Do Transformers Have the Ability for Periodicity Generalization?

Anonymous ACL submission

Abstract

Large language models (LLMs) based on the Transformer have demonstrated strong performance across diverse tasks. However, current models still exhibit limitations in out-of-distribution (OOD) generalization compared with humans. We investigate this gap through periodicity, one of the basic OOD scenarios. Periodicity captures invariance amid variation. Periodicity generalization represents a model's ability to extract periodic patterns from training data and generalize to OOD scenarios. We introduce a unified interpretation of periodicity from the perspective of abstract algebra and reasoning, including both single and composite periodicity, to analyze why Transformers struggle to generalize periodicity. Then we construct **Coper** about composite periodicity, a controllable generative dataset with two OOD settings, **Hollow** and **Extrapolation**. Experiments reveal that periodicity generalization in Transformers is limited, where models can memorize periodic data during training, but cannot generalize to unseen composite periodicity. We release the source code to support future research (Repository, 2026).

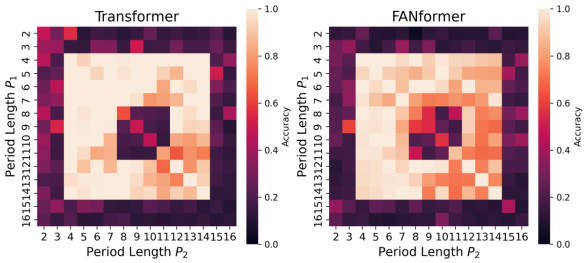

Figure 1: Accuracy heatmaps of Transformer and FANformer on composite periodicity tasks in **Coper**. Models perform well on seen composite periods, but accuracy drops in Hollow and Extrapolation OOD settings, showing limits in periodicity generalization. The full results are shown in Figure 3.

1 Introduction

In recent years, large language models (LLMs) based on the Transformer architecture (Vaswani et al., 2017) have achieved milestone progress in natural language processing (OpenAI, 2023; Google, 2025) and reasoning (DeepSeek-AI et al., 2025; El-Kishky et al., 2025; Chen et al., 2021). Recent studies show that LLMs can even discover new physical laws (Fang et al., 2025) and mathematical theorems (Liu et al., 2025). These achievements lead many researchers to believe that LLMs may eventually reach artificial general intelligence (AGI) (Bubeck et al., 2023).

However, some researchers remain skeptical about the path from LLMs to AGI (Perrigo, 2024). Compared with humans, who can learn complex patterns from limited data with low energy consumption (Sutskever, 2025; Feng et al., 2024), current models still exhibit significant gaps in learning efficiency and out-of-distribution (OOD) generalization (Bender et al., 2021; Song et al., 2025; Wu et al., 2025; Kordi et al., 2025; Hu et al., 2024).

To understand this gap, periodicity provides a key entry point. Periodicity is an essential characteristic that appears widely in nature, mathematics, and human cognition (Buzsáki, 2006; Shapley et al., 1985). It reflects invariance amid variation, from nature (Zalta et al., 2020) to abstract algebra (Dummit and Foote, 2004), and from signal processing (Orfanidis, 1995) to reasoning (Lake et al., 2017; Dong et al., 2025b). Periodicity generalization is the ability to extract periodic patterns from observed data and generalize to unseen scenarios. For humans, it reflects the ability to perform abstract inference from known to unknown. For models, it reflects the ability to generalize learned periodic patterns from training data to OOD scenarios.

Periodicity can be expressed using abstract algebra (Dummit and Foote, 2004; Brin and Stuck, 2002). The essence of periodicity is invariance under specific transformations. When inputs or rules recur in temporal, spatial, or abstract forms, new

instances can be inferred based on previously observed periodic patterns, which is a core feature of reasoning (Dong et al., 2025a).

To this end, we introduce a unified interpretation of periodicity from the perspective of abstract algebra and reasoning, describing periodicity in time-series and reasoning. Our interpretation explains why Transformers struggle to generalize periodicity. From this, we further introduce the concept of **composite periodicity**, which describes the interaction and composition of multiple periodicities.

Then we construct **Coper**, a dataset to characterize **Co**mbinations and transformations of different **per**iodicities, including training and test sets. The test set contains two challenging OOD settings: ❶ **Hollow**: the periods of the test samples remain within the training distribution, but certain period composition is deliberately removed from the training set, to evaluate the model's **interpolation ability**; ❷ **Extrapolation**: the periods of the test samples lie beyond the training distribution, to evaluate the model's **extrapolation ability**.

As shown in Figure 1, experiments show that the Transformers can memorize seen composite periods during training, but fail in OOD settings. Other model architectures (*e.g.,* Mamba, RWKV) and existing periodicity modeling works (*e.g.,* FANformer) exhibit similar failures. We further discover three interesting findings: 1) Transformers with RoPE fail to generalize single periodicity under non-invariant transformations. 2) Transformers struggle to generalize composite periodicity, even when the underlying single periods are seen during training. 3) Increasing data density or model scale improves performance in the Hollow setting, but only by a margin in Extrapolation.

Our contributions are summarized as follows:

1. **A group-theoretic view of periodicity**: proposing a unified interpretation of periodicity, and deriving the boundary of periodicity modeling in Transformers with RoPE under shift-group relative-position invariance.

2. **A dataset for composite periodicity**: constructing a controllable, generative **Coper** dataset with Hollow and Extrapolation scenarios to evaluate models' periodicity generalization ability;

3. **Limits of periodicity generalization**: experiments demonstrate that existing models can memorize seen periods, but struggle to generalize periodicity.

2 Related Work

2.1 Out-of-distribution Generalization

Out-of-distribution (OOD) generalization studies how models generalize learned patterns when the test distribution differs from the training distribution (Shen et al., 2021; Yu et al., 2024). Several works (Wu et al., 2025; Gagnon-Audet et al., 2023) analyze OOD generalization of neural networks in time series and real-world settings, showing that performance can drop sharply under distribution shifts, even when in-distribution accuracy is high. Other works (Nam et al., 2022; Ouellette et al., 2023; de Luca et al., 2024; Huang et al., 2025; Altabaa et al., 2025; Hu et al., 2025; Chang and Bisk, 2025) investigate OOD generalization on logic and reasoning tasks, finding that standard Transformers can fit the training distribution but often fail on variations of the same underlying rules or on much longer inputs.

However, most existing works focus on a single task or a single rule. Generalization across composition of multiple tasks or rules, where models need to compose learned patterns, receives little study (Sinha et al., 2024). Learning by Analogy (Kong et al., 2025) introduces a causal framework for compositional generalization in visual reasoning. Our work treats periodicity as a basic OOD scenario and provides a unified interpretation that links periodic patterns with rule composition.

2.2 Periodicity Modeling

Periodicity is a basic pattern that naturally appears in physical and biological processes. Several works (Wu et al., 2021; Zhou et al., 2022; Woo et al., 2022; Liang et al., 2024; Kui et al., 2025) explicitly model periodicity in time-series forecasting to capture seasonal patterns and long-term dependencies. Autoformer (Wu et al., 2021) decomposes a time series into trend and seasonal components and uses an auto-correlation mechanism to capture periodicity for long-horizon forecasting. FEDformer (Zhou et al., 2022) introduces a frequency-enhanced Transformer that replaces standard attention with Fourier and wavelet blocks to improve long-horizon forecasting.

However, the above approaches only study periodicity in in-distribution settings and do not consider OOD scenarios. FAN (Dong et al., 2025b) extends periodicity modeling to OOD scenarios using a Fourier-based module. FANformer (Dong et al., 2025a) further connects periodicity with rea-

soning, and scales the FAN model to 1B parameters. Through large-scale pretraining of a 1B-parameter model, FANformer (Dong et al., 2025a) integrates FAN into the Transformer architecture, demonstrating that periodicity modeling influences both language modeling and reasoning performance. However, both still struggle with composite periodicity.

3 Unified Interpretation of Periodicity and Reasoning via Group Theory

In this section, we present our unified interpretation of periodicity via group theory. Section 3.1 formalizes the group-theoretic definition of periodicity. Section 3.2 then generalizes the above definition to rule periodicity, grounded in concrete reasoning examples. Section 3.3 explains the limitations of Transformers in generalizing rule periodicity. Finally, Section 3.4 introduces composite periodicity, extending this limitation to composed periodic rules.

3.1 Group-theoretic Definition of Periodicity

Let X be the value space (for example $X = \mathbb{R}$). Consider a discrete sequence defined as a function $f : \mathbb{Z} \to X$. Suppose that there exists $T \in \mathbb{N}^+$ such that

$$f(t+T) = f(t), \quad \forall t \in \mathbb{Z},$$

then we say f admits the sequence period T. Since $T \in \mathbb{N}^+$, we can assume that T is the minimal positive period among all periods without loss of generality. This description shows the periodicity in the temporal dimension.

From the perspective of abstract algebra, the essence of periodicity is to *remain invariant under transformation* (Dummit and Foote, 2004; Brin and Stuck, 2002). Let X be an object space (for example, functions, numbers, or positions), and a group G acts on X. Suppose that there exists a non-identity element $g \in G$ and a minimal integer $n \in \mathbb{N}^+$ such that

$$g^n \cdot x = x, \tag{1}$$

then we say that the element $x \in X$ is **periodic** under the left g-action. The minimal positive period is defined as

$$T(x \mid g) := \min\left\{n \in \mathbb{N}^+ \mid g^n \cdot x = x\right\}. \tag{2}$$

The above definition naturally describes the invariance of periodic functions under group actions. For instance, consider a periodic sequence

$$f : \mathbb{Z} \to \mathbb{R}, \quad f(t+T) = f(t),\ \forall t \in \mathbb{Z},$$

the corresponding shift group is $G = \langle g \rangle$, and the left group action is defined to be[1]

$$g \cdot f \ := \ \phi_g(f) \ : \ t \mapsto f(g^{-1} \cdot t) = f(t-1).$$

Hence, the generator g acts on functions repetitively:

$$\begin{aligned} g^n \cdot f(t) &= g^{n-1} \cdot \phi_g(f)(t) \\ &= g^{n-1} \cdot f(t-1) = \cdots = f(t-n) \end{aligned}$$

Therefore, the minimal positive period under the group action on f is

$$\begin{aligned} &T(f \mid g) := \min\{n \in \mathbb{N}^+ \mid g^n \cdot f = f\} \\ &= \min\{n \in \mathbb{N}^+ \mid f(t-n) = f(t),\ \forall t \in \mathbb{Z}\} \\ &= T, \end{aligned}$$

We call this **sequence periodicity**, the repetitive pattern of the input sequence itself.

