When can isotropy help adapt LLMs' next word prediction to numerical domains?

Anonymous authors
Paper under double-blind review

Abstract

Vector representations of contextual embeddings learned by pre-trained large language models (LLMs) are effective in various downstream tasks in *numerical domains* such as time series forecasting. Despite their significant benefits, the tendency of LLMs to hallucinate in such domains can have severe consequences in applications such as energy, nature, finance, healthcare, retail and transportation, among others. To guarantee prediction reliability and accuracy in numerical domains, it is necessary to open the black box behind the LLM and provide performance guarantees through explanation. However, there is little theoretical understanding of when pre-trained language models help solve numerical downstream tasks. This paper seeks to bridge this gap by understanding when the next-word prediction capability of LLMs can be adapted to numerical domains through a novel analysis based on the concept of isotropy in the contextual embedding space. Specifically, a log-linear model for LLMs is considered in which numerical data can be predicted from its context through a network with softmax in the output layer of LLMs (i.e., language model head in self-attention). For this model, it is demonstrated that, in order to achieve state-of-the-art performance in numerical domains, the hidden representations of the LLM embeddings must possess a structure that accounts for the shift-invariance of the softmax function. By formulating a gradient structure of self-attention in pre-trained models, it is shown how the isotropic property of LLM embeddings in contextual embedding space preserves the underlying structure of representations, thereby resolving the shift-invariance problem and providing a performance guarantee. Experiments across 22 different numerical datasets and 5 different language models show that different characteristics of numerical data and model architectures could have different impacts on the isotropy measures, and this variability directly affects the time series forecasting performances.

1 Introduction

Large language models (LLMs) have been proven to effective in adapting to various downstream tasks in numerical domains, such as finance Garza and Mergenthaler-Canseco (2023); Yu et al. (2023), energy Gao et al. (2024), climate science Jin et al. (2024), healthcare Wang and Zhang (2024), transportation signals Xu et al. (2024), synthetic tabular generation Dinh et al. (2022); Borisov et al. (2023); Xu et al. (2024), among others. Inspired by the success of pre-trained LLMs, several methods have been developed recently in Gruver et al. (2024); Dooley et al. (2023); Nie et al. (2023); Rasul et al. (2024); Woo et al. (2024); Jin et al. (2024); Ansari et al. (2024) by adapting LLM to numerical domains that deal with time series forecasting. For many of these numerical downstream tasks, training a linear classifier on top of the hidden-layer representations generated by the pre-trained LLMs has been shown to achieve near state-of-the-art performance Jin et al. (2024); Ansari et al. (2024). However, the existing models in Gruver et al. (2024); Dooley et al. (2023); Nie et al. (2023); Rasul et al. (2024); Woo et al. (2024); Jin et al. (2024); Ansari et al. (2024) are treated as a 'black box' where numerical forecasts are controlled by complex nonlinear interactions between many parameters. This makes it difficult to understand how models arrive at their predictions and makes it challenging for users to trust the model outputs.

When applied to critical numerical domain use cases, the tendency of LLMs to hallucinate can have serious and detrimental consequences. For example, prediction errors in fraud detection in finance can lead to

huge financial losses and errors in protection onset of sepsis or cardiac arrest in healthcare can result in patient deaths. Thus, to guarantee prediction reliability and accuracy in numerical domains, it is necessary to understand the inner working of the so-called black box and provide performance guarantees through explanation. Although recent empirical studies Jin et al. (2024); Nie et al. (2023); Liu et al. (2024) demonstrate the benefits of vector representations of embedding learned by LLMs in various numerical downstream tasks, there is little theoretical understanding of their empirical success. Thus, a fundamental question arises: *"When (or how) can the next-word prediction capability of LLMs be effectively adapted to numerical domains?"*

The main contribution of this paper is a novel approach for answering this question by exploiting the isotropic property of LLM hidden representations in the contextual embedding space. *Isotropy* refers to the geometric property wherein vector representations in the embedding space are uniformly distributed in all directions, a characteristic critical for maintaining the expressiveness of the embedding space Arora et al. (2016); Mu and Viswanath (2018). To achieve state-of-the-art performance in numerical domains, we show that the hidden representations of LLMs must exhibit *a structured form* in contextual embedding space that accounts for the shift-invariance of the softmax function (i.e., the softmax output remains unchanged when all logits are shifted by a constant). Without such structure, the model can shift the logits while keeping the training loss unchanged, thereby leaving the logits ineffective for numerical downstream tasks. By formulating a gradient structure of self-attention in pre-trained models, we show how the isotropic property of LLM embeddings in the contextual embedding space preserves the underlying structure of representations, thereby resolving the shift-invariance problem of the softmax function. In a nutshell, our key contributions include:

- We consider a log-linear model for LLMs and demonstrate theoretically why hidden representations must exhibit structure to address the shift-invariance problem of the softmax function.
- We take a deeper look into the hidden representations of pre-trained models and show how isotropy preserves the structural integrity of representations. In particular, we derive an upper bound for the Jacobian matrix which collects all first-order partial derivatives of self-attention with respect to the input pattern and show that the m largest eigenvectors of the LLM hidden representations minimize the gradient norm of self-attention. Then, by projecting the representations into lower dimensions using these m largest eigenvectors, we find the isotropy within the clusters in the contextual embedding space.
- Finally, we provide a comprehensive evaluation across 12 real and 10 synthetic time series datasets over 5 different LLMs. Our experiments demonstrate that the isotropy of LLM hidden representations varies significantly based on the input data characteristics (i.e., domain, context length and noise level) and model design choices (i.e., tokenization techniques and architecture), which in turn strongly influences forecasting performance in numerical domains.

