Harmonic Loss Trains Interpretable AI Models

David D. Baek [*1] **Ziming Liu** [*1] **Riya Tyagi** [1] **Max Tegmark** [1]

Abstract

In this paper, we introduce **harmonic loss** as an alternative to the standard cross-entropy loss for training neural networks and large language models (LLMs). Harmonic loss enables improved interpretability and faster convergence, owing to its scale invariance and finite convergence point by design, which can be interpreted as a class center. We first validate the performance of harmonic models across algorithmic, vision, and language datasets. Through extensive experiments, we demonstrate that models trained with harmonic loss outperform standard models by: (a) enhancing interpretability, (b) requiring less data for generalization, and (c) reducing grokking. Moreover, we compare a GPT-2 model trained with harmonic loss to the standard GPT-2, illustrating that the harmonic model develops more interpretable representations. Looking forward, we believe harmonic loss has the potential to become a valuable tool in domains with limited data availability or in high-stakes applications where interpretability and reliability are paramount, paving the way for more robust and efficient neural network models.

1. Introduction

In recent years, machine learning models have gained significant popularity, profoundly impacting various aspects of daily life. Consequently, it has become increasingly important to thoroughly understand the behavior of neural networks. One particularly intriguing characteristic of neural networks is their ability to generalize – empirical evidence shows that neural networks can perform well on unseen data not explicitly encountered during training (Novak et al., 2018). This remarkable ability stems from the networks' capacity to learn generalizable representations and algorithms through training.

However, current models face three key challenges when it comes to generalization:

Ehhh not entirely true, it's just difficult

(1) Lack of interpretability: Neural networks often lack interpretability. This opacity limits our understanding of how these models arrive at their conclusions, which is a critical issue in high-stakes applications like healthcare, finance, and autonomous systems. While mutiple research efforts have advanced our insight into the inner workings of LLMs (Bereska & Gavves, 2024), we are still far from fully explaining their outputs. Ultimately, we believe it is crucial to design systems that are interpretable by design. Without interpretability, it becomes challenging to diagnose errors, ensure fairness, or build trust in a model's decisions.

(2) Low data efficiency: Generalization often requires vast and diverse training data. This raises a critical question: can models generalize effectively with less data? This issue is especially relevant in domains where data availability is scarce, such as rare disease diagnosis, low-resource language processing, or specialized scientific fields like materials science or drug discovery. Previous approaches for improving neural network generalization include efficient data sampling (Li et al., 2024a) and modifications to the training procedure to accelerate training (Wang et al., 2024). However, these methods focus on optimizing existing training procedures rather than addressing the core issues in model design.

(3) Delayed generalization (grokking): Models sometimes experience a phenomenon known as "grokking," (Power et al., 2022; Liu et al., 2021) where there is a noticeable delay between the convergence of the training loss and the convergence of the test loss. This gap is problematic because: (i) it complicates determining the optimal point to stop training in order to achieve generalization, and (ii) it necessitates extended computation time and resources to continue training until grokking occurs.

As the saying goes, "The devil is in the details." We attribute these three challenges in part to the widespread use of cross-entropy loss (for classification) and propose **harmonic loss** as an alternative. Harmonic loss has two desirable mathematical properties that enable faster convergence and improved interpretability: (1) scale invariance, and (2) a finite convergence point, which can be interpreted as a class center. Through a comprehensive set of experiments, we demonstrate that models trained with harmonic loss outperform standard models in terms of reducing grokking, requiring

[*]Equal contribution [1]Massachusetts Institute of Technology, Cambridge, MA, USA. Correspondence to: David D. Baek <dbaek@mit.edu>.

less data for generalization, and enhancing interpretability. Furthermore, we compare a GPT-2 model trained with harmonic loss to the standard GPT-2 and show that the harmonic model develops more interpretable representations.

The remainder of this paper is organized as follows: In Section 2, we review the relevant literature. Section 3 introduces the principles underlying harmonic loss and explains why it is preferable to cross-entropy loss in terms of generalization capabilities. Section 4 details a comprehensive set of experiments on algorithmic datasets, demonstrating that models trained with harmonic loss consistently outperform standard models. In Section 5, we demonstrate the performance of harmonic models on the vision task of MNIST digit classification. In Section 6, we extend our analysis to large models, illustrating that the advantages of harmonic loss also hold at scale. Finally, we conclude the paper in Section 7.

2. Related Works

Representations: In this paper, we aim to improve the interpretability of neural network representations. Numerous studies have shown that LLMs can form conceptual representations across spatial (Gurnee & Tegmark, 2023), temporal (Li et al., 2021), and color domains (Abdou et al., 2021). The structure of such representations includes one-dimensional concepts (Gurnee & Tegmark, 2023; Marks & Tegmark, 2023; Heinzerling & Inui, 2024; Park et al., 2024b), as well as multi-dimensional representations such as lattices (Michaud et al., 2024; Li et al., 2024c) and circles (Liu et al., 2022a; Engels et al., 2024). Recent works have also studied the representations developed during inference-time (Park et al., 2024a). While the structure of these representations often correlates with certain geometric patterns, significant unexplained variance frequently remains, complicating interpretability.

Mechanistic Interpretability: While increasing the number of parameters and the amount of data samples used to train neural networks has enhanced their capabilities, it has also made mechanistically interpreting these models more challenging. This line of work has been explored from two major directions: the circuit level (Michaud et al., 2024; Olah et al., 2020a; Olsson et al., 2022; Templeton et al., 2024), which aims to identify a submodule within an LLM responsible for a specific ability, and the representation level (Liu et al., 2022b; Ding et al., 2024; Zhong et al., 2024). Some works have made progress by designing interpretable systems through the decomposition of models into smaller modules (Olah et al., 2020a; Liu et al., 2023).

Loss Functions: Previous research has shown that loss functions can influence how a model learns to represent data, affecting its abilities in unique ways (Li et al., 2024b; Bosco et al., 2024). Novel loss functions have improved performance on specific tasks, though they often reduce ability in different settings (Bommidi et al., 2023; Seber, 2024). For instance, focal loss, dice loss, and Tversky loss have proven effective for image segmentation (Sudre et al., 2017; Demir et al., 2023; Salehi et al., 2017), but only focal loss is also effective for object detection (Lin, 2017). Luo et al. (2021) found that smoothly approximating non-smooth Hinge loss in Support Vector Machines (SVMs) improved the convergence rate of optimization.