Our group-theoretic definition applies not only to sequence periodicity, but also rule periodicity (discussed in Section 3.2).When the mapping relationship is preserved for inputs or rules under transformations, it can also be viewed as invariance under the action of the corresponding transformation group.

3.2 Rule Periodicity in Reasoning

Reasoning tasks often exhibit **rule periodicity**, referring to the repetition of the same rule. We still use the group action interpretation in Section 3.1. Let the rule set be $\mathcal{R} = \{R_k\}_{k \in I}$, the index set be $I = \mathbb{Z}$, and the value space be $V = \{0, \ldots, p-1\}$, and we introduce two types of group:

- **Shift group** $G_\tau = \langle \tau \rangle \simeq \mathbb{Z}$:
  the generator τ acts as $\tau \cdot k = k + 1$.
- **Modulo group** $G_\sigma = \langle \sigma \rangle \simeq \mathbb{Z}_p$:
  the generator σ acts as $\sigma \cdot x = x + 1 \pmod p$.

Now we define two primitive reasoning rules as mappings (with position parameter k):

Digit-wise addition without carry:

$$R_k : \mathbb{Z} \times \mathbb{Z} \to \mathbb{Z}, \qquad (x_k, y_k) \mapsto x_k + y_k.$$

The shift group $G_\tau = \langle \tau \rangle$ acts on the index set I

[1] Here we use the inverse to ensure the associativity in the induced action. Otherwise, if $g \cdot f(t) := f(g \cdot t)$, then $g_1 \cdot (g_2 \cdot f(t)) = g_1 \cdot \phi_{g_2}(f)(t) = \phi_{g_2}(f)(g_1 \cdot t) = f(g_2 \cdot (g_1 \cdot t)) \neq f((g_1 \cdot g_2) \cdot t) = (g_1 \cdot g_2) \cdot f(t)$. The equation does not hold when the group is not abelian. Inverse solves this problem due to $(g_1 \cdot g_2)^{-1} = g_2^{-1} \cdot g_1^{-1}$.

as $\tau \cdot k = k+1$, and the induced action on the rule set $\mathcal{R}$ is

$$\tau \cdot R_k = R_{\tau^{-1} \cdot k} = R_{k-1}, \quad \tau^n \cdot R_k = R_{k-n}.$$

If there exists $T \in \mathbb{N}^+$ such that

$$\tau^T \cdot R_k = R_k, \quad \forall k \in \mathbb{Z},$$

then the rule R_k is said to admit a period of length T along the positional dimension. For digit-wise addition without carry[2], the same rule is applied identically at adjacent positions. Therefore, R_k satisfies $T = 1$.

The shift group G_τ acts on the rule set $\mathcal{R}$, rather than on a specific input sequence, which induces a group action over the entire function space. As a concrete example, consider the computation of $342 + 117$, units: $2 + 7 = 9$; tens: $4 + 1 = 5$; hundreds: $3 + 1 = 4$, resulting in the sequence 459, where the digit-wise addition rule is applied repeatedly at each position.

Single-position modulo:

$$M_k : \mathbb{Z} \to \mathbb{Z}_p, \qquad x_k \mapsto x_k \pmod p.$$

The modulo group $G_\sigma = \langle \sigma \rangle$ acts on the value space V as $\sigma \cdot x = x + 1 \pmod p$. Under this action, the modulo mapping satisfies

$$\begin{aligned} M_k(\sigma^{T_2} \cdot x_k) &= (x_k + T_2) \pmod p \\ &= x_k \pmod p \\ &= M_k(x_k), \end{aligned}$$

so the modulo operation has the period length $T_2 = p$ on the value space.

From the perspective of function space, the modulo operation can be viewed as the result of shift group acting on rule set $\mathcal{R} = \{M_k : \mathbb{Z} \to \mathbb{Z}_p\}$, *i.e.,* the same modulo rule being applied on different position k. This shows the same rule periodicity as in the addition rules. Here the period length for M_k is $T_2' = 1$ on the index set.

In summary, digit-wise addition and single-position modulo operation can be seen as periodically invariant mapping, respectively under the action of **shift group** and **modulo group**. This structural repetition shows the **rule periodicity** in reasoning: the same rule being repeated in position or value space. Rule periodicity reflects the model's capability of reusage and generalization.

[2]Digit-wise addition with carry can be expressed as a combination of the modulo group and the shift group; see Section 3.4 for details on composite periodicity.

3.3 Limitation of Transformers in Periodicity Generalization

3.3.1 Shift Equivariance Preserved by Transformer with RoPE

We first clarify what kinds of periodicity Transformers with RoPE are able to generalize. Let $f : \mathbb{Z} \to X$ be an input sequence, and let $G = \langle g \rangle$ be a shift group.

Proposition. If f satisfies the sequence periodicity $g^T \cdot f = f$, then a Transformer with RoPE preserves the shift equivariance:

$$\mathcal{T}(g^T \cdot f) = g^T \cdot \mathcal{T}(f). \tag{3}$$

The proof follows from the equivariance of position-wise operations (*i.e.,* MLP, activation function, layer normalization, and residual connection), self-attention with RoPE relative positions, and layer composition. Full proof is provided in Appendix B.1.

In particular, when the input sequence satisfies the sequence periodicity $g^T \cdot f = f$, *i.e.,* $f(t) = f(t+T), \forall t \in \mathbb{Z}$, Transformer with RoPE preserves the equivariant structure induced by the corresponding shift-group invariance as characterized in Eq. (3). From the proof, sequence periodicity further implies relative position invariance: for all i, j, we have

$$f(i) - f(j) = f(i+T) - f(j+T). \tag{4}$$

As a result, Transformers with RoPE can capture sequence periodicity satisfying Eq. (4). Without RoPE (*e.g.,* using absolute positional encodings), the periodicity generalization ability of Transformers is substantially weaker. Experimental evidence supporting our proof is provided in the top of Figure 2 (see Appendix B.2 and B.3 for details).

3.3.2 Limitation in Capturing Rule Periodicity

In reasoning tasks, **rule periodicity** is more crucial, *i.e.,* repeating and reusing the same rule on different structures. Rule periodicity satisfies $R_{k+T} = R_k$, which is different from sequence periodicity: a rule R is applied to sequence f and obtains $R_k(f(k))$, and it repeats on every position **without relying on relative position difference** of the sequence. Now we present a concise proof that Transformer fails to represent rule periodicity.

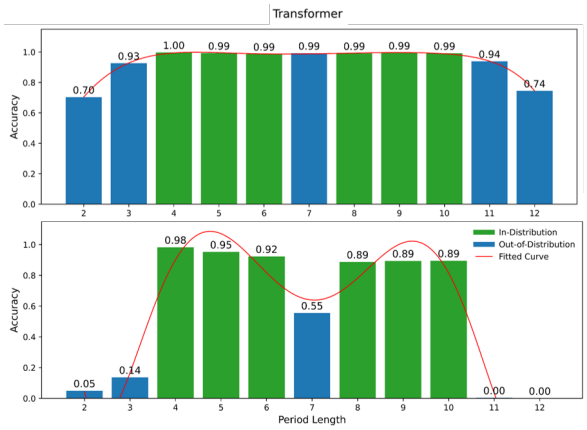

Figure 2: Top: Transformer with RoPE generalizes well for transformations satisfying translation invariance ($f(t+T) = f(t)$). Bottom: Single-period sequences under a non-invariant transformation ($f(t+T) = 2 \cdot f(t)$), showing the generalization failure.

Proof. Assume R_i possesses a period that is different from T, which is the period of f. On one hand, f satisfies $g^T \cdot f = f$, where $\langle g \rangle$ is the shift group acting on the function space. On the other hand, R_i does not exhibit the same shift invariance. In general, the two group actions do not commute with R_i, *i.e.,*

$$\tau^T \cdot R_i \neq R_i \circ g^{-T}. \tag{5}$$

If Eq. (5) does not hold, we would have

$$\begin{aligned} & R_{i+T}(f(i+T)) - R_{j+T}(f(j+T)) \\ = & R_{i+T}(g^{-T} \cdot f(i)) - R_{j+T}(g^{-T} \cdot f(j)) \\ = & \tau^T \cdot R_{i+T}(f(i)) - \tau^T \cdot R_{j+T}(f(j)) \\ = & R_i(f(i)) - R_j(f(j)). \end{aligned}$$

Unfortunately, Eq. (5) holds in general. So for all $i, j \in \mathbb{Z}$, generally we have

$$\begin{aligned} & R_i(f(i)) - R_j(f(j)) \\ \neq & R_{i+T}(f(i+T)) - R_{j+T}(f(j+T)), \end{aligned} \tag{6}$$

which means that rule periodicity cannot be represented by RoPE's relative position information.

Hence, Transformer with RoPE can represent ***sequence periodicity*** satisfying Eq. (4), but cannot represent ***rule periodicity*** according to Eq. (6). Experimental evidence supporting our proof is provided in the bottom of Figure 2 (see Appendix B.4 for details).