2 Problem Setup in Numerical Domains

Time Series Tokens and Similarity Measure. Similar to next-word prediction by LLMs, the next-value prediction in the numerical domain can be modeled by *time series forecasting* techniques which are widely adopted in the machine learning literature Jin et al. (2024); Ansari et al. (2024). Formally, given a time series $\mathbf{x}_{1:T+L} = [x_1, \ldots, x_T, \ldots, x_{T+L}]$, where the first T time instances give the historical context, the next L time instances constitute the forecast region, and $x_t \in \mathbb{R}$ is the observation of each time instance, we are interested in predicting the joint distribution of the next L time instances, $p(\mathbf{x}_{T+1:T+L}|\mathbf{x}_{1:T})$. Since, the pre-trained models operate on tokens from a finite vocabulary, using them for time series data requires mapping the observations to a finite set of tokens. Based on different numerical applications and LLM architectures, various tokenization techniques, e.g., quantization and scaling Ansari et al. (2024); Rasul et al. (2024), patching Woo et al. (2024); Jin et al. (2024); Nie et al. (2023), and adaptation of language model tokenizer

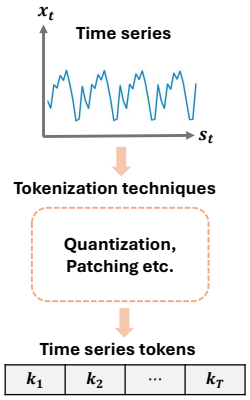


Figure 1: Time series tokenization.

in numerical domians Gruver et al. (2024); Dooley et al. (2023), can be applied to tokenize the time series and create a time series vocabulary $\mathcal{V}$ of N time series tokens, i.e., $|\mathcal{V}| = N$, as shown in Figure 1. Then, the realization of the next L time instances can be obtained by autoregressively sampling from the predicted distribution $p(k_{T+l+1} \,|\, \mathbf{k}_{1:T+l})$, for $l \in \{1, \ldots, L\}$, where $\mathbf{k}_{1:T+l}$ is the tokenized time series and k_i be a time series token in time series vocabulary $|\mathcal{V}|$.

Let $\tilde{\mathbf{\Psi}}(k_i) = \{\psi_1(k_i), \psi_2(k_i), \ldots\}$ be the set of all LLM contextual embedding instances of time series token k_i. Here, different contexts in the time series sequences yield different LLM embeddings of k_i. By constructing $\sum_k |\tilde{\mathbf{\Psi}}(k)| = |\mathcal{V}|$, we define the inter-token cosine similarity as:

$$\zeta_{\cos} \triangleq \mathbb{E}_{i \neq j}[\cos(\psi(k_i), \psi(k_j))], \tag{1}$$

where $\psi(k_i)$ and $\psi(k_j)$ are random samples from $\tilde{\mathbf{\Psi}}(k_i)$. The expectation is taken over all pairs of different tokens. The inter-token cosine similarity metric describes the similarity between different tokens based on the contexts. For notational simplicity, we express $T + l$ as T_l and $T + l + 1$ as T_{l+1} hereinafter.

Model. We consider a general pre-trained model for numerical data and open the black box of the pre-trained model. Specifically, we assume that the observation probability of $k_{T_{l+1}}$ given $\mathbf{k}_{1:T_l}$ satisfies the log-linear model as in Arora et al. (2016)

$$p^*(k_{T_{l+1}} = i \mid \mathbf{k}_{1:T_l}) \propto \exp(\langle \psi^*(\mathbf{k}_{1:T_l}), \psi^*(k_i) \rangle), \tag{2}$$

where $\psi^*(k_i) \in \mathbb{R}^D$ is a vector that only depends on the time series token $k_{i \in \mathcal{V}}$, and $\psi^*(\mathbf{k}_{1:T_l})$ is a function that encodes the tokenized time series sequence $\mathbf{k}_{1:T_l}$ into a vector in $\mathbb{R}^D$. The log-linear modeling aligns with the commonly used LLMs networks whose last layer is typically a softmax layer. Moreover, we do not consider any prior distribution for input, which makes our model more general than previous latent models Arora et al. (2016); Wei et al. (2021).

To define the numerical downstream task, let $z_i^*(k, l) := \langle \psi^*(\mathbf{k}_{1:T_l}), \psi^*(k_i) \rangle$ be the i-th logit of the ground-truth model, and assume that the numerical downstream tasks are defined by a function of the logits, i.e., $f^*(\mathbf{z}^*)$. Also let $Z^*(k, l) = \sum_{i=1}^N \exp(z_i^*(k, l)) = \sum_{i=1}^{|\mathcal{V}|} \exp(\langle \psi^*(\mathbf{k}_{1:T_l}), \psi^*(k_i) \rangle)$ be the partition function Arora et al. (2016), i.e., normalization factor. In LLMs, the partition function is often used to normalize the output probabilities of the model, ensuring that they sum to 1. Then, the normalized ground-truth model $\forall i \in \mathcal{V}$ is given by

$$p(k_{T_{l+1}} = i \mid \mathbf{k}_{1:T_l}) = \frac{\exp(\langle \psi^*(\mathbf{k}_{1:T_l}), \psi^*(k_i) \rangle)}{\sum_{i=1}^{|\mathcal{V}|} \exp(\langle \psi^*(\mathbf{k}_{1:T_l}), \psi^*(k_i) \rangle)} = \frac{\exp(z_i^*(k, l))}{Z^*(k, l)}.$$