3. Harmonic Loss

We first review cross-entropy loss and present the harmonic loss, visualized in Figure 1 (a). Denote the unembedding matrix as $\boldsymbol{W} \in \mathbb{R}^{N\times V}$ (N is the embedding dimension, V is the vocabulary size), and the penultimate representation (the representation prior to the unembedding matrix) as $\boldsymbol{x} \in \mathbb{R}^N$.

Cross-entropy loss: Logits $\boldsymbol{y}$ are defined as the matrix-vector multiplication, i.e., $\boldsymbol{y} = \boldsymbol{W}^T\boldsymbol{x} \in \mathbb{R}^V$ (ignoring biases), or

$$y_i = \boldsymbol{w}_i \cdot \boldsymbol{x}, \tag{1}$$

where $\boldsymbol{w}_i$ is the i^{th} column of $\boldsymbol{W}$. Probability $\boldsymbol{p}$ can be obtained by applying softmax to $\boldsymbol{y}$, i.e.,

$$p_i = \text{SoftMax}(\boldsymbol{y})_i \equiv \frac{\exp(y_i)}{\sum_j \exp(y_j)}. \tag{2}$$

Suppose the real class label is c, then loss $\ell = -\log p_c$. For notational simplicity, we call a linear layer combined with the cross-entropy loss a *cross-entropy layer*.

Harmonic loss: The *harmonic logit* $\boldsymbol{d}$ is the l_2 distance between $\boldsymbol{w}_i$ and $\boldsymbol{x}$, i.e.,

$$d_i = ||\boldsymbol{w}_i - \boldsymbol{x}||_2. \tag{3}$$

(distance)

We interpret $\boldsymbol{w}_i$ as keys and $\boldsymbol{x}$ as a query, so smaller d_i means a higher probability of p_i. We define *harmonic max* (*harmax*) as

Smaller distances are larger

$$p_i = \text{HarMax}(\boldsymbol{d})_i \equiv \frac{1/d_i^n}{\sum_j 1/d_j^n}, \tag{4}$$

where n (*harmonic exponent*) is a hyperparameter that controls the heavy-tailedness of the probability distribution. If the true class label is c, then loss $\ell = -\log p_c$. For notational simplicity, we call a layer combined with the harmonic loss the *harmonic layer*. Since the last step of both losses is the same ($\ell = -\log p$), comparing their values is meaningful. CE and harmonic They only differ in the ways of computing probabilities from representations [1].

[1] Note that when we say "cross-entropy loss," we do not only refer to $\ell = -\log p$, but rather refer to the whole pipeline including penultimate representation, logit, probability, and loss.

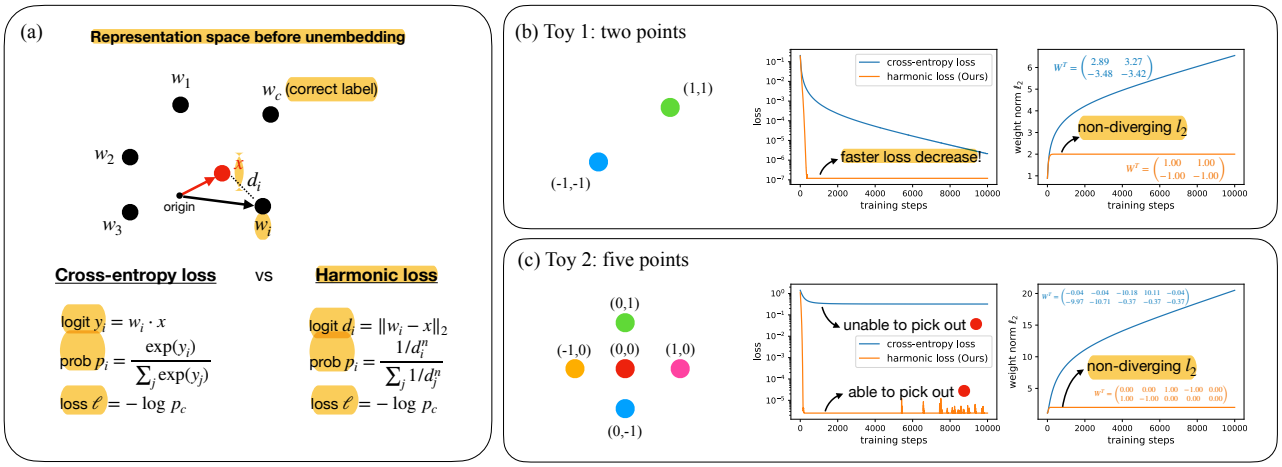

Figure 1. Cross-entropy loss versus harmonic loss (ours). (a) Definitions. Cross-entropy loss leverages the inner product as the similarity metric, whereas the harmonic loss uses Euclidean distance. (b) Toy case 1 with two points (classes). Both the harmonic loss and the l_2 weight norm converge faster for the harmonic loss. (c) Toy case 2 with five points (classes). Harmonic loss can pick out the red point in the middle. By contrast, the cross-entropy loss cannot, since the red point is not linearly separable from other points. The weight matrices are also more interpretable with harmonic loss than with cross-entropy loss.

A reasonable choice of n is $n \sim \sqrt{D}$, where D represents the intrinsic dimensionality of the underlying data. In LLMs, D could be approximated as $D \approx d_{\text{embed}}$, where d_{embed} is the embedding dimension. This approximation arises from considering an embedding initialized from a D-dimensional Gaussian distribution. The squared distance between two points, normalized by the number of dimensions D, is on the order of $1 \pm O(1/\sqrt{D})$. To ensure that the harmonic distance $\left[1 \pm O(1/\sqrt{D})\right]^n$ remains constant as we scale D, we require $n \sim \sqrt{D}$, since $\lim_{x\to\infty}(1 + x^{-1})^x = e$.

Toy cases: To provide intuition about what advantages the harmonic loss has over the cross-entropy loss, we consider two toy cases in 2D, as shown in Figure 1 (b)(c). In each toy case, we train the cross-entropy layer and the harmonic layer with the Adam optimizer. **Toy case 1**: $\boldsymbol{x}_1 = (1, 1)$ and $\boldsymbol{x}_2 = (-1, -1)$ belong to two different classes. The harmonic layer produces a faster loss decrease, because the harmonic loss only requires $d_i \to 0$ (converging point is finite) to get $p_i \to 1$. By contrast, cross-entropy loss requires $y_i \to \infty$ (converging point is infinite) to get $p_i \to 1$. The harmonic loss already produces a l_2 weight norm that plateaus to a constant, while the cross-entropy loss leads to increasing l_2, diverging towards infinity. **Toy case 2**: There are 5 points in 2D, each of which belong to a different class. In particular, the red point $(0, 0)$ is surrounded by the other four points, i.e., cannot be linearly separated. The cross-entropy layer indeed cannot perform well on this task, manifested by a high loss plateau. By contrast, the harmonic layer can drive the loss down to machine precision. Similar to case 1, the harmonic layer has a plateauing l_2 while the cross-entropy layer has an ever-growing l_2. We also observe that the weights of the harmonic layer correspond to $\boldsymbol{x}$, which is more interpretable than the weights of the cross-entropy layer.

