# Can Neural Networks Learn Small Algebraic Worlds? An Investigation Into the Group-theoretic Structures Learned By Narrow Models Trained To Predict Group Operations

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

While a real-world research program in mathematics may be guided by a motivating question, the process of mathematical discovery is typically open-ended. Ideally, exploration needed to answer the original question will reveal new structures, patterns, and insights that are valuable in their own right. This contrasts with the exam-style paradigm in which the machine learning community typically applies AI to math. To maximize progress in mathematics using AI, we will need to go beyond simple question answering. With this in mind, we explore the extent to which narrow models trained to solve a fixed mathematical task learn broader mathematical structure that can be extracted by a researcher or other AI system. As a basic test case for this, we use the task of training a neural network to predict a group operation (for example, performing modular arithmetic or composition of permutations). We describe a suite of tests designed to assess whether the model captures significant group-theoretic notions such as the identity element, commutativity, or subgroups. Through extensive experimentation we find evidence that models learn representations capable of capturing abstract algebraic properties. For example, we find hints that models capture the commutativity of modular arithmetic. We are also able to train linear classifiers that reliably distinguish between elements of certain subgroups (even though no labels for these subgroups are included in the data). On the other hand, we are unable to extract notions such as the concept of the identity element. Together, our results suggest that in some cases the representations of even small neural networks can be used to distill interesting abstract structure from new mathematical objects.

## 1 Introduction

Deep learning-based systems are increasingly being used as a tool to accelerate research mathematics. Though there is a growing body of work that aims for generalist AI scientists Lu et al. [2024], Yamada et al. [2025] or program synthesis systems like AlphaEvolve Novikov et al. [2025], Romera-Paredes et al. [2024], the majority of AI for math work still starts with a specific problem of interest and then builds a system to learn a solution to this problem. This system may be a narrow model trained exclusively on a task related to the problem Davies et al. [2021] or it may be a more sophisticated framework using foundation models like LLMs. What these set-ups have in common is that they are often restricted to revealing solutions to the initial question. Real mathematics research on the other hand is substantially more open-ended with the final output of a research program often varying substantially from the initial motivating question.

Submitted to 1st Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

If the goal is to develop methodologies that enable more open-ended discovery, one potential solution is to look more closely at effective narrow models. Might these already contain valuable insights that were learned in the course of solving the original task? There are a range of case studies that provide precedent for such a hypothesis. For instance, careful analysis of models trained to perform modular arithmetic Nanda et al. [2023a] or composition of permutations Chughtai et al. [2023], Stander et al. [2023], Wu et al. [2024] have revealed sophisticated algorithms that depend on the representation theory of the corresponding groups. Similarly, analysis of a graph neural network designed to perform classification of the mutation equivalence class of a finite or affine type quiver was found to naturally cluster instances in ways that align with human classification schemes He et al..

Motivated by this question, we explore the elementary case of a neural network trained to predict the operation of a group $G$ and ask to what extent we can detect basic group-theoretic concepts from such a network. Notions we consider include commutativity, the identity, and subgroups, all core concepts within group theory. We explore three approaches to detecting these concepts: (i) through training dynamics where we look for changes in loss/accuracy that might correspond to a model learning a new concept, (ii) differences in performance across subsets of input instances (for example, a model that 'understands' the identity element $e$ should always get the correct answer on questions of the form $g \star e$ or $e \star g$, even when it has never seen $g$ before), and (iii) the structure of the internal representations of the group.

We probe MLPs and transformers trained to perform the group operation for cyclic groups, symmetric groups, and dihedral groups of varying sizes. Across these settings we find that concepts which offer shortcuts in task computation and are relevant to many different instances (e.g., commutativity) can be detected via several means, while other concepts (the identity element property) that are only relevant to a minority of instances cannot be detected. More surprisingly, we find that we are able to distinguish between pairs of elements $(g_1, g_2)$ belonging to a subgroup $H$ and pairs that do not belong to $H$ in the latent space of the model, even though the model was not trained with subgroup labels.

All of this suggests that the *mathematical world models* learned by small neural networks (easily accessible to even those with very restricted compute budgets) can contain interesting mathematical insights available to those that are willing to put in the work to extract them. In summary our contributions include the following.

- We describe a framework that uses finite groups and basic notions from group theory to better understand whether narrowly trained neural networks have world models sufficiently rich for open ended mathematical discovery.
- We evaluate a range of approaches to extract the fingerprints of concepts such as commutativity or the notion of a subgroup from a neural network trained for a fixed task, confirming that representation-based approaches that probe the hidden activations of a model are the most promising.
- We give some intuitive rules of thumb for the types of structures that narrow models are likely to learn, and those they are unlikely to learn.

