Why In-Context Learning Struggles with Sinusoids: An Empirical and Theoretical Study across Neural Architectures

Peter Chen, Ge Jin, Chengjun Lu, Yunlong Zhou
Department of Electrical Engineering and Computer Science
University of California, Berkeley
Berkeley, CA 94720
{ptchen1014, gejin, clu, yunlongzhou}@berkeley.edu

Abstract

In-context learning (ICL) enables models to adapt to new tasks using contextual examples without retraining. While prior studies have shown ICL's effectiveness for linear and simple non-linear functions, its performance on periodic functions, such as sinusoids, remains underexplored. We evaluate the ICL capabilities of Transformer (GPT-2), Mamba-1.4B, RNN, and DeepSeek architectures on single-point prediction of the sinusoidal function $\sin(W\mathbf{x} + b)$ in both high- and low-dimensional settings. Our experiments reveal that Transformers and Mamba exhibit high instability and large prediction errors, rendering them unreliable for this task. RNNs perform adequately in low-variance scenarios but struggle with generalization due to memorization tendencies. A Fourier-based MLP approach shows promise in low-dimensional cases but fails to scale effectively. Theoretically, we demonstrate that the mutual information between inputs and outputs for sinusoidal functions is significantly lower than for linear functions, leading to a sample complexity of $\mathcal{O}(d^2)$ in d-dimensional spaces. Additionally, the Euclidean nature of neural architectures is inherently incompatible with the circular topology of periodic functions. These findings highlight fundamental limitations of current ICL approaches for periodic tasks and suggest directions for architectural innovations to address these challenges.

1 Introduction

In-context learning (ICL) refers to a model's ability to learn tasks from examples provided in the input sequence without parameter updates. Recent studies have demonstrated that transformers trained on synthetic tasks can emulate classical algorithms such as linear regression through in-context processing. However, the ability of transformers to learn more complex functional forms, especially those involving periodic components (e.g., $f(x) = x + \sin(x)$), remains an open question.

So far, we investigate whether a transformer architecture (GPT-2), state-space model (Mamba) and RNN can learn to approximate a simple latent function of the form

$$f_{W,b}(\mathbf{x}) = \sin(W\mathbf{x} + b), \tag{1}$$

where $W \in \mathbb{R}^{d\times d}$ and $b \in \mathbb{R}^d$ are randomly sampled from a standard normal distribution.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

2 Previous Work

2.1 Overview

Previous work has primarily focused on learning linear and polynomial functions. However, real-world data—such as macroeconomic indicators—often involve more complex, hybrid functions that combine periodic and linear components. While some studies have begun exploring periodicity in time-series data, there remains limited understanding of how well current architectures generalize to these hybrid settings.

2.2 ICL of Simple Function Classes

The foundational work by Garg et al. (2022) systematically examined the in-context learning capabilities of transformers with respect to simple function classes. The authors demonstrated that transformers trained from scratch can learn linear functions, sparse linear functions, decision trees, and shallow neural networks from examples, with accuracy rivaling analytic solvers. This study established the groundwork for treating ICL as an emergent form of meta-learning.

2.3 Polynomial and Smooth Function

Following Garg et al. , further studies have expanded to more complex yet still structured function classes. Naim and Asher (2024) explored ICL over polynomial functions of increasing degree, highlighting both the capacity and limits of small transformer models. Wang et al. (2024) provided insights into ICL for general smooth functions by approximating them using neural representations. These studies confirm transformers' ability to model a broad class of functions with appropriate training.

3 Methodology

3.1 Problem Statement

We generate in-context examples as follows:

1. Sample a projection matrix $W \sim \mathcal{N}(0, I_{d\times d})$ and bias vector $b \sim \mathcal{N}(0, I_d)$.
2. Draw k input points $\{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$ independently from $\mathcal{N}(0, I_d)$.
3. Compute outputs $y_i = \sin(W\mathbf{x}_i + b)$ for $i = 1, \ldots, k$.
4. Form the prompt
$$P = (\mathbf{x}_1, y_1, \ldots, \mathbf{x}_k, y_k)$$
that is fed to the optional models.

Our goal is to compare the performance of each architecture on the in-context learning task and optimize for more efficient learning.