Benefits of harmonic loss: From these two toy cases, we understand the advantages of harmonic loss: (1) *nonlinear separability*: in case 2, the red dot can be classified correctly even though it is not linearly separable. (2) *fast convergence*: The fact that the converging point is finite leads both to faster loss decay, and plateauing (non-diverging) l_2. (3) *scale invariance*: Harmonic loss is scale-invariant, i.e., $d_i \to \alpha d_i$ leaves p_i (hence loss) invariant, whereas $y_i \to \alpha y_i$ would produce a different cross-entropy loss. (4) *interpretability*: the weight vectors correspond to class centers.

Interesting point

The rest of the paper explores the use of harmonic loss in various applications: algorithmic tasks in Section 4, MNIST in Section 5, and language modeling in Section 6.

4. Algorithmic Datasets

Algorithmic tasks are good benchmarks for network interpretability since they are well-defined mathematically. However, training neural networks on these tasks is non-trivial due to the observation of grokking (delayed generalization) (Power et al., 2022) and the existence of multiple algorithms (Zhong et al., 2024), etc. We will show that harmonic models can learn better representations, are more data-efficient, and experience less grokking.

4.1. Models and Datasets

Models: We compare four models:

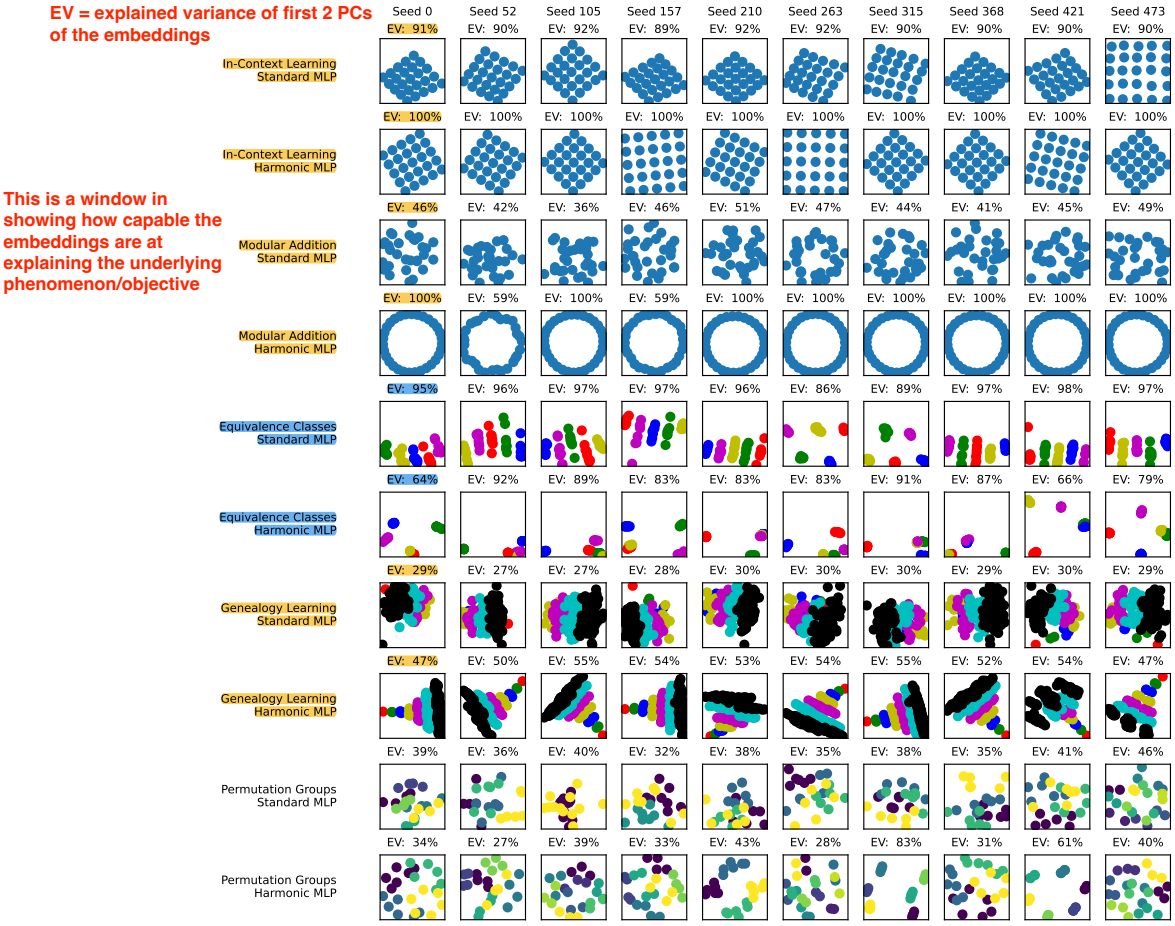

Figure 2. Visualization of the top two principal components of the embeddings in synthetic experiments. The title of each subplot shows the explained variance by the first two principal components. Each row corresponds to a pair of a dataset and a model, while each column represents the embeddings from different training runs with varying seeds. Groups of consecutive two rows belong to the same dataset, with models arranged in the order: {Standard MLP, Harmonic MLP}. The datasets are ordered as follows: {In-Context Learning, Genealogy Learning, Equivalence Classes, Modular Addition, and Permutation Groups}. X and Y axis spans are equal.

1. **Standard MLP**: Tokens are embedded into 16-dimensional embeddings, which are then concatenated and used as the input. The model consists of two hidden layers with widths of 100 and 16, respectively. The SiLU activation function is used.

2. **Standard Transformer**: Tokens are embedded into a 16-dimensional embedding, with a learnable positional embedding added. The input passes through two transformer decoder layers, each comprising two attention heads and an MLP with a hidden dimension of 64.

3. **Harmonic MLP**: Standard MLP with an harmonic unembedding layer of exponent $n = 1$.