## 2 Finite Groups and Their Associated Concepts

The notion of a group is a central concept in modern mathematics. A group is a set $G$ along with a binary operation $\star : G \times G \to G$ which satisfies the following axioms. (1) $\star$ is associative so that for all $g_1, g_2, g_3 \in G$, $(g_1 \star g_2) \star g_3 = g_1 \star (g_2 \star g_3)$. (2) There is an identity element $e \in G$ such that $g \star e = e \star g = g$ for all $g \in G$. (3) Each $g \in G$ has an inverse element $g^{-1} \in G$ such that $g \star g^{-1} = e = g^{-1} \star g$.

Despite arising from only three simple axioms, groups exhibit amazingly rich structure ranging from cyclic groups (familiar from modular arithmetic) to the *monster*, the largest sporadic group which has order $\approx 10^{53}$. Careers have been spent studying groups and one product of this is a rich language that captures the breadth of structure arising in this field. In this work we only scratch the surface of this, asking whether we can recover the following notions from a neural network trained to perform the binary operation $\star$:

- **Commutativity of the binary operation $\star$:** $\star$ is commutative if for all $g_1, g_2 \in G$, we have $g_1 \star g_2 = g_2 \star g_1$.

- **The identity element:** The element $e$ is uniquely defined in $G$ by the fact that $e \star g = g = g \star e$ for all $g \in G$.
- **Subgroup structure:** There may exist proper, non-trivial subsets $H \subseteq G$ that are closed under $\star$ and hence form groups in their own right. These *subgroups* form lattices under the containment relationship.

We look at three different families of groups, which we describe here.

**Cyclic groups, $\mathbb{Z}/p\mathbb{Z}$:** Cyclic groups are familiar since they correspond to modular addition. We can represent $\mathbb{Z}/p\mathbb{Z}$ with the elements $\{\overline{0}, \overline{1}, \ldots, \overline{p-1}\}$ and realize the binary operation as $\overline{a} + \overline{b} = \overline{a + b \mod p}$. $\mathbb{Z}/p\mathbb{Z}$ is commutative and has order $|\mathbb{Z}/p\mathbb{Z}| = p$. The subgroups of $\mathbb{Z}/p\mathbb{Z}$ are in one-to-one correspondence with integers $1 \leq k \leq p$ such that $k$ divides $p$. If $k$ divides $p$, then we can realize the corresponding subgroup as $\{\overline{0}, \overline{k}, \overline{2k}, \ldots\}$.

**Symmetric groups, $S_n$:** The symmetric group $S_n$ is the set of all permutations of $n$ elements with the binary operation of composition of permutations. As such, the order of $S_n$ is $n!$. It is not commutative. $S_n$ has many subgroups but we will work with one of the most well-known, the *alternating (sub)group*, which consists of all permutations of $n$ which are even. The alternating group has size $n!/2$.

**Dihedral groups $D_n$:** The dihedral group $D_n$ can be realized as the set of rotations and reflections that preserve the $n$-gon. It consists of $n$ rotations and $n$ reflections, making it a group of order $2n$. Subgroups of $D_n$ include the subgroup of all $n$ rotations.

# 3 How Can We Detect Whether a Model Has Learned a Mathematical Concept?

Suppose $f : G \times G \to G$ is a neural network that has been trained to perform the binary operation $\star$ of a group. Thus, provided with $g_1, g_2 \in G$, $f$ predicts $g_1 \star g_2$. We outline the three broad approaches that we use to detect whether $f$ has 'learned' algebraic concepts that characterize groups (at least, as a human mathematician would describe them).

- **Learning dynamics:** Detailed analysis of neural network training has revealed that (at least in simple problems), sudden drops in loss may correspond to a network gaining a specific capability . Might we see similar changes in the loss curve when a network trained on a group binary operation learns a concept like commutativity of $\star$? Motivated by this idea, we explore whether changes in accuracy or loss correlate with changes in model performance on specific subpopulations of the test set which capture a certain concept. For example, one can imagine that a sudden drop in loss might correspond to a model achieving high accuracy on instances of the form $e \star g$ or $g \star e$, suggesting that the model has learned the concept of the identity element.
- **Generalization:** Mathematical concepts are valuable precisely because they allow us to reason broadly across instances we have not seen before. One may not have ever actually worked with the numbers $2,483,402$ and $5,840,202$ but we can immediately say that $2,483,402 + 5,840,202 = 5,840,202 + 2,483,402$ based on mathematical concepts that we already understand. One way to evaluate whether a model has learned a concept is to see whether the model can apply that concept to an out-of-distribution example.
- **Structure of Internal Representations:** A model's understanding of a concept may manifest as structure in the hidden activations of the model. For example, a model may encode commutativity by representing $g_1 \star g_2$ and $g_2 \star g_1$ as more similar than arbitrary $g_1 \star g_2$ and $g_3 \star g_4$ even when $g_1 \star g_2 = g_3 \star g_4$. This perspective aligns with the mechanistic interpretability paradigm.