Since we do not know the ground-truth model in reality, we do not have access to the ground-truth model components $\psi^*(k_i)$ and $\psi^*(\mathbf{k}_{1:T_l})$. Instead, we only have access to the student model $\psi(k_i)$ and $\psi(\mathbf{k}_{1:T_l})$ that aims to achieve low pre-training loss. We can define the student logits as $\mathbf{z}(k, l) := \{\langle \psi(\mathbf{k}_{1:T_l}), \psi(k_i) \rangle\}_{i=1}^{|\mathcal{V}|}$. Intuitively, $\mathbf{z}$ are the contextualized representations learned by the student-model during pre-training. Then, the solution of the downstream task is to learn a function $f(k, l)$. Then, the output of the student model $\forall i \in \mathcal{V}$ can be defines as

$$p\left(k_{T_{l+1}} = i \mid \mathbf{k}_{1:T_l}\right) = \frac{\exp(\langle \psi(\mathbf{k}_{1:T_l}), \psi(k_i) \rangle)}{Z(k, l)}. \tag{3}$$

Loss Function. As typical in language models, we use the categorical distribution over the elements in the time series vocabulary $\mathcal{V}$ as the output distribution $p(k_{T_{l+1}} | \mathbf{k}_{1:T_l})$, for $l \in \{1, \ldots, L\}$, where $\mathbf{k}_{1:T_l}$ is the tokenized time series. The student model is trained to minimize the cross entropy between the distribution of the tokenized ground truth label and the predicted distribution. The loss function for a single sequence of tokenized time series is given by Ansari et al. (2024); Wu et al. (2023)

$$\begin{aligned} \mathcal{L} &= -\sum_{l=1}^{L+1} \sum_{i=1}^{|\mathcal{V}|} p^*(k_{T_{l+1}} = i \,|\, \mathbf{k}_{1:T_l}) \log p(k_{T_{l+1}} = j \,|\, \mathbf{k}_{1:T_l}) \\ &= \sum_{l=1}^{L+1} \sum_{i=1}^{|\mathcal{V}|} \mathcal{D}_{\mathrm{KL}}(p^*(k_{T_{l+1}} = i | \mathbf{k}_{1:T_l}) \| \, p(k_{T_{l+1}} = j | \mathbf{k}_{1:T_l})) + H(p^*(k_{T_{l+1}} = j | \mathbf{k}_{1:T_l})), \end{aligned} \tag{4}$$

where $p(k_{T_{l+1}}{=}i \,|\, \mathbf{k}_{1:T_l})$ is the categorical distribution predicted by the student model parametrized by $v_{1:T_l}$, $p^*(k_{T_{l+1}}{=}\, i|\mathbf{k}_{1:T_l})$ is the distribution of ground-truth model, $\mathcal{D}_{\mathrm{KL}}$ is the KL divergence, and $H(p^*(k_{T_{l+1}}{=}\, i|\, \mathbf{k}_{1:T_l}))$ is the entropy of distribution $p^*(k_{T_{l+1}}{=}i|\mathbf{k}_{1:T_l})$ which is a constant. We assume that student model achieves a small loss so that the KL-divergence term in equation 4 is also small.

Downstream Numerical Task. We consider a simple downstream task whose prediction on categorical distribution is linear in $\psi^*(\mathbf{k}_{1:T_l})$, that is, $f^*(k,l) = \langle \psi^*(\mathbf{k}_{1:T_l}), u^* \rangle = \sum_{i=1}^{|\mathcal{V}|} a_i^* z_i^*(k,l)$, where $u^* = \sum_{i=1}^{|\mathcal{V}|} a_i^* \psi^*(k_i) \in \mathbb{R}^D$ and a_j is the coefficient. This model is still not sufficient to provide a performance guarantee to generalize to downstream task in unseen scenarios. However, the log probability difference is proportional to the difference in the value of the perfect model (i.e., ground-truth) $f^*(k,l)$. This allows the student model to alter the signs of $f^*(k,l)$ without resulting in a large KL divergence Wu et al. (2023). Then, it is more reasonable to model the numerical downstream task as

$$f^*(k,l) = \sum_{i=1}^{|\mathcal{V}|} a_i^* \sigma(z_i^*(k,l) - b_i^*) = \sum_{i=1}^{|\mathcal{V}|} a_i^* \sigma(\langle \psi^*(\mathbf{k}_{1:T_l}), \psi^*(k_i) \rangle - b_i^*),$$

where σ is the ReLU function and b_j^* denotes the threshold for the logits. The numerical downstream task only considers the logits that are above the threshold, and thus ignores all the entries with very small probabilities.

Despite the empirical success of language models in numerical domains, there remains a fundamental gap in understanding when and why these models generalize reliably to numeric downstream tasks such as time series forecasting across different numerical domains. A key challenge lies in the mismatch between training and inference behavior, i.e., good training performance does not always guarantee robust performance at inference time. To address this challenge, we propose a novel theoretical framework grounded in the isotropic property of the contextual embedding space. We show that the presence of strong isotropy in LLM hidden representations stabilizes the partition function, effectively resolving the softmax shift-invariance problem and leading to reliable inference performance. The next section formalizes this insight and provides theoretical justification for using isotropy as a key indicator of model reliability in numerical settings.

