdot \varphi_m[\rho] \cdot i$$

- Phases align first, then **accelerate** scale growth
- The fastest-growing representation becomes dominant.

⇒ **LTM:** The winner is determined at initialization: the representation with the best phase alignment and strongest initial target-scale component grows fastest

Thank You! Happy to take questions:)