4. **Harmonic Transformer**: Standard Transformer with an harmonic unembedding layer of exponent $n = 1$.

We trained the MLP models for 7000 epochs and the transformers for 10000 epochs. For all four models, we used the AdamW optimizer with a learning rate of 2×10^{-3}, a weight decay of 10^{-2}, and an L_2 regularization on the embeddings with strength 0.01.

Datasets: We trained the four models above using the following five datasets, and analyzed their performance as well as the resulting representations:

1. **In-Context Learning**: In a 5×5 integer lattice, given three points on the lattice, the model is trained to predict the fourth point that would form a parallelogram with the others. This task exemplifies in-context reasoning in LLMs, mirroring the classic *man:woman::king:queen* analogy by requiring the model to complete the relational pattern such as 'man

is to woman as king is to queen' based on the given context.

2. **Modular Addition**: Given two integers x and y, the model is trained to predict $(x + y)$ mod 31.

3. **Equivalence Classes**: Given two integers $0 \leq x, y < 40$, the model is trained to predict if $x \equiv y$ mod 5.

4. **Genealogy Learning**: In a complete binary tree with 127 nodes, given a subject and a relation, the model is trained to predict the corresponding object. The relation can be one of the following: parent, grandparent, or sibling.

5. **Permutation Composition**: Given two permutations x and y in S_4, the model is trained to predict $x \circ y$. On this dataset, we trained standard and harmonic transformers with an L_2 regularization of 0.005, as we found this configuration led to more complete training.

4.2. Representation Faithfulness

Figure 2 shows the plot of the top two principal components of the models' embeddings for MLP tasks. A complete visualization is available in Figure 2. We show the embedding visualization of transformers in Appendix A. Overall, harmonic loss representations are cleaner and more organized than their cross-entropy counterparts. We found near-perfect circle representations for the modular addition task, a clear tower-like structure for tree learning, and neat clusters for permutation composition.

We examine the representations task by task:

(1) In-context learning: We observe that the representations obtained from standard models are either imperfect lattices or exhibit unexplained variance in higher dimensions, whereas harmonic models almost always perfectly (100%) recover the underlying 2D lattice structure regardless of the random seed.

(2) Modular addition: Harmonic MLPs consistently recover a purely 2D circular representation in almost all runs, whereas the standard MLP often fails to identify the circular structure. While the harmonic transformer has a similar success rate to the standard transformer in constructing circles, the explained variance captured by the first two principal components is generally much higher, indicating that harmonic models discover more compact representations with fewer uninterpretable components.

(3) Equivalence classes: Both standard and harmonic models are able to identify the underlying groups with high variance. However, we note that standard models' representation tends to be more "elongated", or not *completely* grouped, compared to its harmonic counterpart. This could be attributed to the fact that cross-entropy loss does not have an incentive to reduce irrelevant variations to zero.

(4) Genealogy learning: Harmonic MLP is the only model that successfully recovers the underlying tree representation.

(5) Permutation composition: The harmonic MLP generally produces better-separated clusters. A particularly clean representation that appears multiple times contains 6 clusters of 4 permutations, where each cluster is a coset of the subgroup $\langle e, (12)(34), (13)(24), (14)(23) \rangle$ or one of its conjugates. In the harmonic transformer, permutations commonly organize into 4 clusters that are cosets of $\langle e, (13), (14), (34), (134), (143) \rangle$ or one of its conjugates, subgroups isomorphic to S_3 (one element, in this case 2, never permutes).

Figure 3(a) further demonstrates that harmonic representations tend to be more compact than standard models, with fewer uninterpretable components. In particular, harmonic models trained for in-context learning achieve 100% explained variance using only the first two principal components.

4.3. Data Efficiency in Training

Humans can learn from surprisingly few samples, but neural networks usually require much more data to learn. As a result, we want models that can learn efficiently from data. Figure 3(b) shows the test accuracy as a function of train data fraction for our synthetic experiments, indicating how much data is necessary in order for the model to be generalizable. We observe that harmonic models require comparable or much less amount of data to generalize, compared to their cross-entropy counterparts. Such improvement is especially notable for in-context learning, where harmonic models generalize nearly immediately.

4.4. Reduced Grokking

Grokking refers to the phenomenon of delayed generalization (Power et al., 2022): for example, it takes 10^3 steps to reach perfect accuracy on the training data, but it takes 10^5 steps to generalize to the test data. Grokking is a pathological phenomenon that we want to avoid (Liu et al., 2022b). We find that harmonic loss overall reduces grokking, as seen in Figure 3(c). Points on the $y = x$ line represent models which trained without grokking, with train and test accuracy improving together. This improvement is particularly evident in learning modular addition and permutation composition: while the standard MLP exhibits severe grokking, most data points for the harmonic MLP lie much closer to the $y = x$ line.