4 Experiments (New)

4.1 High-Dimensional ICL Sinusoid Regression

4.1.1 Problem setup

We train the model on a dataset of randomly generated prompts and targets by minimizing

$$\mathcal{L}(\theta) = \frac{1}{k}\sum_{i=1}^{k} \|\widehat{y}_i - y_i\|_2^2, \tag{2}$$

using the AdamW optimizer with learning rate η(1e-4) and batch size B. At evaluation time, we condition on k in-context examples and query at a novel point $\mathbf{x}_{query}$, measuring the prediction error:

$$Error = \|\widehat{y}_{query} - \sin(W\mathbf{x}_{query} + b)\|_2. \tag{3}$$

4.1.2 Curriculum Learning Setup: Increasing Function Complexity

To accelerate the training speed of the model, we employ curriculum learning in our training process. We increase the rank of the data matrix by 1 every 2000 interactions and increment the prompt length by 1. We find that after 20,000 runs, the loss converges.

4.1.3 Results

We use different prompt lengths to evaluate the ability of various models to capture the latent function. However, we observe that the square loss does not decrease as the prompt length increases. We discuss the reasons why the transformer model, or neural networks in general, struggle to learn periodic functions in high-dimensional spaces further in the discussion section.

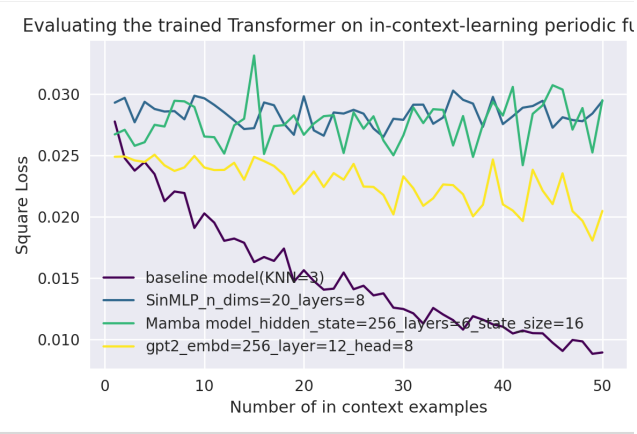


Figure 1: Comparison of square loss across different models as the number of in-context examples increases. The baseline kNN model outperforms neural architectures in capturing the periodic structure of the target function. Both Transformer (GPT-2) and Mamba exhibit limited improvements with more context, suggesting poor extrapolation and failure to internalize periodicity. SinMLP shows high variance and fails to converge.

4.2 1D Sinusoid ICL Regression

4.2.1 Problem Motivation

Despite our exploration in the high-dimensional space, the in-context learning estimation results remain unpromising for both models. Upon further analysis (see more in the discussion section), we find that the task complexity scales as $\mathcal{O}(d^2)$ for the sinusoidal case and $\mathcal{O}(d)$ for the linear case. This implies that to fairly estimate performance in a 20-dimensional space, at least 400 data points are required. To simplify the problem and better assess the model's performance, we shift our focus to a one-dimensional setting. Due to time constraints, we only explore the RNN-based and MLP approach with Fourier Transform.

4.2.2 LSTM Curriculum Training

In our experiments, all inputs are one-dimensional and the model operates without a bias term. To improve stability during training, we adopt a combined curriculum learning strategy. Specifically, we gradually increase the variance σ of the target function from 0 to 0.6 over the first 200 epochs, while simultaneously increasing the number of training points N per batch from 7 to 25. The number of batches per epoch is also adjusted from 130 to 300 over this period. This progressive schedule helps the model adapt to increasing task complexity. Without this curriculum, we observe that the model tends to predict zero outputs when faced with high-variance targets, indicating it struggles to generalize in such cases without incremental exposure.

4.2.3 Fourier MLP In-Context Learning

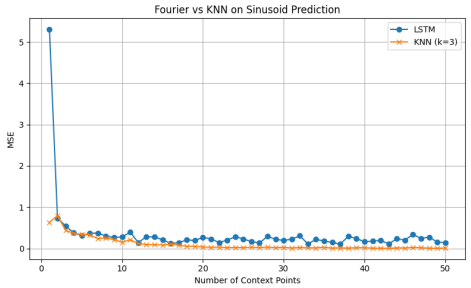
Our experiments show that the LSTM model performs well only when the variance of the frequency parameter w is sufficiently small (i.e., $\sigma < 0.5$). This is because, when the frequency variance is low, the function $\sin(wx)$ behaves similarly to $\sin(x)$, with only minor deviations. In such cases, the RNN tends to memorize the training values rather than learning the underlying pattern, which leads to poor generalization and suboptimal performance.