## 3.1 Experimental details

In our experiments we use MLPs and transformers that are of a scale that is accessible to most researchers but are sufficient to learn the group operation. Our MLPs consist of dense linear layers interleaved with ReLU nonlinearities. Our transformers are decoder-only and use GeLU nonlinearities. The task is framed as prediction and thus uses a standard cross-entropy loss function. All models are trained using the Adam optimizer with varying learning rate values and weight decay on a single

Nvidia A100. In most experiments, we explore a wide range of hyperparameters. We provide the hyperparameters that we use for the paper's visualizations in Section A in the Appendix.

Our experiments use a one-hot encoding of elements of $G$. For MLPs, we encode $g_1 \star g_2$ by concatenating two length $|G|$ vectors into a single $2|G|$-dimensional input. Since the output prediction is an element of $G$, the output dimension is $|G|$. For transformers we encode $g_1 \star g_2$ as a length 3 sequence, the first token corresponding to $g_1$ and the second token corresponding to $g_2$. The final token, on which the transformer's prediction is made, can be taken to correspond to '='. Thus, the transformer operates on a vocabulary of size $|G| + 1$. In our experiments we work with the cyclic groups $C_{64}$, $C_{67}$, $C_{100}$, $C_{256}$, $C_{257}$, $C_{508}$, $C_{512}$, the symmetric groups $S_4$, $S_5$, and $S_6$, and the dihedral groups $D_{30}$, $D_{50}$, $D_{60}$, $D_{120}$, and $D_{240}$.

### 3.2 Commutativity

It is easy to see why knowing that a group is commutative can lead to more efficient computation. If $G$ is commutative, one immediately knows the value of $g_2 \star g_1$ once they know the value of $g_1 \star g_2$. Beyond this, commutativity has deep consequences for the types of structural features that a group can exhibit with commutative groups generally being much simpler. Our first set of experiments aim to understand whether MLPs and transformers capture commutativity using the perspectives described at the beginning of Section 3. Our work extends Kvinge et al., which used the cosine similarity test below to determine whether large language models have an internal notion of commutativity. We begin by describing our experiments.

**Symmetric consistency:** As noted above, one sign that a model has internalized the notion of commutativity would be that the model's prediction of $g_1 \star g_2$ and $g_2 \star g_1$ will tend to be the same, regardless of correctness. The following quantities aim to measure this.

Let $\mathcal{S}$ be the full set of pairs $\big((g_1, g_2), (g_2, g_1)\big)$ in the test set. The *symmetric consistency* is measured by computing the fraction of pairs where $f(g_1, g_2) = f(g_2, g_1)$.

$$\text{Consistency of } f := \frac{1}{|\mathcal{S}|}\#\{f(g_1, g_2) = f(g_2, g_1) \mid (g_1, g_2), (g_2, g_1) \in \mathcal{S}\}.$$

Symmetric consistency values closer to 1 indicate that $f$ makes more consistent predictions across pairs $(g_1, g_2)$ and $(g_2, g_1)$. However, care is needed because this consistency statistic alone can be maximized through better performance on the test set. In other words, any model that achieves 100% accuracy on the test examples will have symmetric consistency equal to 1, even if it is simply a look-up table. To mitigate this, we also measure *equal value consistency* which is the fraction of times that $f$ predicts $f(g_1, g_2) = f(g_3, g_4)$ for $(g_1, g_2), (g_3, g_4) \in S$ such that $g_1 \star g_2 = g_3 \star g_4$. We compare these over the course of training. Increases in symmetric consistency independent of equal value consistency may be evidence of a learned concept of commutativity in the model.

Another way we can probe for commutativity is to look at how strongly symmetric consistency holds on out-of-distribution examples that were not seen during training. We call this version *out-of-distribution (OOD) symmetric consistency*. Here we choose some $g' \in G$ and remove all examples from $S_{g'} := \{(g', g),(g, g') \mid g \in G\}$ from the training set. We then measure symmetric consistency on $S_{g'}$ to see if the model can apply any learned notion of commutativity to elements of $S_{g'}$ which feature the novel element $g'$.

**Cosine similarity:** Commutativity means that, algebraically, we are allowed to treat $g_1 \star g_2$ and $g_2 \star g_1$ as equal. In the context of large language models, Kvinge et al. hypothesized that such an understanding might manifest as the model constructing internal representations of $g_1 \star g_2$ and $g_2 \star g_1$ that are closer in hidden activation space. For a given input $(g_1, g_2)$, denote by $v^k_{g_1,g_2}$ an activation vector corresponding to $(g_1, g_2)$ extracted from the $k$th layer of $f$. The *symmetric representational similarity of $f$ at layer $k$* is

$$\text{Symmetric representational similiarity of } f := \frac{1}{|S|}\sum_{(g_1,g_2),(g_2,g_1)\in\mathcal{S}} \text{Sim}(v^k_{g_1,g_2}, v^k_{g_2,g_1}). \quad (1)$$

As in the case of symmetric consistency, we also compute a version of (1) where the sum is taken over pairs $(g_1, g_2), (g_3, g_4)$ where $g_1 \star g_2 = g_3 \star g_4$ to ensure trends that we see do not simply arise because $g_1 \star g_2 = g_2 \star g_1$.