3.4 Composite Periodicity

To further explore the limitation of Transformer architecture, we introduce the formal definition of **composite periodicity**. Assume there are two periodic sequences: $f_1 : X_1 \to Y_1, \quad f_2 : X_2 \to Y_2$, and define composite operation:

$$C(f_1(t), f_2(t)) := (M \circ R)(f_1(t), f_2(t)), \tag{7}$$

for example modular addition:

$$C(f_1(t), f_2(t)) = (f_1(t) + f_2(t)) \pmod{p}.$$

We further define the group structure of **basic reasoning modulo rule**. Let $G_R = G_\tau = \langle \tau \rangle$ be the shift group acting on addition rules in position space, and $G_M = G_\sigma = \langle \sigma \rangle$ be the modulo group acting on the modulo operation rules. In the perspective of group theory, the composite group acting on reasoning modulo rules can be presented by direct product:

$$G = G_R \times G_M = \langle \tau \rangle \times \langle \sigma \rangle, \tag{8}$$

and the induced action is

$$\begin{aligned} (\tau^i, \sigma^j) \cdot (R_k, M_k) &= (R_{\tau^{-i} \cdot k}, M_k \circ \sigma^j) \\ &= (R_{k-i}, M_k(\cdot + j)). \end{aligned} \tag{9}$$

If the composite group G defined by Eq. (8) is **non-invariant** according to Eq. (6), Transformers cannot generalize such composite periodicity.

Next, Section 4 constructs datasets based on our interpretation, and Section 5 evaluates the performance of multiple models on various composite periodicity tasks as support.

4 Dataset

To investigate the generalization ability of existing model architectures in composite periodicity scenarios, we construct **Coper**, a fully controllable and generative dataset for composite periodicity. **Coper** contains 50k training samples and 3k test samples. Using our data construction script (see supplementary material), **Coper** can be easily scaled to construct more samples. Moreover, by modifying our script, new composite periodicity tasks (*e.g.,* the convolution) can be constructed.

We compose the rules R_k and M_k introduced in Section 3.4, and define a modulo addition operator C, which is explicitly unfolded along the time dimension as

$$\begin{aligned} & C(f_1(t), f_2(t)) = (M \circ R)(f_1(t), f_2(t)) \\ & = \big(f_1(t \bmod P_1) + f_2(t \bmod P_2)\big) \bmod P. \end{aligned}$$

If the periodicity conditions in Eqs. (1) & (2) are satisfied, such periodicity compositions can also

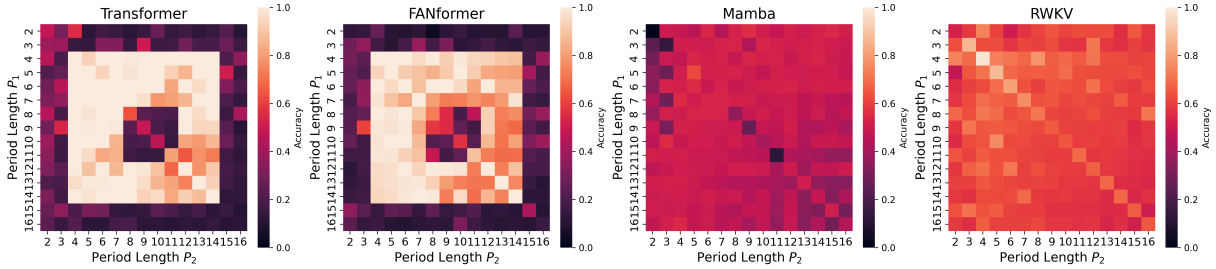

Figure 3: Full accuracy heatmaps across all baseline models on composite periodicity tasks, corresponding to the final models trained as in Figure 5. Notably, even when trained for 1000 epochs, RWKV and Mamba still fail to fit.

be interpreted under our proposed unified interpretation (*e.g.,* convolution tasks; see Section 5.4 for experiments).

We set the training period range as $L \leq P_1, P_2 \leq U$, where L and U denote the lower and upper bounds of the training range, respectively. The training set contains the combinations $\mathcal{P}_{train} = \{(P_1, P_2) \mid P_1, P_2 \in [L, U], (P_1, P_2) \notin \mathcal{P}_{test}^{hollow}\}$.

During dataset construction, we deliberately introduce two types of OOD settings, **Hollow** and **Extrapolation**, to analyze two types of model generalization: interpolation and extrapolation. Each OOD setting contains 1k test samples, in addition to 1k in-distribution test samples. More hyperparameters are listed in Appendix A.

(1) Hollow We define the deliberately held-out hollow combinations within the training range as

$$\mathcal{P}_{test}^{hollow} = \{H_1, H_2, \ldots, H_m\} \subset [L, U]^2,$$

These hollows are located inside the training range $[L, U]$, adjacent to trained combinations; however, they are not included in training. Hollow evaluates the model's interpolation ability to infer unseen composite periodic rules from adjacent, in-distribution combinations.

(2) Extrapolation We define the new combinations outside the training range as

$$\mathcal{P}_{test}^{extra} = \{(P_1, P_2) \mid P_1 \notin [L, U] \text{ or } P_2 \notin [L, U]\}.$$

Extrapolation evaluates the model's extrapolation ability to infer OOD composite periodic rules from the training distribution.

5 Experiments

5.1 Failure of Periodicity Generalization Across Different Architectures

In the first experiment, we evaluate the fitting and generalization abilities of different model architectures about composite periodicity. In addition to

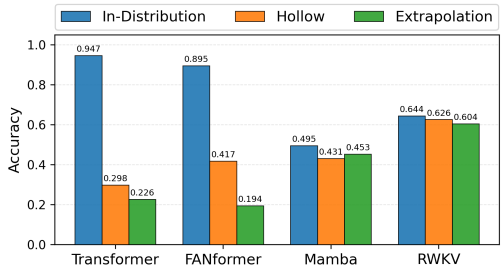

Figure 4: Accuracy of different models on composite periodicity tasks in **Coper** across different OOD settings.

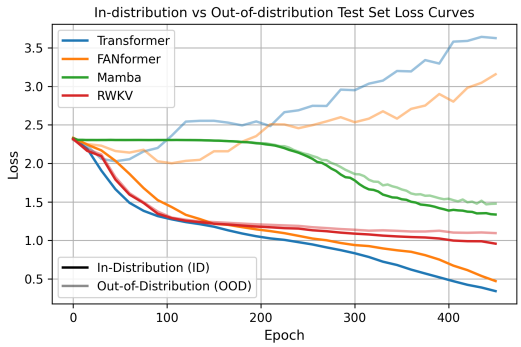

Figure 5: Loss curves on ID and OOD testsets during training.

the Transformer, we include representative architectures for periodicity including FANFormer (Dong et al., 2025a), as well as recently discussed architectures, Mamba and RWKV as baselines. And different models' loss curves on in-distribution and OOD test sets during training are shown in Figure 5. Unless otherwise specified, all models are trained and evaluated using the same random seeds and hyperparameter settings (provided in Appendix A). For evaluation metrics, we report the final test loss on the in-distribution and OOD splits, test accuracy on the in-distribution, Hollow, and Extrapolation splits, together with the overall average. The results are reported in Table 1, Figure 3 and Figure4.

The results show that the Transformer fails to generalize composite periodicity in OOD scenarios. Although the Transformer fits the training distribution well, with 94.7% in-distribution accuracy,

Table 1: Performance comparison across different model architectures on composite periodicity tasks. ID denotes in-distribution.

Model	Testset Loss ↓		ID	Hollow	Extrapolation	Avg.
	ID	OOD				
Transformer	0.34	3.63	94.7	29.8	22.6	49.0
Fanformer	0.47	3.15	89.5	41.7	19.4	50.2
Mamba	1.33	1.48	49.5	43.1	45.3	46.0
RWKV	0.95	1.09	64.4	62.6	60.4	62.5

its accuracy on unseen combinations is low, with only 29.8% on Hollow and 22.6% on Extrapolation. Architectures designed for periodicity modeling do not resolve this limitation. FANFormer also achieves high in-distribution accuracy at 89.5%, while performance on Hollow and Extrapolation remains low at 41.7% and 19.4%. In contrast, state-based models show much weaker fitting under the same training budget. Importantly, OOD generalization can only be interpreted when in-distribution fitting is sufficient. RWKV reaches 62.6% on Hollow and 60.4% on Extrapolation, but in-distribution accuracy is only 64.4%, far below the Transformer. Overall, existing architectures fail to support both accurate in-distribution fitting and OOD generalization simultaneously, and none truly learn the composite periodicity R.

5.2 Effect of Data Density OOD Generalization

In the second experiment, we further investigate the effect of training data density on OOD generalization. Intuitively, the Hollow lies within the training distribution, and the neighboring period combinations of the Hollow are all observed during training. A natural question is whether reducing the hollow size, *i.e.,* making the training data denser, enables models to fill the missing combinations via interpolation. To this end, while keeping the number of training samples fixed, we shrink the period range from $[2, 16]$ to a denser range of $[2, 11]$ to explicitly control the hollow size, retaining only a hollow combination $H = (6, 7) \cup (7, 7)$.

The results are shown in Figure 6 and Figure12. Increasing training data density improves interpolation in the Hollow setting. For example, the Transformer achieves 56.5% on Hollow, compared with 29.8% reported in Table 1, indicating an increase of +26.7% through denser training data. However, performance on Extrapolation remains largely unchanged (22.7% vs 22.6%). FANFormer exhibit similar trends: higher accuracy in Hollow but minimal margin in Extrapolation. These findings indicate that denser training data improves performance in the Hollow, but has minimal margin on Extrapolation, showing that models do not learn the composite periodicity R. More details are shown in Appendix D.

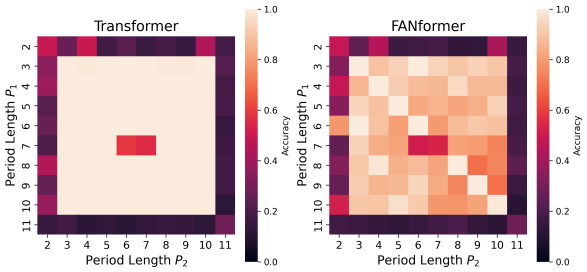

Figure 6: Accuracy heatmaps with increased training data density.