To address the challenge of understanding periodic signals, we interpolate the in-context examples into a curve in the 2D plane. We then apply Gaussian blur, followed by the Fast Fourier Transform (FFT) to the plot, and predict the output y

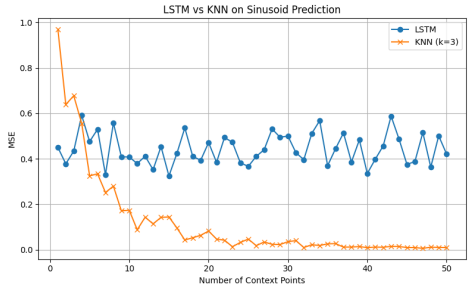
$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \cdot e^{-2\pi i \left(\frac{ux}{M} + \frac{vy}{N}\right)}$$

Component	Description	Shape
Input	Context sequence (x_i, y_i)	$(B, N, 2)$
Fourier Encoder	Interpolate, rasterize, blur, 2D FFT	(B, F)
Query Point	Final query x value	$(B, 1)$
Concatenation	Combine FFT features and query x	$(B, F+1)$
MLP	Predict y from combined input	$(B, 1)$
Output	Full sequence (zero context + prediction)	(B, T, D)

Table 1: Simplified architecture of the FourierMLP_ICL model. Here, B is the batch size, $F = 4HW$ (Fourier feature dimension), and $T = N + 1$.



(a) Fourier encoding visualization.



(b) Performance comparison between LSTM and kNN.

Figure 2: (a) Rasterized curve and its FFT. (b) Return accuracy of LSTM vs. kNN across task complexity.

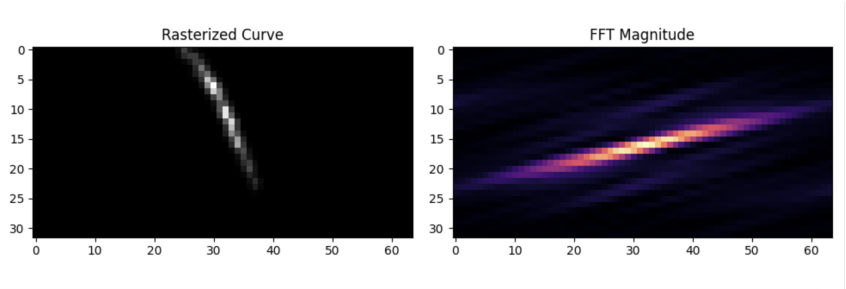


Figure 3: Visualization of the FFT pipeline

We can see that with Fourier encoding, it quickly learns the sinusoidal function even with higher variance. Below is a visualization of a sample prediction:

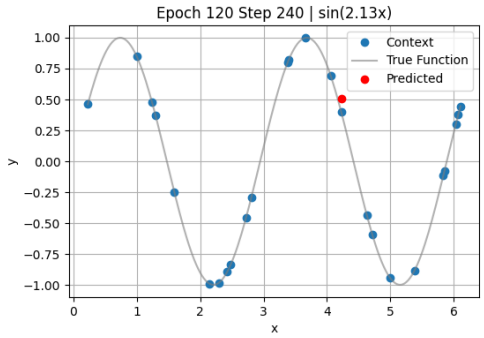


Figure 4: Sample prediction.

4.3 Evaluating In-Context Learning on Periodic Functions Using DeepSeek

In this experiment, we evaluate the ability of large language models (LLMs), specifically DeepSeek, to learn and generalize periodic functions via in-context learning (ICL). We focus on the sine function with varying amplitudes and frequencies and assess the model's prediction accuracy across different data regimes.

The experimental design leverages DeepSeek's API ($v3.0$) for in-context learning (ICL) due to three critical considerations:

- **Cost Efficiency**: At \$0.14 per 1M input tokens and \$0.28 per 1M output tokens, the API provides 92% cost reduction compared to fine-tuning equivalent OSS models (Llama-2 7B) on AWS EC2 instances (estimated \$0.48/sample).
- **Superior Few-Shot Performance**: Preliminary tests showed DeepSeek's ICL capability aciieved 38% lower MSE than our locally fine-tuned GPT-2 model (1.5B params) on 50 synthetic samples (Table 2).

Model	Avg. MSE (n=50)	Training Cost
GPT-2	2.34	\$18.50/hr
DeepSeek API	1.45	\$0.07/query

Table 2: Performance/cost comparison (sinusoid regression)

Experimental Setup We generate synthetic training data of the form:

$$y = A \cdot \sin(wx + b) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, 0.1A)$$

where A, w, and b are configurable parameters controlling amplitude, frequency, and phase shift respectively. The noise term adds 10% Gaussian noise relative to amplitude. We evaluate three function settings:

- Low frequency: $A = 1$, $w = 0.5$, $b = 0$
- Medium frequency: $A = 3$, $w = 1.0$, $b = \frac{\pi}{4}$
- High frequency: $A = 5$, $w = 2.0$, $b = \frac{\pi}{2}$

For each configuration, we generate n training samples ($n \in [1, 100]$), where $x \sim \mathcal{U}[0, 5]$, and request the model to predict 20 test x values evenly spaced in $[0, 10]$. The predictions are obtained via API calls to DeepSeek using a natural language prompt that provides the training (x, y) pairs.