Figure 1: Plots of symmetric consistency vs. equal value consistency **(Left)** and OOD symmetric consistency vs. **(Right)** for five transformers trained to perform modular arithmetic in $C_{100}$. Shaded regions correspond to $95\%$ confidence intervals.

**Do models capture commutativity?:** The only class of commutative groups that we explored were cyclic groups, so we focus our analysis on this setting. We note that by design, transformers are better adapted for capturing commutativity since the representations of the tokens corresponding to $g_1$ and $g_2$ in $g_1 \star g_2$ are the same as the representations of the tokens corresponding to $g_1$ and $g_2$ in $g_2 \star g_1$ up to modification by positional encoding. On the other hand, the one-hot encodings of $g_1 \star g_2$ and $g_2 \star g_1$ are orthogonal.

In general, over the course of training all models tend to have higher symmetric consistency than the consistency of arbitrary $g_1 \star g_2$ and $g_3 \star g_4$ of equal value (though this difference is comparatively small in MLPs). We also found that many (but not all) training runs yielded non-trivial OOD symmetric consistency values. This is seen in Figure 1 for five transformers trained to predict the binary operation of the cyclic group $C_{100}$. Finally, examining performant model's internal representations also reveal the fingerprints of commutativity with higher cosine similarity between pairs $g_1 \star g_2$ and $g_2 \star g_1$ relative to pairs that simply have equal value. An illustration of this can be found in Figure 3 in the Appendix.

Despite these hints of commutativity, we note that the signals described above remain weak with both MLPs and transformers achieve OOD symmetric consistency that is substantially higher than chance $(1/|G|)$ but significantly less than 1 (this is apparent on the left in Figure 1).

Transformers and MLPs appear to learn some notion of commutativity but it remains brittle.

### 3.3 The identity

Once one has identified the identity element $e$ in a group, computations involving $e$ are trivial for a human to perform. This is true even when one otherwise understands little else about the group. As such it is interesting to try to understand whether a neural network trained to predict the group operation also leverages the unique property that $g \star e = g$. We introduce two approaches to understand whether models learn the identity as a special and distinct element of the group like humans do.

**Identity accuracy:** This simply involves tracking the accuracy on test examples of the form $g \star e$ or $e \star g$ in the test set. We then compare this to the global test accuracy. Substantial increases in identity accuracy relative to global accuracy may indicate a point in training where the model learns the identity.

As with symmetric consistency, we can also probe the extent to which a model has a strong concept of the identity element by testing whether it can generalize the properties of the identity to out of distribution examples. To do this we hold out a subset of elements $g'_1, g'_2, \ldots, g'_t \in G$ so that none of these elements appear in the training set. The *OOD identity accuracy* is then the accuracy of the subset of the test set of the form $g'_j \star g$ or $g \star g'_j$ for $1 \leq j \leq t$.

**Do models have a notion of the identity element?:** In all our experiments, identity accuracy tracked the overall accuracy of the model closely giving no hint of a point where the model learned a specific prediction rule around the identity element. This is supported by results on OOD identity accuracy where no models were able to reliably predict that $g \star e = g$ or $e \star g = g$ if they had not seen $g$ during training.

We were unable to surface any evidence that our neural networks recognize the identity as a special element of the group using the proposed techniques above.

### 3.4 Subgroup structure

While understanding the identity element and commutativity offer obvious potential benefits for more efficient computation, the benefits of being able to distinguish different subgroups seems less clear. Nevertheless, analysis in works like McCracken et al. [2025] suggest that in some cases, the structure of certain subgroups may play a role in computation (via for example, the Chinese remainder theorem). Motivated by this, we propose the following tests to explore whether models learn to identify the subgroups of a group.

**Subgroup accuracy:** Analogous to identity accuracy, we can also look at *subgroup accuracy*, the accuracy on pairs of elements $(h_1, h_2)$ belonging to the subgroup $H$. If we see significant changes in subgroup accuracy relative to global accuracy, this may correspond to a point in training where $f$ 'learned' $H$.

**Linear probing for subgroup membership:** We can test whether $f$ captures a distinct representation of $H$ by probing for subgroup membership on the hidden activations of $f$. More specifically, we can collect representation $v^k_{g_1,g_k} \in \mathbb{R}^{d_k}$ corresponding to $g_1 \star g_2$ at layer $k$, label them by whether $g_1, g_2 \in H$, and then train a linear probe to predict these labels.