5.3 Scaling Model and Data Size for Generalization

In the third experiment, we study the effect of model scale on generalization of composite periodicity. The previous experiments show that, under a fixed model scale, both different architectures and different data densities suffer from clear generalization failures in the Hollow and Extrapolation settings. A natural question is whether these failures are caused by insufficient model capacity. We gradually increase the number of layers to scale model parameters and evaluate the generalization performance in the Hollow and Extrapolation settings. All models are trained for the same number of epochs on identical training data.

The results of scaling model parameters are reported in Table 4. Increasing the number of layers consistently improves generalization in both Hollow and Extrapolation settings. For instance, the Transformer improves from 29.8% to 57.2% on Hollow and from 22.6% to 32.7% on Extrapolation when scaling from 3 to 7 layers. FANFormer show similar trends, with OOD loss decreasing from 3.21 to 2.09. These results show that larger models improve OOD generalization for composite periodicity. However, Extrapolation performance remains below in-distribution accuracy, and such improvement comes at the cost of computational overhead, which will be discussed in Section 6. More details are shown in Appendix E.

5.4 Failure under Other Composite Periodicity

Previous experiments analyzed the failure of generalization for composite periodicity from the per-

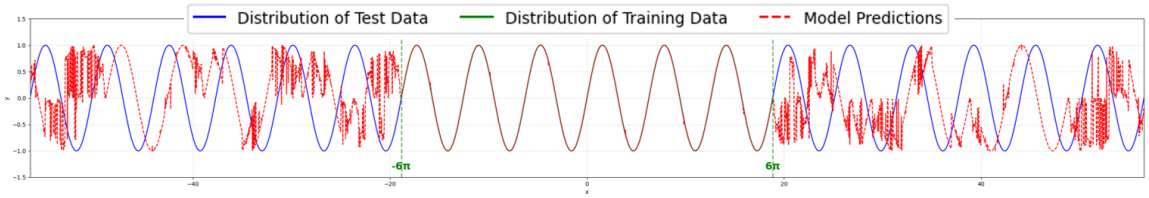

Figure 7: Although Transformer fits the ID training data $y = \sin(x)$ within the range $[-6\pi, +6\pi]$ well, it struggles to preserve the same periodicity when extrapolating, where x is a token sequence.

Table 2: Performance comparison across different model architectures on circular convolution.

Model	Testset Loss ↓		ID	Hollow	Extrapolation	Avg.
	ID	OOD				
Transformer	0.42	3.16	80.6	15.9	17.8	38.1
Fanformer	1.24	1.66	47.4	34.6	24.9	35.6
Mamba	1.99	2.05	17.7	15.4	17.2	16.8
RWKV	2.11	2.15	14.2	11.9	12.1	12.7

spectives of model architecture, data density, and model scale. However, they are based on the same composite periodicity. A key question is whether the observed failure depends on the specific rule periodicity or represents a more general phenomenon, as suggested in Section 3.3. Prior work conducted experiments on carry-addition tasks (Hu et al., 2024). In our final experiment, we use the **circular convolution** task as a new composite periodicity rule to further evaluate OOD generalization. The group-theoretic interpretation of circular convolution is provided in Appendix C.

The results are reported in Table 2. Under circular convolution, models exhibit the same generalization failure observed in previous experiments. The Transformer achieves 80.6% in-distribution accuracy but performs poorly on Hollow and Extrapolation, with accuracies of 15.9% and 17.8%, respectively. State-based models like Mamba and RWKV fail to fit the training distribution within the same training budget, reflecting low learning efficiency. Overall, the generalization failure persists across more composite periodicity scenarios, rather than being specific to a unique case. More details are shown in Appendix F and Appendix G.

6 Discussion

6.1 More Types of Periodicity

Beyond sequence and rule periodicity, there exist **hidden periodicity** patterns, where individual samples do not show obvious repetition, and the periodicity only appears at the level of the full data distribution. An example is the sine function. Rather than being contained in one piece of data (*e.g.,* $12341234\cdots$), the periodic patterns in the sine function can be learned only after seeing many the (x, y) data points across the distribution. We conduct an experiment to train a Transformer to fit $y = \sin(x)$. Consistent with **Coper**, we divide the data into in-distribution $[-6\pi, +6\pi]$ for training and OOD beyond this range. As shown in Figure 7, the Transformer fails to generalize such periodicity. More experimental details are provided in Appendix H. In real-world tasks, periodicity appears as combinations of different types.

6.2 Hollow, Data Density, and Scaling Behavior

OOD performance is influenced by data density and model size, which helps explain "grokking" behavior observed during scaling (Power et al., 2022). Larger models and denser data enable stronger generalization beyond the training distribution. However, our group-theoretic interpretation and experiments show that increasing data density or model size does not enable rule generalization. Even with minimal Hollow size, missing periodic combinations remain unresolved. As large-scale pretraining data approaches saturation, further improvements require exponentially higher training cost or model parameters. Progress beyond Hollow and Extrapolation therefore requires model architectures to have at least the ability to represent reasonable OOD generalization, like periodicity.

7 Conclusion

Our work proposes a unified interpretation of periodicity and reasoning through group theory to explain why Transformers struggle to generalize periodicity. Based on this, we further propose the composite periodicity dataset **Coper** to enable controlled evaluation of periodicity generalization. Experiments show that current models fail in Hollow and Extrapolation scenarios, revealing the limitations of current architectures in periodicity generalization.

Limitation

Our limitations include 1) not studying external reasoning mechanisms such as chain-of-thought (CoT) or memory, and 2) applying our interpretation only to Transformer with RoPE, without more models like Mamba. Future works include 1) extending the group-theoretic interpretation to more reasoning scenarios and 2) designing architectures that can model composite periodicity.

References

Awni Altabaa, Siyu Chen, John Lafferty, and Zhuoran Yang. 2025. Unlocking out-of-distribution generalization in transformers via recursive latent space reasoning. *CoRR*, abs/2510.14095.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *FAccT*, pages 610–623. ACM.

Michael Brin and Garrett Stuck. 2002. *Introduction to dynamical systems*. Cambridge university press.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott M. Lundberg, Harsha Nori, Hamid Palangi, Marco Túlio Ribeiro, and Yi Zhang. 2023. Sparks of artificial general intelligence: Early experiments with GPT-4. *CoRR*, abs/2303.12712.

György Buzsáki. 2006. *Rhythms of the Brain*. Oxford university press.

Yingshan Chang and Yonatan Bisk. 2025. Language models need inductive biases to count inductively. In *ICLR*. OpenReview.net.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. Evaluating large language models trained on code. *CoRR*, abs/2107.03374.

Artur Back de Luca, George Giapitzakis, Shenghao Yang, Petar Velickovic, and Kimon Fountoulakis. 2024. Positional attention: Out-of-distribution generalization and expressivity for neural algorithmic reasoning. *CoRR*, abs/2410.01686.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 81 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948.

Yihong Dong, Ge Li, Xue Jiang, Yongding Tao, Kechi Zhang, Lecheng Wang, Hao Zhu, Huanyu Liu, jiazheng ding, Jia Li, Jinliang Deng, and Hong Mei. 2025a. Reasoning is periodicity? improving large language models through effective periodicity modeling. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Yihong Dong, Ge Li, Yongding Tao, Xue Jiang, Kechi Zhang, Jia Li, Jinliang Deng, Jing Su, Jun Zhang, and Jingjing Xu. 2025b. Fourier analysis network. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

David S Dummit and Richard M Foote. 2004. Abstract algebra. john wile & sons. *Inc., Hoboken, NJ*.

Ahmed El-Kishky, Alexander Wei, Andre Saraiva, Borys Minaiev, Daniel Selsam, David Dohan, Francis Song, Hunter Lightman, Ignasi Clavera Gilaberte, Jakub Pachocki, Jerry Tworek, Lorenz Kuhn, Lukasz Kaiser, Mark Chen, Max Schwarzer, Mostafa Rohaninejad, Nat McAleese, o3 contributors, Oleg Mürk, and 5 others. 2025. Competitive programming with large reasoning models. *CoRR*, abs/2502.06807.

You-Le Fang, Dong-Shan Jian, Xiang Li, and Yan-Qing Ma. 2025. AI-newton: A concept-driven physical law discovery system without prior physical knowledge. *CoRR*, abs/2504.01538.

Tao Feng, Chuanyang Jin, Jingyu Liu, Kunlun Zhu, Haoqin Tu, Zirui Cheng, Guanyu Lin, and Jiaxuan You. 2024. How far are we from AGI: are llms all we need? *Trans. Mach. Learn. Res.*, 2024.

Jean-Christophe Gagnon-Audet, Kartik Ahuja, Mohammad-Javad Darvishi Bayazi, Pooneh Mousavi, Guillaume Dumas, and Irina Rish. 2023. WOODS: benchmarks for out-of-distribution generalization in time series. *Trans. Mach. Learn. Res.*, 2023.

Team Google. 2025. Gemma 3 technical report. *CoRR*, abs/2503.19786.

Yi Hu, Shijia Kang, Haotong Yang, Haotian Xu, and Muhan Zhang. 2025. Beyond single-task: Robust multi-task length generalization for llms. *Preprint*, arXiv:2502.11525.

Yi Hu, Xiaojuan Tang, Haotong Yang, and Muhan Zhang. 2024. Case-based or rule-based: How do transformers do the math? In *ICML*. OpenReview.net.

Xinting Huang, Andy Yang, Satwik Bhattamishra, Yash Raj Sarrof, Andreas Krebs, Hattie Zhou, Preetum Nakkiran, and Michael Hahn. 2025. A formal framework for understanding length generalization in transformers. In *ICLR*. OpenReview.net.

Lingjing Kong, Shaoan Xie, Yang Jiao, Yetian Chen, Yanhui Guo, Simone Shao, Yan Gao, Guangyi Chen, and Kun Zhang. 2025. Learning by analogy: A causal framework for composition generalization. *Preprint*, arXiv:2512.10669.