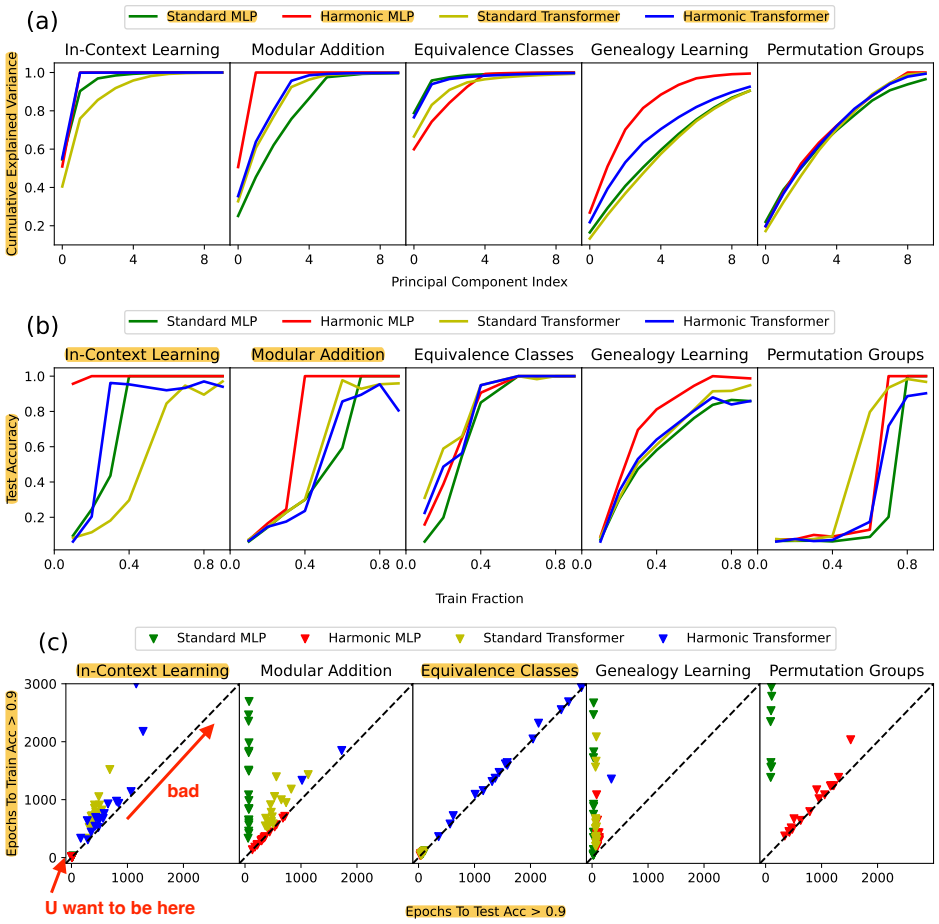

Figure 3. (a) Cumulative explained variance as a function of principal components (median over 20 seeds). Harmonic representations are more compact than standard counterparts. (b) Test Accuracy as a function of Training Fraction. Harmonic models generalize faster with less data than standard counterparts. (c) Epochs to Test Accuracy > 0.9 vs Epochs to Train Accuracy > 0.9 for 20 consecutive times. $y = x$ line represents no grokking, where train and test accuracy improve simultaneously. Points closer to the y-axis indicate a greater degree of grokking. Results from 20 different random seeds are plotted, and the runs that were not able to achieve 90% accuracy were omitted.

Seems cherry picked since it's the best for the Harmonic Loss

4.5. Case Study: Modular Addition

In this section, we study modular addition as a case study and analyze why the harmonic MLP encourages more interpretable representations and better generalization compared to the standard MLP. The standard MLP trained for modular addition without weight decay often fails to generalize, as shown in Figure 4. Generalization is only achieved with the addition of strong weight decay; however, (a) significant grokking occurs, as depicted in Figure 4, and (b) while the first two principal components form an approximate circle, they explain far less than the total variance, leaving significant unexplained variance. In contrast, the harmonic model trained for modular addition generalizes quickly without grokking. Furthermore, the embedding forms a perfect circle, as shown in Figure 4.

The better formation of a circle and improved generalization in harmonic MLP can be attributed to the properties of harmonic loss, as explained in Section 3. To drive the probability to 1, the standard cross-entropy loss requires driving the representation to infinity – *i.e.*, making the logit infinite. In contrast, harmonic loss achieves this by driving the harmonic logit to zero, which is easily accomplished by learning $\boldsymbol{w}_i = \boldsymbol{x}$ in Equation 3. The existence of such a finite converging point results in (a) faster convergence, (b) better generalization, and (c) more interpretable representations.

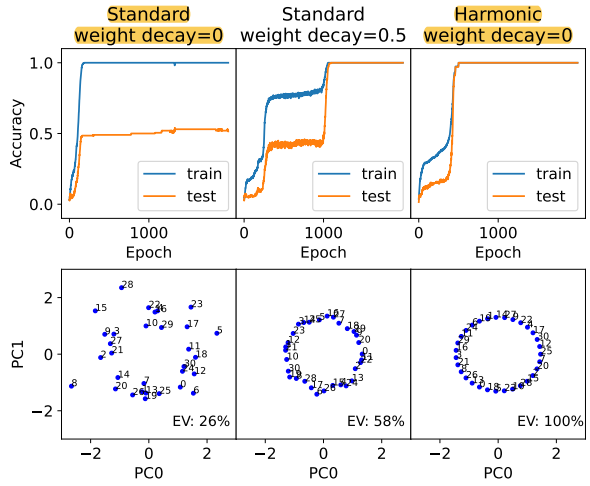

Figure 4. Case study on modular addition. Standard MLP trained for modular addition without weight decay often fails to generalize. Generalization is only achieved with the addition of strong weight decay; however, (a) significant grokking occurs, and (b) while the first two principal components form an approximate circle, they explain far less than the total variance. In contrast, the harmonic model trained for modular addition generalizes quickly without grokking. Moreover, the embedding forms a perfect 2D circle. EV in the plot represents the explained variance by the first two principal components of the embedding.

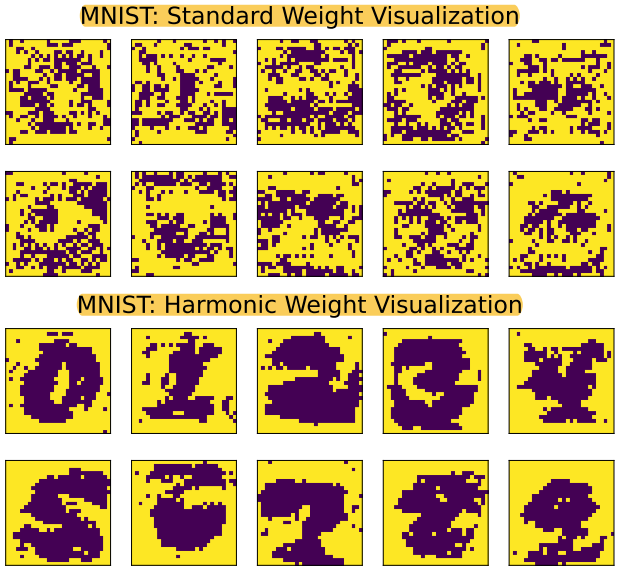

Figure 5. Visualization of model weights trained for MNIST. Yellow cells show values less than 0.01. Both models achieved $\approx 92.5\%$ test accuracy.

5. MNIST Experiments

For vision tasks, convolutional neural networks are shown to be (at least somewhat) interpretable by demonstrating "edge detectors", "wheel detectors", etc. (Olah et al., 2020b). However, fully connected networks do not appear to be interpretable (e.g., see Figure 5 top). In this section, we demonstrate that the harmonic loss can lead to a more interpretable network for the MNIST dataset.

As a proof of concept, we choose our setup to be as simple as possible. Thus, we compare 1-layer neural networks trained using cross-entropy loss and harmonic loss. The input images are first flattened and passed through a 784×10 linear layer to obtain the logits. The models were trained with a batch size of 64, a learning rate of 0.001, and for 10 epochs, achieving a 92.50% test accuracy for cross-entropy loss and 92.49% test accuracy for harmonic loss. We now analyze the weights learned by the neural network.