Note that in this test we should expect different behavior based on the data representation of transformers and MLPs. In the MLP case where input is a one-hot encoding of $g_1$ stacked on a one-hot encoding of $g_2$, this task should be easy in input space since the probe just needs to learn the $|H|$ indices corresponding to elements of $H$ in the first $|G|$ dimensions, the $|H|$ indices corresponding to elements of $H$ in the second $|G|$ dimensions, and be able to perform an 'and' operation over these. We expect this task to become more challenging as $f$ transforms this initial representation of $(g_1, g_2)$ when computing $g_1 \star g_2$. In the case of transformers where the input is three tokens long: one token for $g_1$, one token for $g_2$, and one token for '=' and we predict $g_1 \star g_2$ from the third token, the task is impossible in input space (since the third token is the same for all input). It only becomes tractable as information from the first and second tokens representing $g_1$ and $g_2$ are transferred onto the third token via successive self-attention layers. In this situation if the representation of '=' retains information identifying the first argument as $g_1$ and the second argument as $g_2$, then it is possible that the subgroup could be learned via the same procedure described for the MLPs.

To better calibrate probe accuracy, we compare to a random labeling of $|H|$ elements of the group.

**Do models see subgroups?:** Unlike the process whereby a human might learn a group, first understanding a simpler subgroup and then building toward understanding the whole group, we find that across groups, subgroups, architectures, and hyperparameters, subgroup accuracy tracks global accuracy. This aligns with the behavior of identity accuracy seen in Section 3.3.

On the other hand, we find substantial evidence that performant models sometimes capture subgroup structure within their internal representations which we can access via the linear probing. We probe for several large subgroups of cyclic groups including the order 50 and order 20 subgroups generated in $C_{100}$, the subgroups generated by all rotations in $D_{30}$ and $D_{50}$ (order 30 and 50 respectively), and the order 60 alternating subgroup in $S_5$. The accuracy of these probes, trained on both the true subgroup labels (solid lines) and random labels (dashed lines) over the course of training ($x$-axis) and at different layers (colors) are shown in Figure 2 for a transformer trained on $S_5$ (left), and an MLP trained on $D_{30}$ (right).

We stress that probe performance is variable across runs. Sometimes models achieve high accuracy predicting the group operation and yet their probe performance is close to guessing.

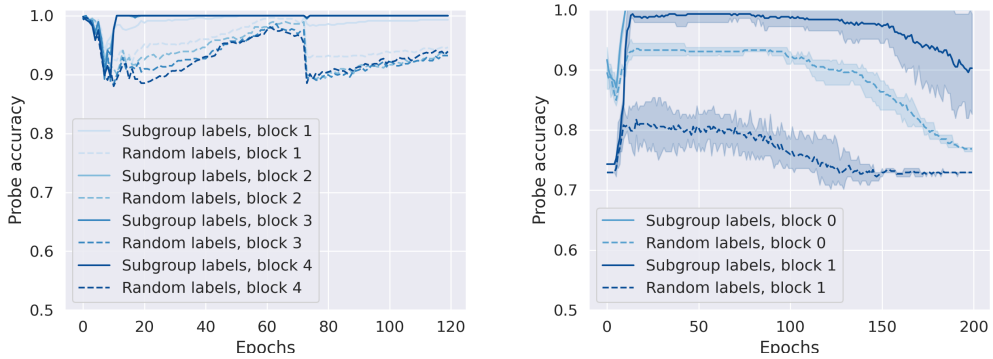


Figure 2: Linear probe performance on a **(Left)** transformer trained to predict the binary of $S_5$ and a **(Right)** MLP trained to predict the binary operation on $D_{30}$. Solid lines are probe accuracy when labels correspond to the alternating subgroup and rotation subgroup respectively, while the dashed line corresponds to a probe trained on a random labeling of the group. In the right plot, shading indicates 95% confidence intervals over three random initializations (transformer performance on $S_5$ varied too much between random initializations to be useful in the case of the transformer). In the case of the MLP, probing was performed after each ReLU layer, in the transformer it was performed after each attention layer.

The shadows of subgroups can be discerned through linear probing of a model's internal representations.

# 4 Discussion

**We find analysis of model internals to be the most effective approach to detecting mathematical structure learned by a model.** This aligns with current explainability paradigms such as mechanistic interpretability. On the other hand, we found that most models displayed weak generalization with learned mathematical structures decaying substantially when we move to out-of-distribution input (e.g., models almost never predicted that $e \star g' = g'$ in cases where they had not seen $g'$ during training). Similarly, the training dynamics associated with specific structures mostly tracked the general progress of the model. For instance, subgroup and identity accuracy tracked the global accuracy of the model closely. This highlights an important difference between the way that these small neural networks learn a new group and the way that a human learns a new group. The human will often start by trying to understand simpler patterns and components, progressively building toward an understanding of the whole. On the other hand, either we have not found the simpler building blocks that models learn or models simply learn everything 'all at once'.