Yeganeh Kordi, Nihal V. Nayak, Max Zuo, Ilana Nguyen, and Stephen H. Bach. 2025. Revisiting generalization across difficulty levels: It's not so easy. *Preprint*, arXiv:2511.21692.

Xiaoyan Kui, Canwei Liu, Qinsong Li, Zhipeng Hu, Yangyang Shi, Weixin Si, and Beiji Zou. 2025. TFKAN: time-frequency KAN for long-term time series forecasting. *CoRR*, abs/2506.12696.

Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. 2017. Building machines that learn and think like people. *Behavioral and brain sciences*, 40:e253.

Daojun Liang, Haixia Zhang, Dongfeng Yuan, and Minggao Zhang. 2024. Periodformer: An efficient long-term time series forecasting method based on periodic attention. *Knowl. Based Syst.*, 304:112556.

Yuanhang Liu, Yanxing Huang, Yanqiao Wang, Peng Li, and Yang Liu. 2025. AI mathematician: Towards fully automated frontier mathematical research. *CoRR*, abs/2505.22451.

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljacic, Thomas Y. Hou, and Max Tegmark. 2024. KAN: kolmogorov-arnold networks. *CoRR*, abs/2404.19756.

Andrew J. Nam, Mustafa Abdool, Trevor Maxfield, and James L. McClelland. 2022. Achieving and understanding out-of-distribution generalization in systematic reasoning in small-scale transformers. *Preprint*, arXiv:2210.03275.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Sophocles J Orfanidis. 1995. *Introduction to signal processing*. Prentice-Hall, Inc.

Simon Ouellette, Rolf Pfister, and Hansueli Jud. 2023. Counting and algorithmic generalization with transformers. *CoRR*, abs/2310.08661.

Billy Perrigo. 2024. Meta's chief ai scientist yann lecun on agi, open-source, and ai risk. *TIME*.

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

Anonymous Repository. 2026. Periodicity generalization project repository. `https://anonymous.4open.science/r/Periodicity-Generalization-EF06`.

Robert Shapley, Peter Lennie, and 1 others. 1985. Spatial frequency analysis in the visual system. *Annual review of neuroscience*, 8(1):547–581.

Zheyan Shen, Jiashuo Liu, Yue He, Xingxuan Zhang, Renzhe Xu, Han Yu, and Peng Cui. 2021. Towards out-of-distribution generalization: A survey. *CoRR*, abs/2108.13624.

Sania Sinha, Tanawan Premsri, and Parisa Kordjamshidi. 2024. A survey on compositional learning of AI models: Theoretical and experimental practices. *Trans. Mach. Learn. Res.*, 2024.

Peiyang Song, Pengrui Han, and Noah Goodman. 2025. A survey on large language model reasoning failures. In *2nd AI for Math Workshop@ ICML 2025*.

Ilya Sutskever. 2025. We're moving from the age of scaling to the age of research. YouTube video. Uploaded by Dwarkesh Patel.

Ryotaro Takehara. 2025. *Comparative evaluation of novel neural network architectures*. Ph.D. thesis, The University of Texas at Austin.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, pages 5998–6008.

Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven C. H. Hoi. 2022. Etsformer: Exponential smoothing transformers for time-series forecasting. *CoRR*, abs/2202.01381.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *NeurIPS*, pages 22419–22430.

Xin Wu, Fei Teng, Xingwang Li, Ji Zhang, Tianrui Li, and Qiang Duan. 2025. Out-of-distribution generalization in time series: A survey. *CoRR*, abs/2503.13868.

Han Yu, Jiashuo Liu, Xingxuan Zhang, Jiayun Wu, and Peng Cui. 2024. A survey on evaluation of out-of-distribution generalization. *CoRR*, abs/2403.01874.

Arnaud Zalta, Spase Petkoski, and Benjamin Morillon. 2020. Natural rhythms of periodic temporal attention. *Nature communications*, 11(1):1051.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. 2022. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *ICML*, volume 162 of *Proceedings of Machine Learning Research*, pages 27268–27286. PMLR.

A Experimental Details of Training and Dataset

Table 3: Training hyperparameters and dataset parameters.

Training Hyperparameters	
Batch size	32
Number of epochs	450
Learning rate	1×10^{-5}
Weight decay	0.01
Hidden state size	896
Model Layers	$3 \sim 7$
Dataset Parameters	
Number of training samples	50,000
Number of test samples	3,000
Sequence period range (train)	$[4, 14]$
Total range	$[2, 16]$
In-distribution pairs	$(P_1, P_2) \in [4, 14]^2 \setminus$ Hollow
Hollow test pairs	$P_1, P_2 \in [8, 11]$
Extrapolation test pairs	$(P_1$ or $P_2 \notin [4, 14])$
Output token	$0 \sim 9$

Dataset Construction. Training sequences are sampled such that

$$(P_1, P_2) \in [4, 14]^2 \setminus \text{Hollow},$$

where **Hollow** defines a held-out region within the training range for interpolation evaluation.

The test set contains two types of OOD scenarios:

- **Hollow**: combinations within $[4, 14]^2$ but excluded from training, *i.e.,* $P_1, P_2 \in [8, 11]$.
- **Extrapolation**: combinations where P_1 or P_2 is outside the training range $[4, 14]$, *e.g.,* $(1, 16), (2, 15)$, to evaluate extrapolation ability.

Each sequence is generated by sampling two sequences of length P_1 and P_2, repeating them to the least common multiple length, summing element-wise modulo 10, and concatenating as `sequence1 + '+' + sequence2 + '='`. The next-token prediction task is then defined over this sequence.

For example, let $f_1(t) = (1, 2, 3)$ with period $P_1 = 3$ and $f_2(t) = (1, 2)$ with period $P_2 = 2$. Then, their composition via Eq. (7) produces

$$\begin{aligned} &C(f_1(t), f_2(t)) \\ &= (f_1(t \bmod 3) + f_2(t \bmod 2)) \bmod 10 \\ &= (2, 4, 4, 3, 3, 5, \dots). \end{aligned}$$

We choose to use the pretrained Qwen2.5 tokenizer and embedding and keep them fixed during training to avoid introducing spurious numeric priors. Our prior experiments show that training embeddings from scratch on numeric tasks inadvertently allows the model to capture numeric magnitude information directly, which can lead to shortcut solutions and hack the intended evaluation of reasoning capability. By using fixed pretrained embeddings, we ensure that models rely on learned reasoning rather than trivial numeric cues, enabling a fair comparison across all models. All experiments are conducted on 8×A100 (40GB) GPUs for about few hours.

We train the model three times using fixed random seeds, evaluate each run separately, and report the averaged results. And for each model, we compute accuracy by performing token-wise comparisons between the model's predicted sequence and the ground-truth sequence, and averaging the correctness over all positions and all samples in the dataset.

B Further Analysis: Capacity and Limitations of Transformer with RoPE in Periodicity Modeling

B.1 Equivariance of Layer Normalization, Residual Connections, and Layer Composition

This appendix provides the detailed proof of Proposition (3) by showing that all core Transformer components preserve shift equivariance under the shift-group action. The proof consists of two parts. First, position-wise operations, including MLP, Layer Normalization, and residual connections, preserve shift equivariance since the same parameters are shared across all token positions. Second, self-attention preserves shift equivariance under the sequence periodicity condition $g^T \cdot f = f$ together with the relative position invariance introduced by RoPE.

Proof.

❶ **Position-wise operations.** MLP, activation, normalization, and residual connection are all position-wise (or token-wise) operations in Transformer, *i.e.,* they are independently applied to each position while sharing the same parameters across all positions. For example, the MLP satisfies

$$\text{MLP}(f(i)) = \text{Act}(f(i)W_1 + b_1)W_2 + b_2.$$

Under the shift group action in Section 3.1,

$$\begin{aligned}&\mathrm{MLP}((g^T \cdot f)(i))\\ &= \mathrm{Act}(f(i-T)W_1 + b_1)W_2 + b_2\\ &= \mathrm{MLP}(f(i-T))\\ &= (g^T \cdot \mathrm{MLP}(f))(i).\end{aligned}$$

Therefore, position-wise operations preserve the same shift equivariance property. Since activation functions, normalization layers, and residual connections are also independently applied to each token and do not depend on the position index, the same shift equivariance property holds.

❷ Residual Connection. For a residual block,

$$\mathrm{Res}(f)(i) = f(i) + F(f)(i),$$

where F denotes a Transformer sublayer (e.g., MLP or self-attention).

Assume F is shift equivariant, i.e.,

$$F(g^T \cdot f) = g^T \cdot F(f).$$

Then,

$$\begin{aligned}\mathrm{Res}(g^T \cdot f)(i) &= (g^T \cdot f)(i) + F(g^T \cdot f)(i)\\ &= f(i-T) + F(f)(i-T)\\ &= \mathrm{Res}(f)(i-T)\\ &= (g^T \cdot \mathrm{Res}(f))(i).\end{aligned}$$

Thus,

$$\mathrm{Res}(g^T \cdot f) = g^T \cdot \mathrm{Res}(f).$$

❸ Layer Normalization. Layer Normalization is applied independently to each token:

$$\mathrm{LN}(f(i)) = \gamma \frac{f(i) - \mu(f(i))}{\sigma(f(i))} + \beta,$$

where $\mu(\cdot)$ and $\sigma(\cdot)$ are computed over feature dimensions and are independent of the position index i.

Under the shift group action,

$$(g^T \cdot f)(i) = f(i-T),$$

we have

$$\begin{aligned}\mu((g^T \cdot f)(i)) &= \mu(f(i-T)),\\ \sigma((g^T \cdot f)(i)) &= \sigma(f(i-T)).\end{aligned}$$

Therefore,

$$\begin{aligned}&\mathrm{LN}((g^T \cdot f)(i))\\ &= \gamma \frac{f(i-T) - \mu(f(i-T))}{\sigma(f(i-T))} + \beta\\ &= \mathrm{LN}(f(i-T))\\ &= (g^T \cdot \mathrm{LN}(f))(i).\end{aligned}$$

Hence,

$$\mathrm{LN}(g^T \cdot f) = g^T \cdot \mathrm{LN}(f).$$

❹ Self-attention. Let $Q_i = f(i) \cdot W_Q, K_i = f(i) \cdot W_K, V_i = f(i) \cdot W_V$. Since the linear projections are independently applied to each token,

$$Q_{i-T} = (g^T \cdot f)(i)W_Q = (g^T \cdot Q)(i).$$

and similarly for K_i and V_i.