3 The Role of Isotropy in Adapting LLMs to Numerical Data

As discussed in Section 2, we consider LLM networks whose last layer is usually a softmax layer and the numerical downstream task is determined by the function of the logits. The underlying relation between the logits and softmax function determines the performance of the numerical downstream tasks. However, the softmax function is shift-invariant, that is, the output of the softmax function remains unchanged when all logits are shifted by a constant. Since we do not have any control over the logit shift of the student model on unseen data, good performance during training does not necessarily provide any performance guarantee for the numerical downstream task on unseen scenarios. The lack of perfromance guarantee under uncontrolled logit shifts on unseed data can be formalized in the following theorem:

Theorem 1. *Let the logits of the ground-truth model be bounded. Then for any $f^*(k,l)$, there exists a set of functions $\{\hat{z}_i(k,l)\}_{i=1}^{|\mathcal{V}|}$ such that for all k and T_{l+1}, the predictive distribution of the student model $\hat{p}\,(k_{T_{l+1}} \,|\, \mathbf{k}_{1:T_l})$ matches that of ground-truth model $p^*(k_{T_{l+1}} \,|\, \mathbf{k}_{1:T_l})$ and $\hat{f}(k,l) = 0$. In other words, there exists a student model with the same pre-training loss as the ground-truth model, but its logits are ineffective for the numerical downstream tasks.*

Proof. The proof is provided in Appendix A. □

Theorem 1 shows that without any structure in the hidden representations of LLM embeddings, student model can shift the logits for any sample while keeping the pre-training loss unchanged and leaving logits ineffective for the numerical downstream tasks. Consequently, a theoretical guarantee for numerical downstream task performance will require structure in the LLM representations learned by the pre-trained model.

In this paper, we make an observation that to prevent the shift-invariance problem from influencing the performance of the numerical downstream tasks, it is necessary to keep the partition function stable. Let $\mathbf{\Psi} = (\psi_1(k), \ldots, \psi_{|\mathcal{V}|}(k))^\top \in \mathbb{R}^{|\mathcal{V}| \times D}$ be the hidden representations of input time series sequence. Then, the stability of the partition function can be assessed through the isotropy in the contextual embedding space Arora et al. (2016); Mu and Viswanath (2018) as follows

$$I(\{\psi(k)\}) = \frac{\min_{\psi(\mathbf{k})\in\mathcal{C}} Z(k,l)}{\max_{\psi(\mathbf{k})\in\mathcal{C}} Z(k,l)}, \tag{5}$$

where $\mathcal{C} = \mathbf{\Psi}^\top \mathbf{\Psi}$ is the input correlation matrix of input pattern and $l = 1, \ldots, L$. From equation 5, we can see that when the partition function is constant (i.e., stable) for different samples, $I(\{\psi(k)\})$ becomes close to 1 which indicates that the contextual embedding space $\{\psi(k)\}$ is more isotropic Arora et al. (2016); Mu and Viswanath (2018). Note that in equation 3, the probability of a value in any time instance is the exponential of the corresponding logit $z_i(k,l)$ divided by the partition function $Z(k,l)$. If the partition function remains stable for different samples, the logits can be solely determined by the probabilities, thereby resolving the shift-invariance problem of the softmax function.

Building on this theoretical foundation, we now turn to the following empirical question: *"How can we measure and interpret isotropy in practice, and how does it relate to generalization across numerical domains?"*. Motivated by Theorem 3.1 and the need for structural constraints in LLM representations, we analyze the effective dimensionality and cluster organization of LLM's hidden representations in the contextual embedding space. These analyses reveal how isotropy manifests in pre-trained LLMs and how its presence correlates with the model's ability to generalize to time series forecasting across different numerical domains, and hence, provides a performance guarantee. Section 4 introduces methods for quantifying this internal structural integrity using spectral alignment mechanism, principal component analysis (PCA) and cluster-based isotropy metric, and consequently, linking theoretical reliability (i.e., performance guarantee) to empirical generalizability.

4 Study of isotropy in LLM hidden representations

Analysis Settings. For this study, we consider five different language models including Chronos-T5 Ansari et al. (2024), Chronos-Bolt[1], PatchTST Nie et al. (2023), Moirai-1.0-R Woo et al. (2024), and Lag-Llama Rasul et al. (2024). For illustration, we randomly select a real dataset (i.e., finance-Dataset 1) from a broader collection of 22 numerical datasets that we use in this paper since we see similar results with all of these datasets. The details of these models and datasets could be found in Section 5.

4.1 Effective Dimensions

Table 1: The effective dimension $d(0.8)$

Layer	1	2	3	4	5	6	7	8	9	10	11	12
Chronos-T5	4	4	4	4	4	4	4	4	4	4	4	4
Chronos-Bolt	1	1	1	1	1	1	1	1	1	1	1	1
PatchTST	1	1										
Moirai	1	1	1	1	1	1						
Lag-Llma	2	2	2	2	2	2	2	2				

In each layer of each model, we start with a data matrix $\mathbf{A} \in \mathbb{R}^{|\mathcal{V}|\times D}$, where $|\mathcal{V}|$ represents the number of tokens in the input time series sequence, and D corresponds to the embedding dimension. We apply PCA to reduce the dimensionality from D to m i.e., $\tilde{\mathbf{A}} \in \mathbb{R}^{|\mathcal{V}|\times m}$. Then, the fraction of variance captured by the reduced representation is given by: $r_m = \frac{\sum_{i=0}^{m-1}\sigma_i}{\sum_{i=0}^{D-1}\sigma_i}$ where σ_i denotes the i-th largest eigenvalue of the covariance matrix of $\mathbf{A}$. We define the ϵ-effective dimension as $d(\epsilon) \triangleq \arg\min_m r_m \geq \epsilon$. For instance, if $d(0.8) = 3$, then three principal dimensions retain 80% of the variance. A higher d suggests a more isotropic space Cai et al. (2021), where information is spread across multiple dimensions rather than being concentrated in a narrow subspace. Table 1 presents the values of $d(0.8)$ for different layers and models. Surprisingly, all of these models have very small effective dimensions as compared to original embedding dimensions. For instance, Chronos-Bolt has very small effective dimensions, with $d(0.8) = 1$ for layers 1 through 12, as compared to its original embedding dimensions $D = 512$. The small effective dimensionality is another way of telling that that Chronos-Bolts's embedding vectors lie in a subspace defined by a very narrow cone Ethayarajh (2019), and consequently, their inter-token cosine similarity is large. If all the embedding vectors lie on a 1-dimensional line, the inter-token cosine similarity would be close to 1, and there would be hardly any model capacity. Surprisingly, despite having such low effective dimensionality, these language models still perform well in numerical domains. This counterintuitive result motivate us to look deeper into the contextual embedding space.