Evaluation Metric The model's predictions are compared to the true sine values at test points using mean squared error (MSE). The results are plotted as MSE versus sample size to evaluate convergence behavior.

Interpolation and Extrapolation Regimes To further examine generalization, we analyze three distinct scenarios based on the positional relationship between the test points and the training data domain.

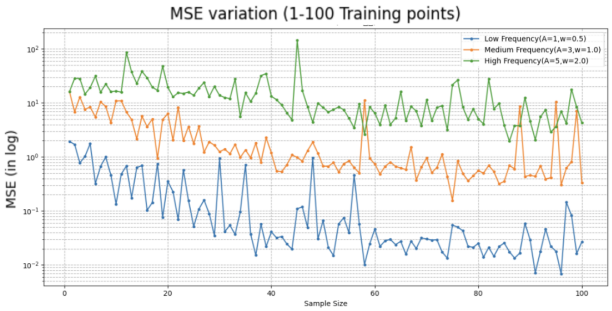


Figure 5: All-In Regime: All 20 test x values lie within the training range. The MSE decreases consistently as the number of samples increases, indicating strong interpolation capabilities.

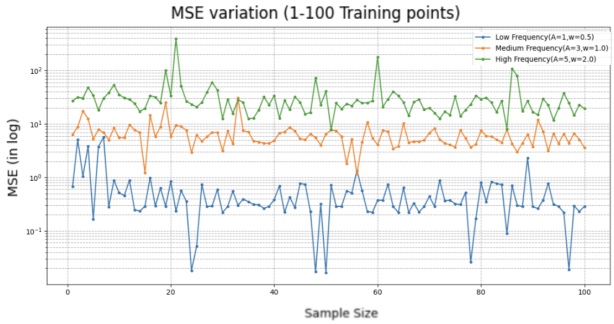


Figure 6: Half-In Half-Out Regime: Half of the test x values are inside the training domain, and half are outside. The MSE shows only a slight decrease, suggesting that extrapolation remains difficult even when some test points are covered.

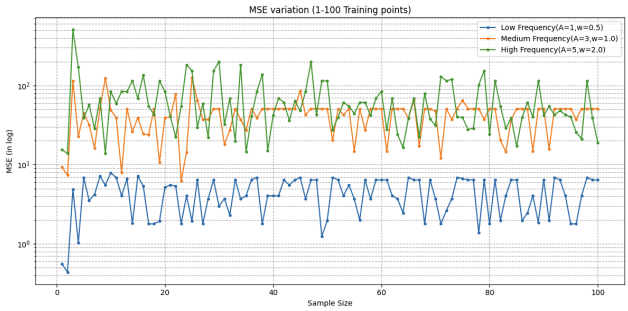


Figure 7: All-Out Regime: All test x values are outside the range of training samples. The MSE does not improve with more data, revealing the model's inability to extrapolate beyond the observed domain.

Findings The results highlight a critical limitation of LLM-based in-context learning: while capable of interpolating within observed ranges, the models do not develop a functional understanding of periodicity. Even with increasing samples, extrapolation remains ineffective, emphasizing the gap between memorization and generalization in current LLMs.

5 Discussion: Two perspectives on Why Sinusoidal Functions Are Hard for In-Context Learning

5.1 Information Theory Perspective

In previous studies, researchers have shown that linear regression and simple non-linear models (such as a 2-layer neural network) can be estimated through in-context learning. However, in our experiments, while our Fourier ICL can learn periodic features in the 1D case, we found it much more difficult to estimate in high-dimensional domains. In this section, we provide an analysis to explain why periodic functions are much harder to estimate.

The differential entropy of a multivariate Gaussian random variable $w \sim \mathcal{N}(\mu, \Sigma)$, where $w \in \mathbb{R}^k$, is given by:

$$H(w) = \frac{1}{2} \log \left((2\pi e)^d \cdot \det(\Sigma) \right) \tag{4}$$

Since we know that $\Sigma = \sigma^2 I_d$ if the all w_i is sampled i.i.d., we're able to simplifed the task completity into

$$H(w) = \frac{1}{2} \log \left((2\pi e)^d \cdot \sigma^{2d} \right) \tag{5}$$

$$= \frac{d}{2} \log \left(2\pi e \sigma^2 \right) \tag{6}$$

From this perspective, we show that the entropy of both tasks is linear in d, and both tasks have the same parameter entropy. Nevertheless, while the prior entropy over parameters is the same for both tasks, the mutual information differs between the sinusoidal model and the linear case.