Consider two positions (i, j) and their translated positions $(i+T, j+T)$. Under RoPE, the attention score depends on both token content and relative positional encoding,

$$s_{i,j} = \frac{Q_i K_j^\top}{\sqrt{d}} = h(f(i), f(j), \mathrm{RoPE}(i,j)).$$

Since $(i+T) - (j+T) = i - j$, the relative positional encoding remains unchanged:

$$\mathrm{RoPE}(i+T, j+T) = \mathrm{RoPE}(i,j).$$

We obtain

$$\begin{aligned}&s_{i,j} = h(f(i), f(j), \mathrm{RoPE}(i,j))\\ &= h(f(i+T), f(j+T), \mathrm{RoPE}(i+T, j+T))\\ &= s_{i+T,j+T}.\end{aligned}$$

The attention weight is

$$\alpha_{i,j} = \frac{\exp(s_{i,j})}{\sum_k \exp(s_{i,k})} = \alpha_{i+T,j+T}.$$

The attention output is

$$\mathrm{Attn}(f)(i) = \sum_j \alpha_{i,j} V_j.$$

Then,

$$\begin{aligned}\mathrm{Attn}(g^T \cdot f)(i) &= \sum_j \alpha_{i-T,j-T} V_{j-T}\\ &= \sum_k \alpha_{i-T,k} V_k = \mathrm{Attn}(f)(i-T)\\ &= (g^T \cdot \mathrm{Attn}(f))(i).\end{aligned}$$

❺ Composition of Layers. A Transformer is a composition of layers:

$$\mathcal{T} = F_L \circ \cdots \circ F_1.$$

We prove by induction that each composition preserves shift equivariance.

Base case. For F_1,

$$F_1(g^T \cdot f) = g^T \cdot F_1(f).$$

Inductive step. Assume

$$F_{k-1} \circ \cdots \circ F_1(g^T \cdot f) = g^T \cdot (F_{k-1} \circ \cdots \circ F_1)(f).$$

Then,

$$\begin{aligned} &F_k \circ \cdots \circ F_1(g^T \cdot f) \\ &= F_k\big(g^T \cdot (F_{k-1} \circ \cdots \circ F_1)(f)\big) \\ &= g^T \cdot F_k\big((F_{k-1} \circ \cdots \circ F_1)(f)\big) \\ &= g^T \cdot (F_k \circ \cdots \circ F_1)(f). \end{aligned}$$

Thus,

$$\mathcal{T}(g^T \cdot f) = g^T \cdot \mathcal{T}(f).$$

Therefore, the complete Transformer preserves shift equivariance.

B.2 Formal Illustration of RoPE and Single Period Shift Invariance

In Section 3.3, we dive into sequence periodicity and rule periodicity. Here we analyze how Transformer models **simple sequence periodicity** satisfying Eq. (4) through RoPE, for example the time shift invariance of a single periodic sequence $f(t) = f(t+T)$.

The core idea of RoPE is to map time step t onto the phase rotation in the complex plane:

$$\mathrm{RoPE}(x_t) = x_t e^{i\theta t}, \quad \theta = \frac{2\pi}{T},$$

where T is the implicit period length. When a time shift Δt occurs on the input sequence, we have

$$\begin{aligned} \frac{\mathrm{RoPE}(x_{t+\Delta t})}{\mathrm{RoPE}(x_t)} &= \frac{x_{t+\Delta t} \cdot e^{i\theta(t+\Delta t)}}{x_t \cdot e^{i\theta t}} \\ &= \frac{x_{t+\Delta t}}{x_t} \cdot e^{i\theta \Delta t}, \end{aligned}$$

and if the periodicity ensures $x_{t+\Delta t} = x_t$, then

$$\mathrm{RoPE}(x_{t+\Delta t}) = e^{i\theta \Delta t} \cdot \mathrm{RoPE}(x_t),$$

hence shift equivariance remains in the self-attention computation:

$$\begin{aligned} &\langle \mathrm{RoPE}(x_{t+\Delta t}), \mathrm{RoPE}(x_{s+\Delta t}) \rangle \\ &= \langle \mathrm{RoPE}(x_t), \mathrm{RoPE}(x_s) \rangle. \end{aligned}$$

This means that RoPE can realize the sequence shift invariance under single period T, therefore captures the simple **sequence periodic pattern** of the input sequence. We conduct an experiment to compare the ability of modeling single periodicity between RoPE and traditional absolute positional encoding (*i.e.,* Sinusoidal Positional Encoding, SinPE) in Appendix B.3.

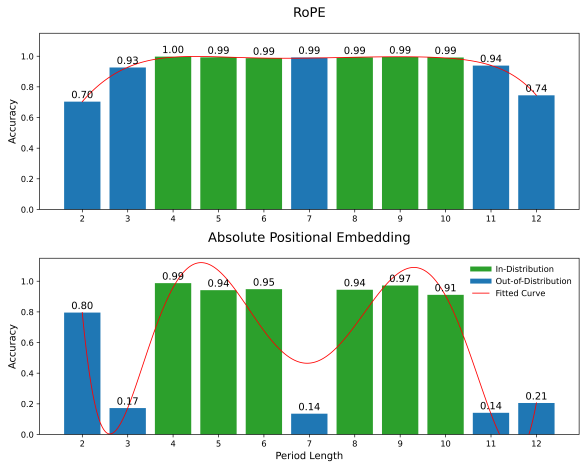

Figure 8: Comparison of periodicity generalization between RoPE and absolute positional encoding.

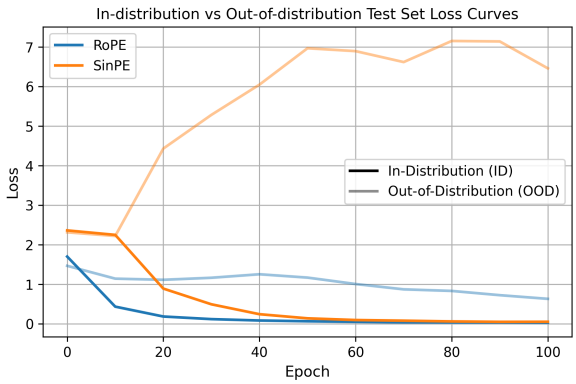

Figure 9: Loss curves for models with RoPE and absolute positional encoding on single-periodicity generalization tasks. The SinPE means Sinusoidal Positional Encoding, which is an absolute positional encoding in the original Transformer.

B.3 Experimental Validation of RoPE Modeling Single Periodicity

In Appendix B.2, we analyze how Transformer models simple sequence periodicity through RoPE.

To prove that RoPE contributes to this, we conduct an experiment to compare RoPE with absolute positional encoding on the task of generalizing single periodicity.

We construct a periodic sequence continuation task. Given a periodic numerical sequence with more than two cycles, the model is required to predict the subsequent tokens. For example, here is an example for $T = 6$, where a single cycle is `955884`:

Input:	`9558849558849`
Target output:	`55884955884`

The datasets are constructed as follows:

- **Training Set (In-distribution)**: Sequences with periods $T \in \{4, 5, 6, 8, 9, 10\}$, totaling 10,000 samples.
- **Test Set (OOD)**: Totaling 3,000 samples.
  - **Hollow**: Sequences with periods $T = 7$.
  - **Extrapolation**: Sequences with periods $T \in \{2, 3, 11, 12\}$.

We compare two transformer architectures: one with RoPE and another with absolute positional encoding. Both models are trained until convergence (100 epochs). The accuracy we use to measure the performance is at the token level, which is the ratio of correctly predicted tokens across all predicted ones.

The result is shown in Fig. 8, where the two graphs represent the performance of two positional encoding, and the two colors stand for in-distribution and OOD generalization ability. Fig. 11 reports loss of both positional encodings on in-distribution and OOD datasets during training. We draw the following conclusions:

In the in-distribution scenario ($T \in \{4, 5, 6, 8, 9, 10\}$), RoPE and absolute positional encoding both achieve near 1.00 accuracy, indicating their common ability to memorize the seen periodic patterns. In the case of hollow interpolation ($T = 7$), RoPE remains a high accuracy, but the performance of absolute positional encoding drops significantly. In extrapolation cases ($T \in \{2, 3, 11, 12\}$), the accuracy of RoPE declines at a certain degree. On the contrast, the accuracy of absolute positional encoding is far less than RoPE's. The failure of absolute positional encoding on generalizing OOD cases proves its incompetence in modeling single periodicity. Notably, when $T = 2$, absolute positional encoding gains accuracy ≈ 0.80, which is due to the features of absolute positional encoding. It captures low-range parity information, and gains advantages through even period lengths in training sets, which is not an evidence to the extrapolation ability of absolute positional encoding.

The result confirms the theoretical conclusion in Appendix B.2. By utilizing relative positional invariance compared to absolute positional encoding, RoPE has the ability to model single periodic patterns and make effective generalization. However, although RoPE performs well in generalizing single periodicity, it fails to generalize when facing composite periodicity as in Section 3.3.2.