[1] `https://huggingface.co/autogluon/chronos-bolt-base`

4.2 Spectral Alignment for Generalization in Numerical Settings

Let $G(\mathbf{\Psi}) = (g_1(\mathbf{\Psi}), \ldots, g_{|\mathcal{V}|}(\mathbf{\Psi}))^\top : \mathbb{R}^{|\mathcal{V}|\times D} \mapsto \mathbb{R}^{|\mathcal{V}|\times D}$ be the function for self-attention, i.e., $g_i(\mathbf{\Psi}) = \text{softmax}(\mathbf{\Psi}\mathbf{\Lambda}\mathbf{\Psi}^\top)\mathbf{\Psi}$, where $\mathbf{\Lambda} = \mathbf{W}_Q\mathbf{W}_K^\top \in \mathbb{R}^{D\times D}$, and $\mathbf{W}_Q \in \mathbb{R}^{D\times m}$, $\mathbf{W}_K \in \mathbb{R}^{D\times m}$ are the parameter matrices for the query and key matrices of self-attention. The lemma below provides insights into how the isotropic property of pre-trained LLMs enables generalization in numerical domains. The proof of this lemma follows the analysis in Kim et al. (2021) is provided in Appendix B for completeness.

Lemma 1. *Consider the Jacobian matrix* $\mathbf{J} = \left[\frac{\partial g_i(\mathbf{\Psi})}{\partial \psi_j}\right]_{i,j=1}^{|\mathcal{V}|}$, *which represents the gradient of the self-attention mapping* $G(\mathbf{\Psi})$ *with respect to the input time series token embeddings. Then the spectral norm of* $\mathbf{J}$ *satisfies* $\|\mathbf{J}\|_2 \leq |\mathbf{\Lambda}|_2 \sum_{i=1}^{|\mathcal{V}|} \left(p_{i,i} + \frac{1}{2}\right) \left|\psi_i - \sum_{j=1}^{|\mathcal{V}|} p_{i,j}\psi_j\right|^2 + \Delta$, *where the residual term* Δ *is given by* $\Delta = |\mathbf{\Lambda}|_2 \sum_{i\neq j}^{|\mathcal{V}|} p_{i,j} \left|\psi_j - \sum_{q=1}^{|\mathcal{V}|} p_{i,q}\psi_q\right|^2 + \frac{|\mathbf{\Lambda}|_2}{2} \sum_{j=1}^{|\mathcal{V}|} |\psi_j|^2$, *and the attention weights* $p_{i,j}$ *are defined as* $p_{i,j} = \frac{\exp(\psi_i^\top \mathbf{\Lambda}\psi_j)}{\sum_{k=1}^{|\mathcal{V}|} \exp(\psi_i^\top \mathbf{\Lambda}\psi_k)}$.

From Lemma 1, we can see that, in order to minimize the norm of the gradient $\|\mathbf{J}\|_2$, we essentially need to make $\sum_{i=1}^{|\mathcal{V}|} \left|\psi_i - \sum_{j=1}^{|\mathcal{V}|} p_{i,j}\psi_j\right|^2$ small. When $\mathbf{\Lambda}$ is small and all the input time series token embeddings are centered at the origin, $\sum_{i=1}^{|\mathcal{V}|} \psi_i = 0$, we have $\sum_{i=1}^{|\mathcal{V}|} \left|\psi_i - \mathbf{\Psi}^\top p_{i,:}\right|^2 \approx \sum_{i=1}^{|\mathcal{V}|} \left|\psi_i - \mathbf{\Psi}^\top\mathbf{\Psi}\mathbf{\Lambda}\psi_i\right|^2$ (see Appendix B).

Next, we prove that $\mathbf{\Lambda}$ minimizes the objective $\sum_{i=1}^{|\mathcal{V}|} \left|\psi_i - \mathbf{\Psi}^\top\mathbf{\Psi}\mathbf{\Lambda}\psi_i\right|^2$ and contains the m largest eigenvectors of correlation matrix $\mathbf{\Psi}^\top\mathbf{\Psi}$ of time series token embeddings, where m is the rank of $\mathbf{\Lambda}$.

Theorem 2. *Let the eigenvalues of the correlation matrix* $\mathbf{\Psi}^\top\mathbf{\Psi}$ *be ordered as* $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_D$, *and let* $\gamma_i \in \mathbb{R}^D$ *for* $i = 1, \ldots, D$ *denote their associated eigenvectors. Then, the matrix* $\mathbf{\Lambda}^*$ *that minimizes the quantity* $\sum_{i=1}^{|\mathcal{V}|} \left|\psi_i - \mathbf{\Psi}^\top\mathbf{\Psi}\mathbf{\Lambda}\psi_i\right|^2$ *has the optimal form* $\mathbf{\Lambda} = \sum_{i=1}^{m} \frac{1}{\lambda_i}\gamma_i\gamma_i^\top$.