Figure 5 verifies that the weights in the model trained with harmonic loss are highly interpretable. Consistent with its core principle, we observe that the weights align with the class centers, which, in this case, correspond to images representing each number. Additionally, most peripheral pixels have weights that are almost exactly zero, in contrast to the model trained with cross-entropy loss. The latter, by design, lacks an incentive to reduce the irrelevant background weights to zero.

6. GPT2 Experiments

Many mechanistic interpretability works have been dedicated to understanding large language models. For example, probing and attribution methods are good post hoc analysis tools. Despite their (partial) success, these tools are not creating interpretable models in the first place but are trying to find needles in the haystack. We argue that it would be nicer if we could pre-train the language models to be more interpretable. By using harmonic loss in training, we can produce a language model that can "grow" crystal-like representations, while having comparable performance with a standard one (trained with the cross-entropy loss).

We pre-train a GPT-2 small model (128M, based on NanoGPT) on OpenWebText. The embedding matrix and the unembedding matrix are tied (share the same weights). We use 8 V100 GPUs, choose block size 1024, batch size 480 blocks. We use the Adam Optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.95$. For the harmonic loss, we choose $n = \sqrt{768} \approx 28$, following the discussion on harmonic exponent in Section 3. For standard (harmonic) GPT, we use a linear warmup learning rate schedule for 2k (1k) steps to maximum learning rate 6×10^{-4} (6×10^{-3}), and a cosine decay schedule from 2k to 10k, ending at lr 3×10^{-5} (3×10^{-4}). As shown in Figure 6 top left, Harmonic GPT shows faster converging initially (partially due to larger learning rates), and converges to similar performance in the end (at 10k

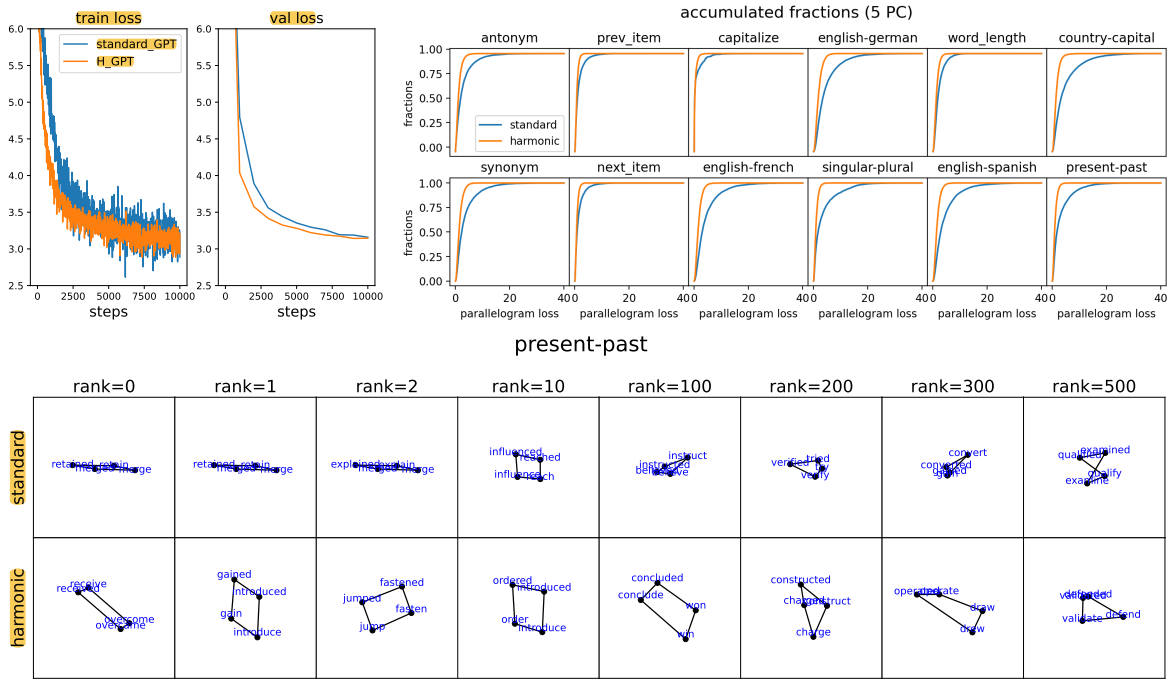

Figure 6. GPT2 experiments, trained on OpenWebText for 10k steps. (Top left) loss curves. Harmonic GPT achieves a slightly lower loss compared to the standard GPT. (Top right) accumulated distribution function with respect to parallelogram loss, for twelve function-vector tasks. The Harmonic GPT consistently shows lower parallelogram losses (i.e., better parallelograms). (Bottom) Parallelograms (1st and 2nd principal component) with quality ranked in descending order from left to right. The Harmonic GPT tends to produce parallelograms that are more 'rectangular', while standard GPT produces flat 'parallelograms'.

steps). The final validation losses are 3.159 (standard) and 3.146 (harmonic). From training loss curves, harmonic GPT also seems to have smaller fluctuations. This suggests the effectiveness of the harmonic loss on real-world models.

To testify the interpretability of the learned embeddings, we take twelve function-vector tasks from (Todd et al., 2023). Each dataset contains many input-output pairs that have a certain relation. For example, the "present-past" dataset contains pairs like: jump-jumped, fasten-fastened, win-won, etc. To construct parallelograms, we can draw two different pairs from the dataset, obtaining quadruples like (jump, jumped, fasten, fastened) which are expected to form parallelograms. Each word is tokenized into tokens; if multiple tokens are obtained, we use the last token. We project token embeddings onto the first two principal components. The quadruple (i, j, m, n) has 2D PC embeddings $(\boldsymbol{E}_i, \boldsymbol{E}_j, \boldsymbol{E}_m, \boldsymbol{E}_n)$; we define the parallelogram loss l_{para} to be

$$l_{\text{para}} = \|\boldsymbol{E}_i + \boldsymbol{E}_n - \boldsymbol{E}_j - \boldsymbol{E}_m\|/\sigma, \tag{5}$$

where $\sigma = \sqrt{\frac{1}{V}\sum_{k=1}^{V}\|\boldsymbol{E}_k\|^2}$ is a scale factor that normalizes the loss ($\boldsymbol{E}_k \to a\boldsymbol{E}_k$ leaves l_{para} invariant). We obtain 10000 quadruples, measuring the parallelogram qualities by computing their parallelogram losses. We plot their cumulated distribution function in Figure 6 top right: for every task, the harmonic GPT produces lower parallelogram loss (better parallelograms) than standard GPT. We show the parallelograms obtained in the present-past task in Figure 6 bottom. The parallelograms are ranked with quality in descending order from left to right. The harmonic GPT tends to produce visually appealing parallelograms that are more 'rectangular', while standard GPT produces flat 'parallelograms'.