**What mathematical structures tend to be captured by a small neural network?** This is an important question to answer as it helps us identify whether the small model paradigm will be useful for a particular research program. Based on the successes and failures described above, we suggest three intuitive rules of thumb to predict whether a given mathematical property $X$ is likely to be learned by a small model trained on a task $Y$.

- Encoding $X$ allows the neural network to find a simpler solution applicable across many instances for the task $Y$. As an example, encoding commutativity and the trivial nature of the identity element can both lead to simpler solutions. But whereas commutativity applies to all instances, the special property of the identity only applies to $2|G| - 1$ instances out of $|G|^2$.
- $X$ fits into an existing probing framework. For example, we found it hard to investigate whether models recognize the significance of the identity element because this property is challenging to formulate within most model probing techniques, as it does not easily lend itself to basic analysis techniques in latent space.

**Mathematical world models are sensitive to hyperparameter choice and initialization.** This may in part be due to the nature of small algorithmic problems. Like others, we have found that the performance in these settings tends to be more sensitive to initial conditions and sources of randomness in training than other small-scale tasks (e.g., CIFAR10, MNIST). But even among training runs with identical hyperparameters (that converged) we found significant differences in the mathematical structures we were able to extract. For example, one trend that we noticed when training transformers on $S_5$ is that when the model converged earlier (after $20 - 50$ epochs) the alternating subgroup was undetectable via linear probing but that when the model converged after training for longer ($> 100$ epochs), the alternating subgroup was detectable. It may be that these mathematical properties offer a window to fundamentally different solutions learned by a model. Overall, we strongly recommend that when training models that will be used in downstream mathematical exploration, many different hyperparameters and initializations should be explored.

# 5 Related work

This work presents a framework to understand whether narrow models trained to perform a single algebraic task—predicting a group operation—learn group-theoretic notions. This adds to the growing literature on world models of neural networks, which has focused largely on sequence models in strategy game environments, and more recently LLMs Mitchell [2023], Li et al. [2022], Nanda et al. [2023b], Rohekar et al. [2025], while the algebraic world models of narrow models trained on mathematics tasks are much less explored.

Closest to our setting are mechanistic studies that reverse-engineer how small MLP and transformer models learn group operations. Nanda et al. [2023a], Zhong et al. [2023], Chughtai et al. [2023], Stander et al. [2023], Wu et al. [2024], McCracken et al. [2025]. These works have often found that the learned algorithms use group-theoretic structure, although they sometimes differ in the specific algorithms they reverse-engineer Chughtai et al. [2023], Stander et al. [2023]. Our work is complementary, asking whether we can extract evidence of abstract notions like commutativity or subgroup structure, regardless of whether we can extract the precise algorithm used.

Our work is motivated by the potential to use AI systems for mathematical discovery. This is a rapidly growing field, including using AI to generate proofs Yang et al. [2024], discover counterexamples Wagner [2021], and find mathematical constructions Charton et al. [2024], Alfarano et al. [2024], Yip et al. [2025]. Some of this work trains a model to solve a task closely related to the problem, and then relies on expert probing to extract mathematical insight Davies et al. [2021], He et al.. In contrast, we take a step back to ask whether narrow models learn abstractions needed in order to gain useful insights.

# 6 Limitations

This work uses the elementary setting of finite groups to explore the prospects for using machine learning models to surface interesting mathematics that falls outside the task a model was designed to solve. Given the set-up, we know in advance the kinds of structures we are looking for. As such, we sidestep the main technical challenge in this research program. However, we hope that our results which show that interesting structure does appear in models (particularly their latent space), will support the idea that this is a worthwhile research direction that remains accessible to researchers with medium to small computational resources.

# 7 Conclusion

In this paper we explore the question of whether neural networks trained on a simple mathematical task, predicting the binary operation that defines a group, capture interesting structure not specified in the task itself. Our results show that even small neural networks can learn interesting structure, in the case of this paper the commutativity of the group operation or the existence of particular subgroups. There are also important group properties, such as the existence of the identity, that we are not able find. Despite our positive results, we see the most significant technical challenge in the widespread use of this paradigm as being able to effectively extract insights from a model when one does not already know what they are.

# References


Alberto Alfarano, François Charton, and Amaury Hayat. Global lyapunov functions: a long-standing open problem in mathematics, with symbolic transformers. *ArXiv*, abs/2410.08304, 2024. URL https://api.semanticscholar.org/CorpusID:273323255.