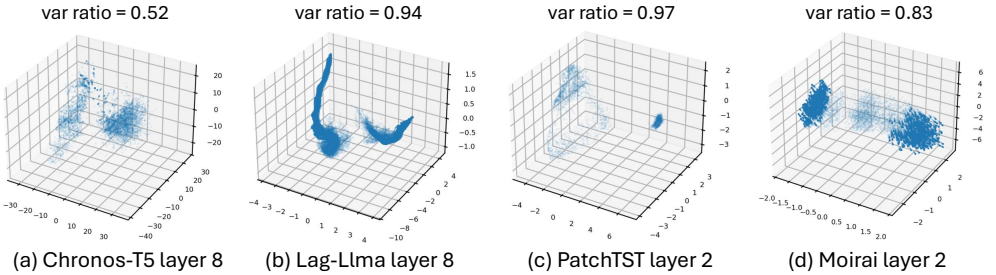
Proof. The proof of Theorem 2 is provided in Appendix C. □

Theorem 2 shows that the self-attention mechanism effectively projects input time series tokens onto a low-dimensional contextual embedding space defined by the top eigenvectors of the correlation matrix $\mathbf{\Psi}^\top\mathbf{\Psi}$. This result reveals that the self-attention mechanism in LLMs implicitly aligns with the dominant directions (i.e., top eigenvectors) of the contextual embedding space, and hence, suggesting that isotropy is not just a geometric artifact but a learned structural property that supports effective generalization to numerical downstream tasks.

While the self-attention aligns input representations with the dominant eigenvectors of the embedding space, the alignment may vary across different subregions of the contextual embedding space due to variations in the input sequences, token types, or contextual patterns. As a result, the degree of isotropy may differ across subregions of the contextual embedding space, which motivates the need to assess isotropy at a local (i.e., cluster) level rather than relying solely on a global metric. The next section explores these local structural patterns and examines the geometry of the hidden representations through principal component analysis (PCA), which helps reveal how variance is distributed across embedding dimensions.

4.3 Clusters in the Contextual Embedding Space

Motivated by the results of Lemma 1 and Theorem 2, this section investigates local structural patterns by projecting the LLMs' hidden representations into a lower-dimensional space using the top $m = 3$ eigenvectors via PCA, as shown in Figure 2. The three axes of the figure represent the first three principal components of the covariance matrix of LLM representations of each layer. For instance, in Figure 2b and 2d, the first three principal components account for 94% of the total variance in layer 8 of Lag-Llama and 83% in layer 2 of Moirai. From Figure 2 a, 2 b, 2 c and 2 d, we can see that there are disconnected or slightly overlapping islands that are far away from each other. In equation 1, the space isotropy is measured on pairs of arbitrary time series token representations, which could reside in two disconnected clusters. However, given that the variance is dominated by distances between clusters, such estimation would be biased by the inter-cluster



(a) Chronos-T5 layer 8 (b) Lag-Llma layer 8 (c) PatchTST layer 2 (d) Moirai layer 2

Figure 2: Isolated or slightly overlapping cluster islands exist in the contextual embedding space for all models. For brevity, we only show a few representative middle layers from each model.

distances. Hence, it is more reasonable to consider a per-cluster (i.e., local) investigation rather than a global estimate.

Isotropy within Clusters. We start by performing clustering on the LLM representations in the contextual embedding space. There are various methods for performing clustering, such as K-means and DBSCAN algorithm Ester et al. (1996). We select K-means clustering method because it is reasonably fast in high embedding dimensions. We use the classical silhouette score analysis Rousseeuw (1987) to determine the number of clusters $|C|$ in the contextual embedding space (see Appendix D for details). Since each LLM contextual embedding instance $\psi(k_i)$ belongs to a particular cluster through clustering, the cosine similarity should be measured after shifting the mean to the origin Mu and Viswanath (2018). Accordingly, we subtract the mean for each cluster (i.e., centroid) and calculate the adjusted $\zeta_{\cos}$ in Section 2. Assuming we have a total of $|C|$ clusters, let $\psi_c(k_i) = \{\psi_c^1(k_i), \psi_c^2(k_i), \dots\}$ be the set of token k's contextual embeddings in cluster $c \in C$, and $\psi_c(k_i)$ be one random sample in $\psi_c(k_i)$. We define the adjusted inter-token cosine similarity as

$$\zeta'_{\cos} \triangleq \mathbb{E}_c \left[\mathbb{E}_{i \neq j} \left[\cos \left(\bar{\psi}_c(k_i), \bar{\psi}_c(k_j) \right) \right] \right], \tag{6}$$

where $\bar{\psi}_c(k_i) = \psi_c(k_i) - \mathbb{E}_{\psi_c}[\psi_c(k_i)]$. Here $\mathbb{E}_c$ is the average over different clusters, and $\bar{\psi}_c(k_i)$ is the original contextual embedding shifted by the mean, with the mean taken over the samples in cluster c Kim et al. (2021). The inter-token cosine similarity takes values between -1 and 1. A value close to 0 indicates strong isotropy and ensures the existence of structure in the LLM representations.

To put it in a nutshell, this section provides a theoretical foundation showing that self-attention projects input tokens onto a low-dimensional subspace aligned with the dominant eigenvectors of the embedding correlation matrix. This alignment induces isotropy in LLM hidden representations, stabilizing the partition function and preserving the structure needed for reliable numerical downstream task performances. In Section 5, we extend this analysis by empirically evaluating how isotropy in different language models' hidden representations correlates with time series forecasting performances across a wide range of numerical datasets, varying context lengths, and noise levels.