7. Conclusions

In this paper, we introduced harmonic loss as an alternative to the standard cross-entropy loss for training neural networks and large language models (LLMs). We found that models trained with harmonic loss outperform standard models by: (a) reducing grokking, (b) requiring less data for generalization, and (c) improving interpretability. We also compared a GPT-2 model trained with harmonic loss to the standard GPT-2, illustrating that the harmonic loss-trained model develops more interpretable representations. Further study is needed to explore the scalability and applicability of our findings to even larger models.

Impact Statement

This paper presents work whose goal is to enhance the interpretability of machine learning systems. Our harmonic loss approach enables a deeper understanding of model behavior, thereby improving the trustworthiness of machine learning systems.

References

Abdou, M., Kulmizev, A., Hershcovich, D., Frank, S., Pavlick, E., and Søgaard, A. Can language models encode perceptual structure without grounding? a case study in color. *arXiv preprint arXiv:2109.06129*, 2021.

Bereska, L. and Gavves, E. Mechanistic interpretability for ai safety–a review. *arXiv preprint arXiv:2404.14082*, 2024.

Bommidi, B. S., Teeparthi, K., and Kosana, V. Hybrid wind speed forecasting using iceemdan and transformer model with novel loss function. *Energy*, 265:126383, 2023.

Bosco, E., Magenes, G., and Matrone, G. Echocardiographic image segmentation with vision transformers: A comparative analysis of different loss functions. In *2024 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, pp. 1–6. IEEE, 2024.

Demir, A., Massaad, E., and Kiziltan, B. Topology-aware focal loss for 3d image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 580–589, 2023.

Ding, X. D., Guo, Z. C., Michaud, E. J., Liu, Z., and Tegmark, M. Survival of the fittest representation: A case study with modular addition. *arXiv preprint arXiv:2405.17420*, 2024.

Engels, J., Liao, I., Michaud, E. J., Gurnee, W., and Tegmark, M. Not all language model features are linear. *arXiv preprint arXiv:2405.14860*, 2024.

Gurnee, W. and Tegmark, M. Language models represent space and time. *arXiv preprint arXiv:2310.02207*, 2023.

Heinzerling, B. and Inui, K. Monotonic representation of numeric properties in language models. *arXiv preprint arXiv:2403.10381*, 2024.

Li, B. Z., Nye, M., and Andreas, J. Implicit representations of meaning in neural language models. *arXiv preprint arXiv:2106.00737*, 2021.

Li, C., Yao, Z., Wu, X., Zhang, M., Holmes, C., Li, C., and He, Y. Deepspeed data efficiency: Improving deep learning model quality and training efficiency via efficient data sampling and routing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 18490–18498, 2024a.

Li, X., Sun, Q.-L., Zhang, Y., Sha, J., and Zhang, M. Enhancing hydrological extremes prediction accuracy: Integrating diverse loss functions in transformer models. *Environmental Modelling & Software*, 177:106042, 2024b.

Li, Y., Michaud, E. J., Baek, D. D., Engels, J., Sun, X., and Tegmark, M. The geometry of concepts: Sparse autoencoder feature structure. *arXiv preprint arXiv:2410.19750*, 2024c.

Lin, T. Focal loss for dense object detection. *arXiv preprint arXiv:1708.02002*, 2017.

Liu, J., Shen, Z., He, Y., Zhang, X., Xu, R., Yu, H., and Cui, P. Towards out-of-distribution generalization: A survey. *arXiv preprint arXiv:2108.13624*, 2021.

Liu, Z., Kitouni, O., Nolte, N. S., Michaud, E., Tegmark, M., and Williams, M. Towards understanding grokking: An effective theory of representation learning. *Advances in Neural Information Processing Systems*, 35:34651–34663, 2022a.

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

Liu, Z., Gan, E., and Tegmark, M. Seeing is believing: Brain-inspired modular training for mechanistic interpretability. *Entropy*, 26(1):41, 2023.

Luo, J., Qiao, H., and Zhang, B. Learning with smooth hinge losses. *Neurocomputing*, 463:379–387, 2021.

Marks, S. and Tegmark, M. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. *arXiv preprint arXiv:2310.06824*, 2023.

Michaud, E. J., Liao, I., Lad, V., Liu, Z., Mudide, A., Loughridge, C., Guo, Z. C., Kheirkhah, T. R., Vukelić, M., and Tegmark, M. Opening the ai black box: program synthesis via mechanistic interpretability. *arXiv preprint arXiv:2402.05110*, 2024.

Novak, R., Bahri, Y., Abolafia, D. A., Pennington, J., and Sohl-Dickstein, J. Sensitivity and generalization in neural networks: an empirical study. *arXiv preprint arXiv:1802.08760*, 2018.

Olah, C., Cammarata, N., Schubert, L., Goh, G., Petrov, M., and Carter, S. Zoom in: An introduction to circuits. *Distill*, 5(3):e00024–001, 2020a.

Olah, C., Cammarata, N., Schubert, L., Goh, G., Petrov, M., and Carter, S. Zoom in: An introduction to circuits. *Distill*, 2020b. doi: 10.23915/distill.00024.001. https://distill.pub/2020/circuits/zoom-in.

Olsson, C., Elhage, N., Nanda, N., Joseph, N., DasSarma, N., Henighan, T., Mann, B., Askell, A., Bai, Y., Chen, A., et al. In-context learning and induction heads. *Transformer Circuits Thread*, 2022.

Park, C. F., Lee, A., Lubana, E. S., Yang, Y., Okawa, M., Nishi, K., Wattenberg, M., and Tanaka, H. Iclr: In-context learning of representations. *arXiv preprint arXiv:2501.00070*, 2024a.

Park, K., Choe, Y. J., Jiang, Y., and Veitch, V. The geometry of categorical and hierarchical concepts in large language models. *arXiv preprint arXiv:2406.01506*, 2024b.