Francois Charton, Jordan S. Ellenberg, Adam Zsolt Wagner, and Geordie Williamson. Patternboost: Constructions in mathematics with a little help from ai. *ArXiv*, abs/2411.00566, 2024. URL https://api.semanticscholar.org/CorpusID:273798668.

Bilal Chughtai, Lawrence Chan, and Neel Nanda. A toy model of universality: Reverse engineering how networks learn group operations. In *International Conference on Machine Learning*, pages 6243–6267. PMLR, 2023.

Alex Davies, Petar Velickovic, Lars Buesing, Sam Blackwell, Daniel Zheng, Nenad Tomasev, Richard Tanburn, Peter W. Battaglia, Charles Blundell, Andras Juhasz, Marc Lackenby, Geordie Williamson, Demis Hassabis, and Pushmeet Kohli. Advancing mathematics by guiding human intuition with ai. *Nature*, 600:70 – 74, 2021. URL https://api.semanticscholar.org/CorpusID:244837059.

Jesse He, Helen Jenne, Herman Chau, Davis Brown, Mark Raugas, Sara C Billey, and Henry Kvinge. Machines and mathematical mutations: Using gnns to characterize quiver mutation classes. In *Forty-second International Conference on Machine Learning*.

Henry Kvinge, Elizabeth Coda, Eric Yeats, Davis Brown, John Buckheit, Sarah McGuire Scullen, Brendan Kennedy, Loc Truong, William Kay, Cliff Joslyn, et al. Probing the limits of mathematical world models in llms. In *ICML 2025 Workshop on Assessing World Models*.

Kenneth Li, Aspen K. Hopkins, David Bau, Fernanda Vi'egas, Hanspeter Pfister, and Martin Wattenberg. Emergent world representations: Exploring a sequence model trained on a synthetic task. *ArXiv*, abs/2210.13382, 2022. URL https://api.semanticscholar.org/CorpusID:253098566.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The ai scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*, 2024.

Gavin McCracken, Gabriela Moisescu-Pareja, Vincent Letourneau, Doina Precup, and Jonathan Love. Uncovering a universal abstract algorithm for modular addition in neural networks. *arXiv preprint arXiv:2505.18266*, 2025.

Melanie Mitchell. Ai's challenge of understanding the world. *Science*, 382 6671:eadm8175, 2023. URL https://api.semanticscholar.org/CorpusID:265103674.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability, 2023a. URL https://arxiv.org/abs/2301.05217.

Neel Nanda, Andrew Lee, and Martin Wattenberg. Emergent linear representations in world models of self-supervised sequence models. In *BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, 2023b. URL https://api.semanticscholar.org/CorpusID:261530966.

Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco JR Ruiz, Abbas Mehrabian, et al. Alphaevolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint arXiv:2506.13131*, 2025.

Raanan Yehezkel Rohekar, Yaniv Gurwicz, Sungduk Yu, Estelle Aflalo, and Vasudev Lal. A causal world model underlying next token prediction: Exploring GPT in a controlled environment. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=qA3xHJzF6B.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M Pawan Kumar, Emilien Dupont, Francisco JR Ruiz, Jordan S Ellenberg, Pengming Wang, Omar Fawzi, et al. Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475, 2024.

Table 1: Summary of algebraic structure captured in small LLMs and MLPs trained to perform a groups binary operation.

| Name | Type | Captured structure |
|---|---|---|
| Commutativity | | |
| Symmetric consistency | Learning dynamics | Yes |
| OOD symmetric consistency | Generalization | Yes |
| Symmetric representational similarity | Representation structure | Yes |
| Identity | | |
| Identity accuracy | Learning dynamics | No |
| Identity generalization | Generalization | No |
| Subgroup structure | | |
| Subgroup accuracy | Learning dynamics | No |
| Hidden activation probing | Representation structure | Yes |


Dashiell Stander, Qinan Yu, Honglu Fan, and Stella Biderman. Grokking group multiplication with cosets. *arXiv preprint arXiv:2312.06581*, 2023.

Adam Zsolt Wagner. Constructions in combinatorics via neural networks. *ArXiv*, abs/2104.14516, 2021. URL https://api.semanticscholar.org/CorpusID:233443896.

Wilson Wu, Louis Jaburi, Jacob Drori, and Jason Gross. Towards a unified and verified understanding of group-operation networks. *arXiv preprint arXiv:2410.07476*, 2024.

Yutaro Yamada, Robert Tjarko Lange, Cong Lu, Shengran Hu, Chris Lu, Jakob Foerster, Jeff Clune, and David Ha. The ai scientist-v2: Workshop-level automated scientific discovery via agentic tree search. *arXiv preprint arXiv:2504.08066*, 2025.