5 Experiments

Baselines. We consider popular pre-trained LLMs as the baselines for numerical downstream tasks, including Chronos-T5 Ansari et al. (2024) and Choronos-Bolt (https://huggingface.co/autogluon/chronos-bolt-base), PatchTST Nie et al. (2023), Moirai-1.0-R Woo et al. (2024) and Lag-Llama Rasul et al. (2024). The considered models use different architectures, time series tokenization techniques and hyperparameters for numerical downstream tasks. For instance, Lag-Llma use decoder only transformer, PatchTST and Moirai-1.0-R use vanilla Transformer enoder, while Chronos-T5 and Choronos-Bolt use encoder-decoder transformer. Different baselines achieve contextual embedding in different ways. For example, PatchTST focuses on tokenizing time series as patches and uses self-attention for modeling dependencies within each patch and across patches, while CHORNOS-T5 and CHRONOS-Bolt adapt language modeling architectures minimally and generate categorical tokens by applying scaling and quantization. The details of these baselines are summarized in Table 2.

Table 2: LLM models architectures, time series tokenization techniques and hyperparameter choices. L stands for context length, d_h for hidden layer dimension, n_L for number of layers, n_H for number of heads, and η for learning rate.

Model	**Architecture**	**Tokenization Technique**	**Hyperparameters**
Chronos-T5	Encoder-Decoder with autoregressive forecasting	Scaling & Quantization	Default
Chronos-Bolt	Encoder-Decoder with multi-step forecasting	Scaling & Quantization	Default
PatchTST	Vanilla Encoder	Patching	Patch length: 16, Stride: 8, $d_h = 32, n_L = 2, n_H = 4$
Moirai	Encoder	Patching	$L = 1024$, Patch length: selected by dataset-specific validation
Lag-Llama	Decoder	Lag Feature	$L = 32$

Table 3: Real and Synthetic Datasets

Data Subset	**Domain**	**Dataset 1**	**Dataset 2**
Real Datasets	Energy	Australian Electricity – Queensland State	Australian Electricity – South Australia
	Weather	Solar Radiation	Rainfall
	Finance	Exchange Rate	NN5 Weekly Cash Withdrawals
	Healthcare	Hospital Patient Counts	COVID-19 Deaths
	Transportation	Transportation Signaling 1	Transportation Signaling 2
	Retail	Car Sales	Dominick
Synthetic Datasets	Linear	DotProduct kernel (C=0)	DotProduct kernel (C=1)
	seasonality	seasonality kernel (period = 0.5W)	seasonality kernel (period = 0.25H)
	Trend	RationalQuadratic kernel ($\alpha = 1$)	RationalQuadratic kernel ($\alpha = 10$)
	Non-Linear	RBF kernel (length scale = 0.1)	RBF kernel (length scale = 1)
	Stochastic	WhiteKernel (noise level = 0.1)	WhiteKernel (noise level = 1)

Datasets. We conduct a comprehensive evaluation using 12 different real time series datasets from various numerical domains, including energy, nature, finance, healthcare, retail and transportation. The sources of these open-source datasets along with their descriptions, including how each dataset is used across different LLM can be found in Table 4 of Appendix E. We also illustrate our findings using KernelSynth Ansari et al. (2024) (see Algorithm E in Appendix E for details), a method that generates 10 additional synthetic datasets via Gaussian processes in Section 5. We select two different datasets from each numerical domain (as shown in Table 3) and then perform qualitative analysis with synthetic datasets and quantitative analysis with real datasets. The results of these analyses are provided in the next two sections.

5.1 Qualitative Analysis

We now analyze the time series forecasting by the baseline LLMs qualitatively. We focus on synthetically generated time series for a controlled analysis of different types of time series patterns which belong to 5 different domains, such as linear, seasonality, trend, non-linear and stochastic. We are particularly interested in the isotropic measurement in the LLM's last layer as it is related to the logits and probabilistic inference as explained in Section 2. So all isotropic measure provided in this section is based on the last layer of the baselines.

We begin by analyzing time series forecasting performance (i.e., NMSE) for different baselines and its relation with isotropy in Figure 3. For instance, in Figure 3 b, we have (NMSE = 0.0000066 and cosine similarity = $|-0.00076|$) for seasonality-Dataset 1 and (NMSE = 0.00012 and cosine similarity = 0.0047) for seasonality-Dataset 2 for Chronos-T5. This shows that stronger isotropy exists (i.e., inter-token cosine similarity value is close to 0) in Chronos-T5's embedding space for seasonality-Dataset 1 which preserves the structure in its hidden representations and causes good downstream task performance. On the other hand, a weaker isotropy exists (i.e., inter-token cosine similarity value is far from 0) in Chronos-T5's embedding space for seasonality-Dataset 2, which, in turn, causes a lack of structure in its hidden representations, thereby leading to bad forecasting performance as compared to seasonality-Dataset 1. The NMSE and inter-type cosine similarity can also vary across different language models and datasets. For example, in Figure 3c, the