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

Salehi, S. S. M., Erdogmus, D., and Gholipour, A. Tversky loss function for image segmentation using 3d fully convolutional deep networks. In *International workshop on machine learning in medical imaging*, pp. 379–387. Springer, 2017.

Seber, P. Predicting o-glcnacylation sites in mammalian proteins with transformers and rnns trained with a new loss function. *arXiv preprint arXiv:2402.17131*, 2024.

Sudre, C. H., Li, W., Vercauteren, T., Ourselin, S., and Jorge Cardoso, M. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: Third International Workshop, DLMIA 2017, and 7th International Workshop, ML-CDS 2017, Held in Conjunction with MICCAI 2017, Québec City, QC, Canada, September 14, Proceedings 3*, pp. 240–248. Springer, 2017.

Templeton, A., Conerly, T., Marcus, J., Lindsey, J., Bricken, T., Chen, B., Pearce, A., Citro, C., Ameisen, E., Jones, A., et al. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. transformer circuits thread, 2024.

Todd, E., Li, M. L., Sharma, A. S., Mueller, A., Wallace, B. C., and Bau, D. Function vectors in large language models. *arXiv preprint arXiv:2310.15213*, 2023.

Wang, Z., Jiang, Y., Zheng, H., Wang, P., He, P., Wang, Z., Chen, W., Zhou, M., et al. Patch diffusion: Faster and more data-efficient training of diffusion models. *Advances in neural information processing systems*, 36, 2024.

Zhong, Z., Liu, Z., Tegmark, M., and Andreas, J. The clock and the pizza: Two stories in mechanistic explanation of neural networks. *Advances in Neural Information Processing Systems*, 36, 2024.

A. Full Representation Visualization

Figure 7 shows the visualization of representations for all models and datasets.

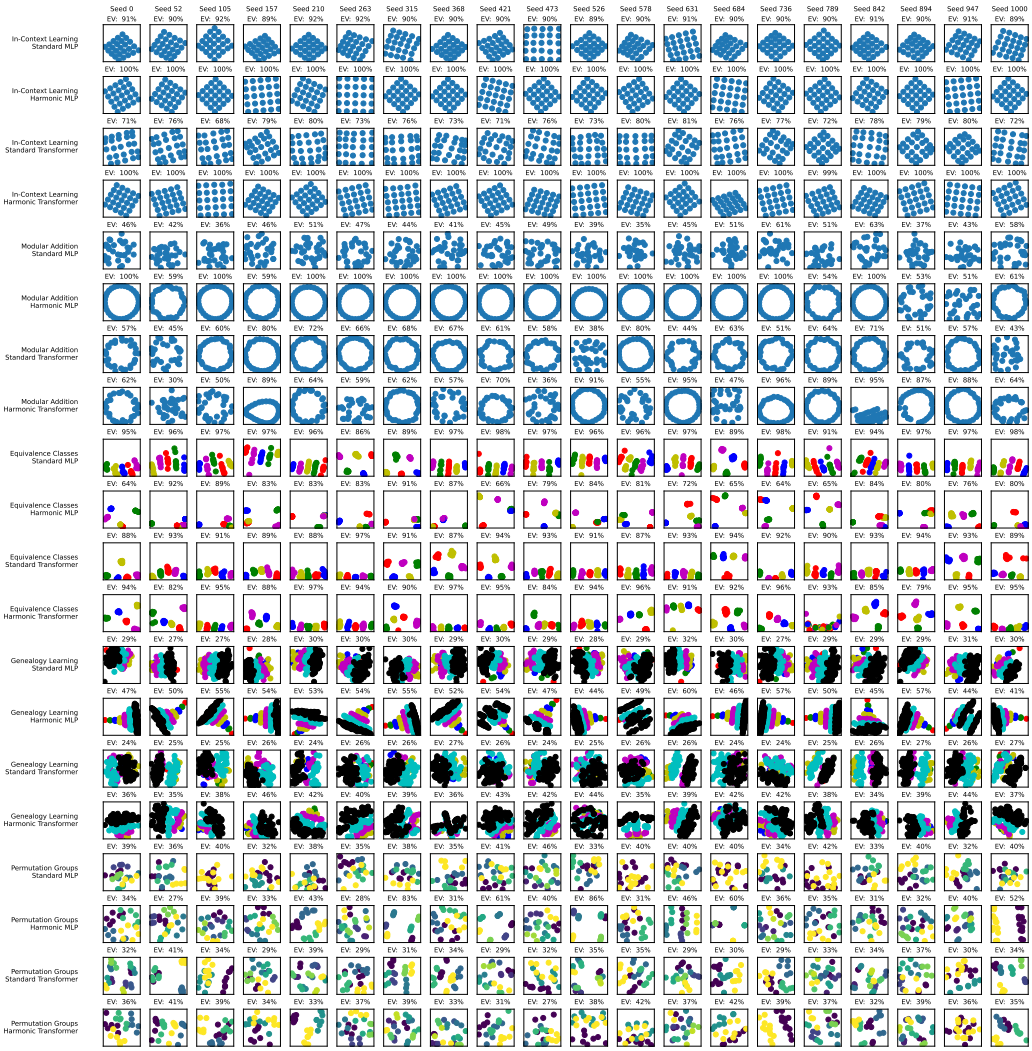

Figure 7. Visualization of the top two principal components of the embeddings in synthetic experiments. The title of each subplot shows the explained variance by the first two principal components. Each row corresponds to a pair of a dataset and a model, while each column represents the embeddings from different training runs with varying seeds. Groups of four rows belong to the same dataset, with models arranged in the order: {Standard MLP, Harmonic MLP, Standard Transformer, Harmonic Transformer}. The datasets are ordered as follows: {In-Context Learning, Genealogy Learning, Equivalence Classes, Modular Addition, and Permutation Groups}.

B. Identifying Coset Structure in Permutation Representations

To explore the coset structure in permutation representations of S_4, we began by enumerating its subgroups. Using this enumeration, we computed all possible left and right cosets of each subgroup in S_4, yielding 28 distinct left cosets and 28 distinct right cosets.

Among these cosets, two pairs are equivalent, since we consider two of the four normal subgroups of S_4: the alternating group A_4 and the Klein-4 group. To focus on meaningful structures, the trivial subgroup and the entire group were excluded from further analysis.

The coset partitions were then compared using the silhouette score, a metric for evaluating the quality of clustering. This comparison helped identify the partition with the most structured coset organization, which is likely the structure that the model has captured during training. We then color the representation according to the best-clustered partition, with each coset being a different color.