B.4 The Failure of RoPE Modeling Single Periodicity under Non-invariant Transformation

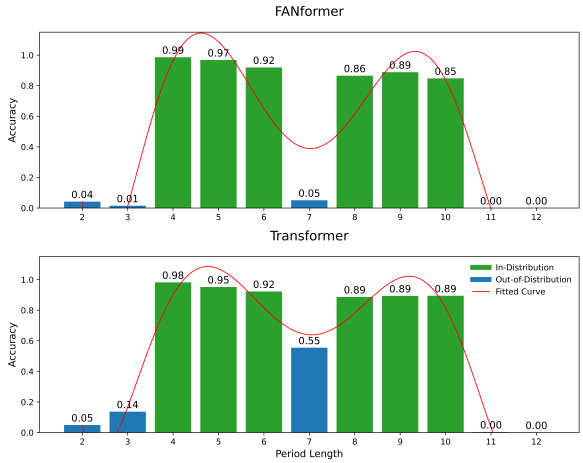

Figure 10: Prediction accuracy on single-period sequences under a non-invariant transformation ($f(t + T) = 2 \cdot f(t)$) showing Transformer's failure with RoPE.

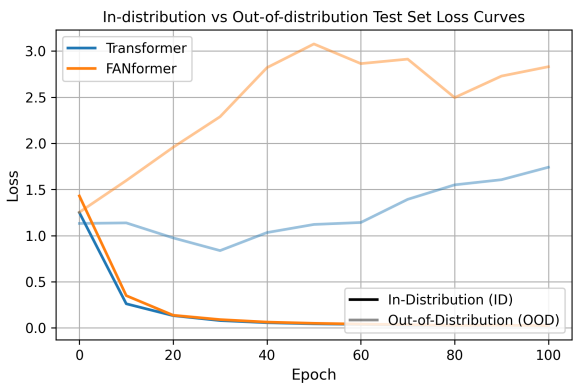

Figure 11: Loss curves for Transformer and FANformer with RoPE on single-period sequences under non-invariant transformations.

This appendix presents a counterexample showing that RoPE fails to model single periodicity when the underlying transformation violates relative positional invariance.

Consider a single-period sequence continuation task with period length T. The periodic transformation is defined as

$$f(t+T) = 2 \cdot f(t),$$

which induces a scale change across periods. Under this transformation, the relative positional relation is no longer preserved Eq.(6). Therefore, the core assumption required for RoPE-based periodic modeling does not hold.

We train a Transformer equipped with RoPE on this task using the same training protocol as in the successful single-periodicity experiment in Appendix B.3. The model fails to learn the periodic continuation rule. As shown in Fig. 10, the prediction accuracy remains low across all evaluation settings. The training loss curve in Fig. 11 further indicates that the model does not converge to a stable solution.

These results demonstrate that RoPE cannot model single periodicity when the periodic structure involves non-invariant transformations. RoPE relies on relative positional consistency across cycles. When such consistency is violated, RoPE fails to support effective learning and generalization, even in the single-period setting.

This counterexample provides further verification of Eq. (6). Regardless of how the equation becomes invalid, whether by adding a composite periodicity as in Section 3.3.2 or by scaling the original periodic sequence, the model consistently fails to generalize the periodicity. Hence, the relative position invariance instilled by RoPE is crucial for periodicity generalization.

C Group-Theoretic Interpretation of Circular Convolution

In Section 5.4, we conduct an experiment utilizing circular convolution as another example of composite periodicity. Here we provide a theoretic explanation, demonstrating how circular convolution effectively represents composite periodicity.

Let $f_1, f_2 : \mathbb{Z}_N \to \mathbb{R}$ be two input sequences with period P_1, P_2, where N is a common multiple of P_1 and P_2, usually $N = \mathrm{lcm}(P_1, P_2)$. The circular convolution of f_1, f_2 is defined as

$$(f_1 * f_2)(t) := \sum_{n=0}^{N-1} f_1(n) \cdot f_2(t-n), \tag{10}$$

where n and $t-n$ are calculated under modulo P_1 and P_2 respectively.

Now we use the group $G = \langle g \rangle \simeq \mathbb{Z}_N$ to induce an action on the sequence space $\mathcal{F} = \{f \mid f : \mathbb{Z}_N \to \mathbb{R}\}$. The group action is defined as

$$g \cdot t = t + 1 \pmod N.$$

The induced action is[3]

$$g \cdot f(t) = f(g^{-1} \cdot t) = f(t - 1 \bmod N) = f(t-1).$$

Hence, $g^n \cdot f(t) = f(g^{-n} \cdot t) = f(t-n)$.

For $i = 1, 2$, assume $\mathcal{G}_{P_i} := \langle g^{P_i} \rangle$, since f_i has a period of P_i, for all $\gamma \in \mathcal{G}_{P_i}$ we have $\gamma \cdot f_i = f_i$, which implies the sequence periodicity. Because $P_i \mid N$, we have $g^n \cdot f_i(t) = f_i(t-n) = f_i(t - n \bmod P_i)$.

Therefore, we can reform Eq. (10) as:

$$f_1 * f_2 = \sum_{n \in \mathbb{Z}_N} f_1(n) \cdot (g^n \cdot f_2).$$

Next we prove that $g^k \cdot (f_1 * f_2) = (g^k \cdot f_1) * f_2$. On one hand,

$$(g^k \cdot (f_1 * f_2))(t) = (f_1 * f_2)(t-k) = \sum_{n \in \mathbb{Z}_N} f_1(n) \cdot f_2(t-k-n).$$

On the other hand,

$$((g^k \cdot f_1) * f_2)(t) = \sum_{n \in \mathbb{Z}_N} (g^k \cdot f_1)(n) \cdot f_2(t-n) = \sum_{n \in \mathbb{Z}_N} f_1(n-k) \cdot f_2(t-n).$$

Replace $n-k$ with m, then $n = k+m$, and as n ranges over $\mathbb{Z}_N$, so does m. We have

$$((g^k \cdot f_1) * f_2)(t) = \sum_{m \in \mathbb{Z}_N} f_1(m) \cdot f_2(t-k-m).$$

This is the same formula as above, hence we have proved the equation.

Let $P = \mathrm{lcm}(P_1, P_2)$, then $g^P \cdot f_1 = f_1$, and $g^P \cdot f_2 = f_2$. From this we know that the circular convolution $f_1 * f_2$ has a period of P:

$$g^P \cdot (f_1 * f_2) = (g^P \cdot f_1) * f_2 = f_1 * f_2, \quad i.e.\ (f_1 * f_2)(t-P) = (f_1 * f_2)(t).$$

This is exactly the composite periodicity of two independent sequence periods.

[3] Here we use the inverse to ensure the associativity in the induced action. See footnote 1.

D Complete Results for Effect of Training Data Density on OOD Generalization (Experiment 2)

The full category accuracy bar and the loss curves after increasing data density are shown in Fig. 12 and Fig. 13. TFKAN (Kui et al., 2025) employs a dual time–frequency KAN (Liu et al., 2024) architecture that models nonlinear relationships and periodicity in time and frequency domains. In the following appendix, we include TFKAN as a baseline. The experimental results of KAN (or TFKAN) are similar to those reported in prior work (Takehara, 2025).

E Complete Experimental Results for Scaling Model Parameters (Experiment 3)

The full results for different model and different parameter scales are shown in Table 4.

F Complete Experimental Results for the Circular Convolution Task (Experiment 4)

The full accuracy heatmap, the category accuracy bar, and the loss curves for the circular convolution task are shown in Fig. 14, Fig. 15, and Fig. 16, respectively.

G Complete Experimental Results for More Composite Periodicity

In Section 5, our main experiment evaluates models' ability to generalize composite periodicity using the **Coper** dataset introduced in Section 4. We utilize two rules: addition and modulo operation, which are both invariant, *i.e.,* both with period $T = 1$. In this appendix, we further test models on additional composite periodicity to extend the concept of rule periodicity. We replace the addition rule with a periodic operation: applying addition and subtraction alternately. The composite operation can be formalized as

$$
\begin{aligned}
&C'(f_1(t), f_2(t)) = (M \circ R')(f_1(t), f_2(t)) \\
&= \begin{cases} \big(f_1(t \bmod P_1) + f_2(t \bmod P_2)\big) \bmod P, & 2 \mid t \\ \big(f_1(t \bmod P_1) - f_2(t \bmod P_2)\big) \bmod P, & 2 \nmid t \end{cases} \\
&= \big(f_1(t \bmod P_1) + (-1)^t f_2(t \bmod P_2)\big) \bmod P.
\end{aligned}
$$

Consequently, the period of rule R' becomes $T = 2$, and $R'_{t+2} = R'_t$.

Table 4: Effect of model scaling on composite periodicity generalization. Importantly, OOD generalization can only be interpreted when in-distribution fitting is sufficient. With different ID performance and loss, OOD loss and OOD accuracy are not strictly positively correlated.

Model	Layers	Testset Loss ↓		ID	Hollow	Extrapolation	Avg.
		ID	OOD				
Transformer	3	0.34	3.63	94.7	29.8	22.6	49.0
Transformer	5	0.05	3.89	99.8	36.4	24.4	53.5
Transformer	7	0.02	2.88	99.9	57.2	32.7	63.3
FANFormer	3	0.91	3.21	71.0	36.1	21.4	48.8
FANFormer	5	0.47	3.15	89.5	41.7	19.4	50.2
FANFormer	7	0.61	2.09	83.1	47.5	25.2	51.9
TFKAN	3	0.45	3.73	91.5	31.7	20.9	48.0
TFKAN	5	0.17	2.38	98.1	50.5	27.6	58.7
TFKAN	7	0.21	2.14	97.4	52.4	35.4	61.7
Mamba	3	1.36	1.42	49.4	48.4	46.7	48.2
Mamba	5	1.33	1.48	49.5	43.1	45.3	46.0
Mamba	7	1.47	1.63	45.6	42.0	36.0	41.2
RWKV	3	1.08	1.14	60.5	57.9	59.5	59.3
RWKV	5	0.95	1.09	64.4	62.6	60.4	62.5
RWKV	7	0.96	1.03	65.3	65.0	62.1	64.1

For example, let $f_1(t) = (1, 2, 3)$ with period $P_1 = 3$ and $f_2(t) = (1, 2)$ with period $P_2 = 2$. Then, their composition produces

$$
\begin{aligned}
&C'(f_1(t), f_2(t)) \\
&= \big(f_1(t \bmod 3) + (-1)^t f_2(t \bmod 2)\big) \bmod P \\
&= (2, 0, 4, 9, 3, 1, \dots).
\end{aligned}
$$

Fig. 17, Fig. 18, and Fig. 19 show the results of the experiment. Just like what happens in Section 5, the models fail to generalize the periodic patterns, whether in the hollow or in the extrapolation area. This experiment provides an additional verification of the conclusion in Section 3.3.2: when it fails to retain relative position difference, Transformer with RoPE fails to represent such periodicity.