Kaiyu Yang, Gabriel Poesia, Jingxuan He, Wenda Li, Kristin Lauter, Swarat Chaudhuri, and Dawn Xiaodong Song. Formal mathematical reasoning: A new frontier in ai. *ArXiv*, abs/2412.16075, 2024. URL https://api.semanticscholar.org/CorpusID:274965430.

Jacky H. T. Yip, Charles Arnal, François Charton, and Gary Shiu. Transforming calabi-yau constructions: Generating new calabi-yau manifolds with transformers. *ArXiv*, abs/2507.03732, 2025. URL https://api.semanticscholar.org/CorpusID:280151317.

Ziqian Zhong, Ziming Liu, Max Tegmark, and Jacob Andreas. The clock and the pizza: Two stories in mechanistic explanation of neural networks. *ArXiv*, abs/2306.17844, 2023. URL https://api.semanticscholar.org/CorpusID:259309406.


# A Hyperparameters

We provide the hyperparameters used to generate the plots below.

- In order to generate the plots in Figures 1 and 3 we used decoder-only transformers with: 4 attention blocks and 4 MLP blocks, residual stream dimension $1,000$, 8 attention heads, learning rate $0.0001$, weight decay $0.001$. We used $80\%$ of all $10,000$ instances for training and $20\%$ for test.
- For Figure 2 (left), we used 3 decoder-only transformers with: 4 attention blocks and 4 MLP blocks, residual stream dimension $1,000$, 8 attention heads, learning rate $0.0001$, weight decay $0.001$. We used $80\%$ of all $14,400$ instances for training and $20\%$ for test.
- For Figure 2 (right), we used 3 MLPs of depth 2 and width $1,000$, learning rate $0.001$, and weight decay $0.0005$. We used $80\%$ of all $3,600$ instances for training and $20\%$ for test.

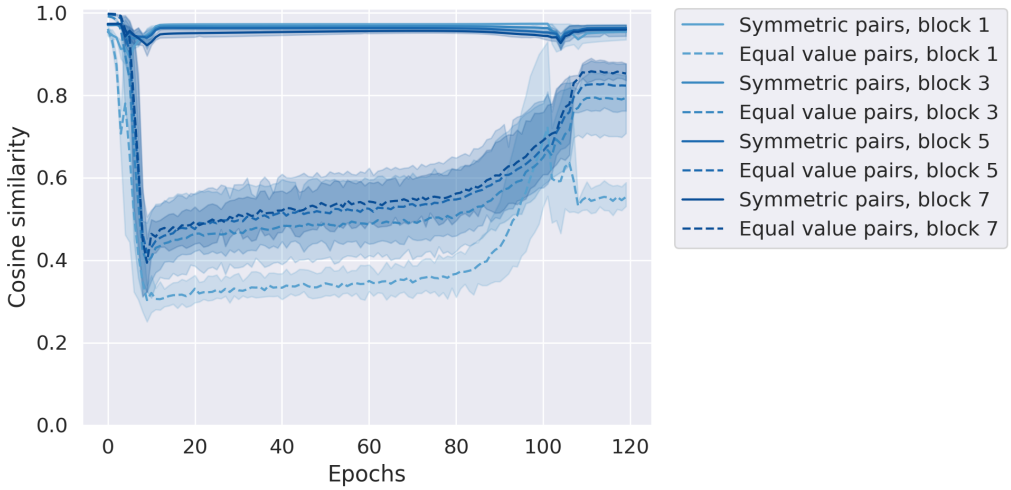


Figure 3: A plot of the symmetric representational similarity (solid lines) and equal value similarity (dashed lines) at different blocks of a decoder only transformer trained on the group $C_{100}$. Shading corresponds to 95% confidence intervals over 5 random initializations.

# TAG-DS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

**The checklist answers are an integral part of your paper submission.** They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "TAG-DS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately reflect the findings of the paper. The main contributions are clearly stated in a bulleted list at the end of the introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations of this work are discussed in a separate "Limitations" section (Section 6).

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: This paper does not contain any theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The details of our experimental set up can be found in Section 3.1, with futher details on our linear probing experiments in Section 3.4. We discuss potential challenges in reproducing results in the Discussion (Section 4) and include the recommendation that many different hyperparameters and initializations should be explored.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We intend to release our code available upon acceptance.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Most of this information can be found in Section 3.1. As discussed in this section, we experimented with a variety of hyperparameters.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Figures include a shaded region representing a 95% confidence interval. In the case where this is not included, the figure caption gives an explanation.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Yes, this information can be found in Section 3.1. As mentioned in the introduction, these experiments are accessible to researchers with small compute resources.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We adhere to the Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: There are many potential societal consequences of using AI to augment or even automate mathematical discovery. We did not think any of these consequences needed to be specifically highlighted here.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: This paper does not use existing assets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release any new assets at this time, although we plan to release our code upon acceptance.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our paper does not include crowdsourcing nor human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development did not involve LLMs.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.