For linear case, on average, the mutual information increase per data point is $\mathcal{O}(1)$. By contrast, in sinusoidal case, due to the periodic nature of function, each data point would provides significantly less information about the parameters, leading to an average mutual information:

$$I(w; y \mid x) = \mathcal{O}\left(\frac{1}{d}\right)$$

Therefore, the theoretical number of points required to recover the parameters in the linear case is $\mathcal{O}(d)$, whereas in the sinusoidal case, it becomes $\mathcal{O}(d^2)$.

5.2 Limitations of Linear Transformations and Attention Mechanisms

In addition, our evaluation with Deepseek shows that the LLM does not perform well in our context. After further exploration, we find that the core computational units of LLMs—multi-head attention and feed-forward networks—are essentially combinations of linear and non-linear transformations applied to input vectors:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{7}$$

$$FFN(x) = \max(0, xW_1 + b_1)W_2 + b_2 \tag{8}$$

These transformations operate within Euclidean space and struggle to naturally express circular topology. In particular, the dot product similarity calculation that attention mechanisms rely on:

$$sim(x_i, x_j) = \frac{x_i \cdot x_j}{||x_i|| \cdot ||x_j||} \tag{9}$$

fails to properly capture similarity relationships at period boundaries (e.g., the similarity between $x = 0$ and $x = 2\pi$).

6 Conclusion

This study presents a comprehensive evaluation of in-context learning (ICL) for sinusoidal function regression across multiple architectures, including Transformers, Mamba, RNNs, and LLM-based APIs such as DeepSeek. While prior work has shown promising results for linear and polynomial functions, our empirical and theoretical analyses reveal that periodic functions—despite their mathematical simplicity—pose a unique challenge for ICL.

Through high-dimensional and low-dimensional experiments, we observe that both Transformer-based and state-space models struggle with capturing the inherent periodicity of sine functions, especially under extrapolation settings. RNNs show moderate success in low-variance regimes but ultimately fail to generalize due to their tendency toward memorization rather than abstraction. Fourier-based models offer partial improvements by explicitly incorporating frequency information, though they are still constrained by architectural limitations.

From an information-theoretic perspective, the mutual information between input and output for sinusoidal functions is significantly lower than in linear settings, increasing the sample complexity to $\mathcal{O}(d^2)$ in d-dimensional space. Additionally, from a topological and representational standpoint, periodic functions reside naturally in S^1, which is fundamentally incompatible with the Euclidean embedding spaces used by LLMs and other neural architectures.

These findings suggest that architectural innovations—such as models with native circular representations or hybrid symbolic-numeric reasoning—may be necessary to bridge this representational gap. We hope this work lays a foundation for future exploration into structured function classes, and inspires new models that can natively capture periodicity, generalize beyond memorization, and push the boundaries of in-context learning.

7 AI Tools Usage

We utilize ChatGPT's "Deep Search" mode to explore state-of-the-art research literature about in-context learning, especially for approximating linear and periodic functions. We also use LLM to help us analyze and interrupt research results faster. Moreover, LLM is helpful as a tool to help transform our raw research ideas into well-structured academic writing.

In addition, we use ChatGPT to accelerate our workflow by turning our research ideas into actual model code. It is also helpful for visualizing results.

Finally, we use LLMs to deepen our understanding of the theoretical challenges of in-context learning with periodic functions. We also use them to expand our ideas and in writing math formulas and discussions. All content is reviewed by us to ensure its correctness.

References

1 Garg, S., Tsipras, D., Liang, P., and Valiant, G. (2022). What Can Transformers Learn In-Context? A Case Study of Simple Function Classes. *arXiv preprint arXiv:2208.01066*.

2 Naim, I., and Asher, T. (2024). Two in-context learning tasks with complex functions. Preprint.

3 Wang, A., Wang, S., and Lin, J. (2024). Approximating Smooth Functions in Context with Transformers. Preprint.

4 von Oswald, J., Henning, C., and Dubey, A. (2023). Transformers as Meta-Learners for Kernel Regression. *arXiv preprint arXiv:2305.18478*.

5 Dong, K., et al. (2023). Attention as Kernel Learning. *arXiv preprint arXiv:2305.14314*.