H Further Analysis of Hidden Periodicity

This Appendix evaluates the generalization ability of the Transformer and other periodicity modeling works, such as FANFormer, on hidden periodicity. The experiment trains models to fit $y = \sin(x)$. The training data is divided into in-distribution $[-6\pi, +6\pi]$ for training and OOD beyond this interval.

To enable the Transformer's self-attention mechanism, x is tokenized into a fixed 10-digit sequence (*e.g.,* +3.1415926), including the sign ($+$ or $-$), the decimal point, and digits 0–9. The model output is decoded back to numerical values using the

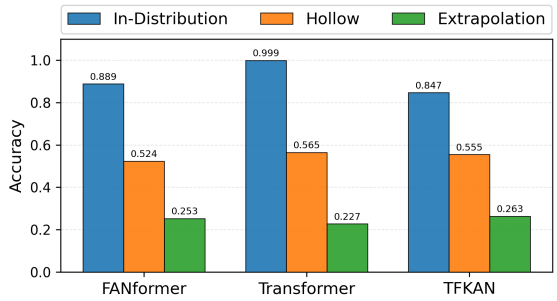

Figure 12: Category accuracy heatmap across models after increasing data density.

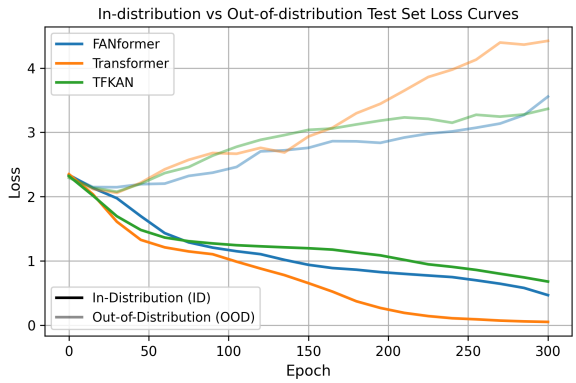

Figure 13: Loss curves on ID and OOD test sets after increasing data density.

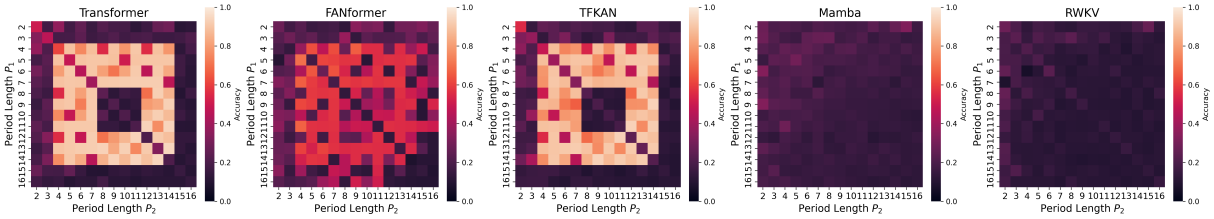

Figure 14: Heatmaps of model performance (450 training epochs) on the circular convolution task across different period lengths.

same digit alignment. The Qwen2.5 tokenizer and embedding are frozen during training.

Beyond sequence and rule periodicity, we further define *hidden periodicity*, where individual samples do not exhibit explicit repetitive patterns, and the periodic structure only emerges at the level of the full data distribution. Formally, let the input be represented as a token sequence $(x_1, x_2, x_3, \dots)$. Hidden periodicity exists if there is a non-zero shift vector $(T_1, T_2, T_3, \dots)$ such that

$$\begin{aligned} &f(x_1, x_2, x_3, \dots) \\ &= f(x_1 + T_1, x_2 + T_2, x_3 + T_3, \dots). \end{aligned}$$

An example is the sine function. Although $\sin(x)$ satisfies $\sin(x) = \sin(x + 2\pi)$, this periodicity cannot be directly observed from a single sample. Instead, after tokenization, adding one period changes multiple digit positions simultaneously due to carry operations. For example,

$$\sin(+3.1415926) = \sin(+6.2831852),$$

which corresponds to a coordinated token-level transformation such as

$$f(x_1, x_2, x_3, \dots) = f(x_1 + 3, x_2 + 1, x_3 + 4, \dots).$$

Therefore, the periodicity can only be inferred after observing many (x, y) pairs across the full data distribution.

As shown in Figure 20, FANFormer also fails to generalize hidden periodicity, similar to the Transformer.

I LLMs Usage

In preparing this manuscript, we use LLMs to aid and polish the writing. Specifically, LLMs improve clarity, grammar, and phrasing, ensuring the text is concise and readable. The use of LLMs **does not** influence the technical contributions or the interpretation of experimental findings. All content polished by LLMs is carefully checked by the authors.

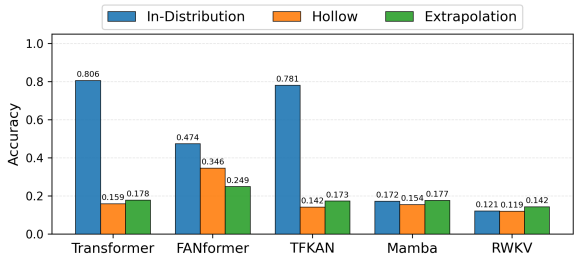

Figure 15: Accuracy by category of different models on the circular convolution task on various OOD settings.

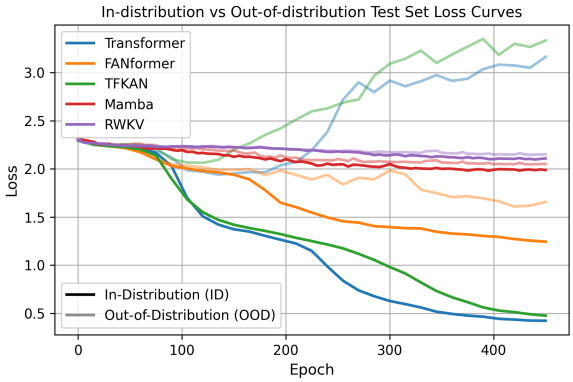

Figure 16: Loss curves on ID and OOD test sets during training on circular convolution tasks.

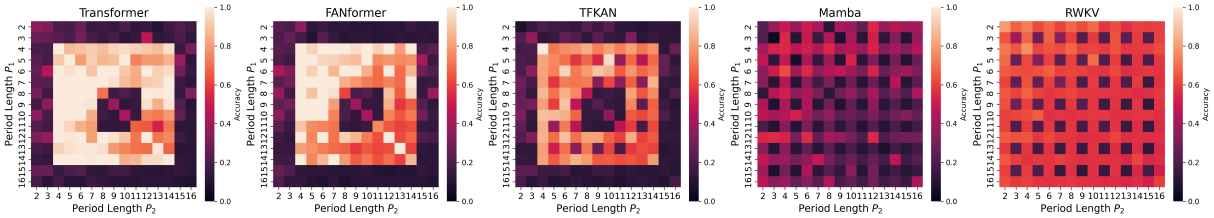

Figure 17: Heatmaps of model performance (450 training epochs) on the add–subtract alternating task across different period lengths.

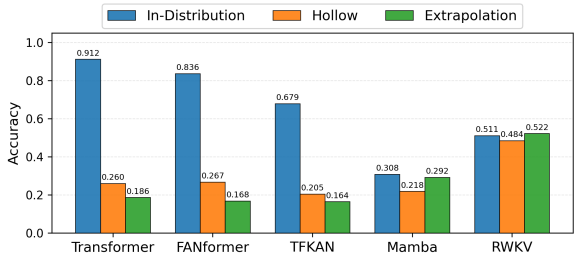

Figure 18: Accuracy by category of different models on the add-subtract alternating task under various OOD settings.

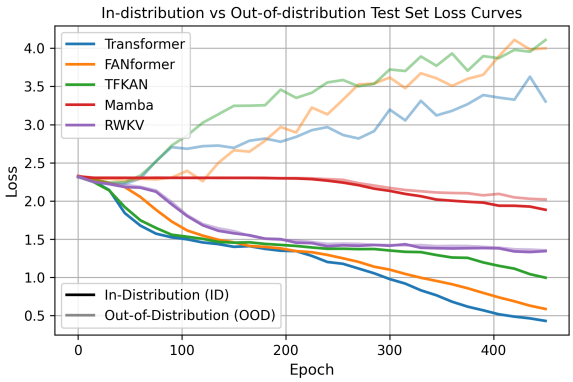

Figure 19: Training loss curves of different models on the add-subtract alternating task.

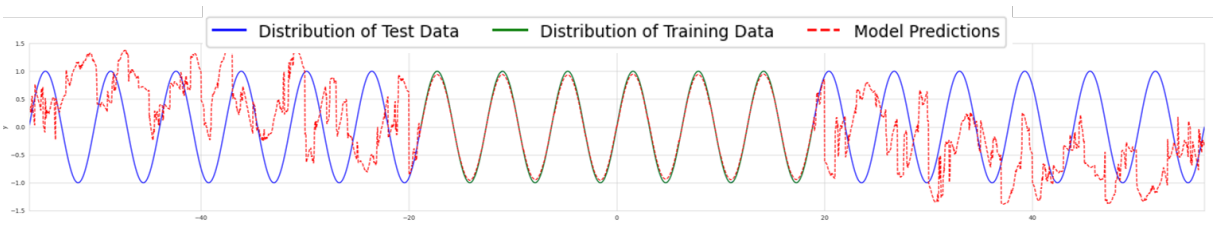

Figure 20: FANformer fails to generalize y = sin(x), where x is a token sequence