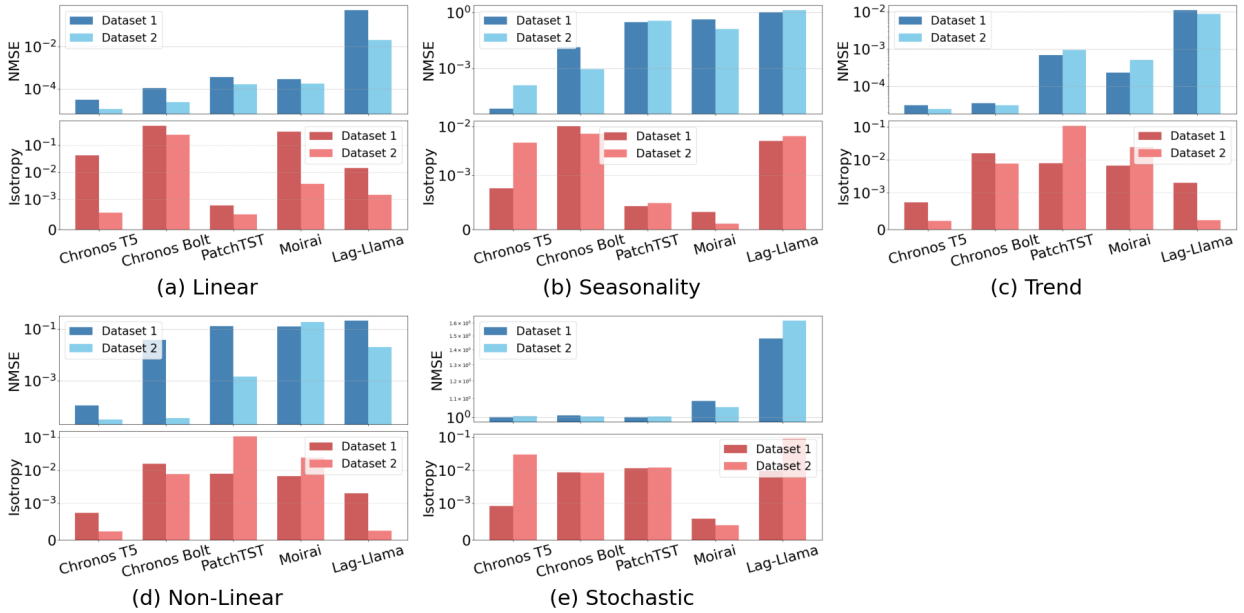


Figure 3: NMSE vs isotropy analysis for 10 different synthetic datasets of 5 different domains.

NMSE for trend-Dataset 1 is lower for PatchTST and Moirai, but higher for Chronos-T5, Chronos-Bolt, and Lag-Llama, compared to their respective NMSE on trend-Dataset 2. Conversely, for trend-Dataset 2, the NMSE is lower for Chronos-T5, Chronos-Bolt, and Lag-Llama, but higher for PatchTST and Moirai, compared to their respective NMSE on trend-Dataset 1. A similar analysis can also be observed for other synthetic datasets and baselines in Figures 3 b, 3 d, and 3 e. This shows that any dataset from any particular

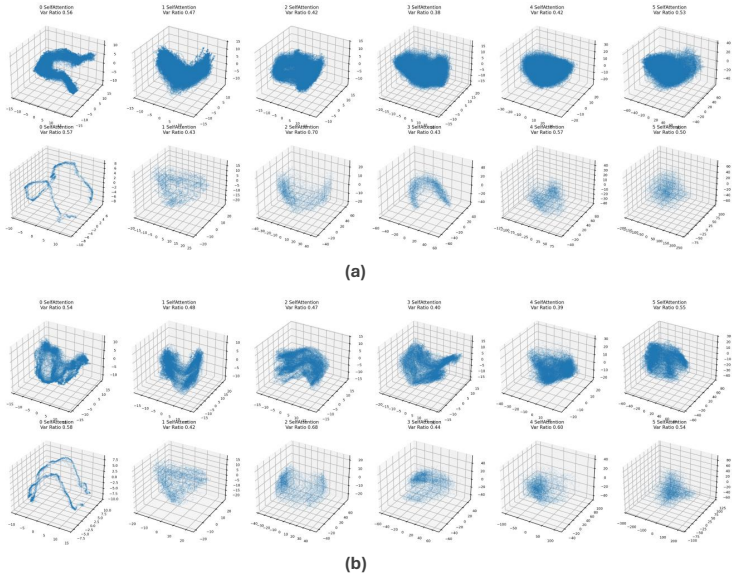


Figure 4: Variations in Chronos-T5's hidden representations for different input context lengths for the same synthetic dataset non-linear-Dataset 1 : (a) Contextual embedding space for input context length $L = 500$. (b) Contextual embedding space for input context length $L = 100$.

domain may cause different forecasting performances for different baselines, as it generates different hidden representations (see Appendix F for full visualization) in contextual embedding spaces, and hence, different isotropy measures.

Next, we examine the influence of isotropy on forecasting performance in two important scenarios: a) different input context lengths, and b) different levels of noises in the input data. The first scenario is important as it provides an analysis that helps guide in selecting reasonable input context lengths rather than selecting the length through random trials and errors. The second scenario is important as it gives us ideas on how the level of noise in noisy data impacts performance, since the data in the real world is mostly noisy.

Isotropy in different input context lengths. We first analyze the effect of isotropy under varying input context lengths. We begin with an illustration in Figure 4 where we show how the hidden representations of Chronos-T5 vary for two different input context lengths, such as $L = 500$ and $L = 100$, for non-linear-Dataset 1, which generates different isotropic measures for different input context lengths.

In Figure 5, we compare the NMSE vs isotropy across two different input context lengths, $L = 500$ and $L = 100$, for different synthetic datasets. We use Choronos-T5 as an example model for this experiment. As can be seen from the figure, the isotropy values vary across different input context lengths and datasets. For instance, in seasonality-Dataset 1, we have (NMSE= 0.0000066, cosine similarity= $|-0.00076|$) and (NMSE= 0.0793, cosine similarity= 0.0011) for $L = 500$ and $L = 100$, respectively. The decrease in isotropy significantly increases the NMSE for the input context length $L = 100$. In contrast, in linear-Dataset 2, we have (NMSE= 0.000025, cosine similarity= 0.2474) and (NMSE= 0.000009, cosine similarity= 0.0644) for $L = 500$ and $L = 100$, respectively. In this scenario, the isotropy increases for the input context length $L = 100$, which causes the decrease in NMSE for chornos-T5. In practice, the input context length is often selected randomly or through trial and error, which may cause higher forecasting errors for different datasets. Isotropy analysis enables us to understand how varying input context lengths influence the hidden representations of the language model. This insight helps guide improvements in forecasting performance by examining the isotropic properties of the contextual embedding space.

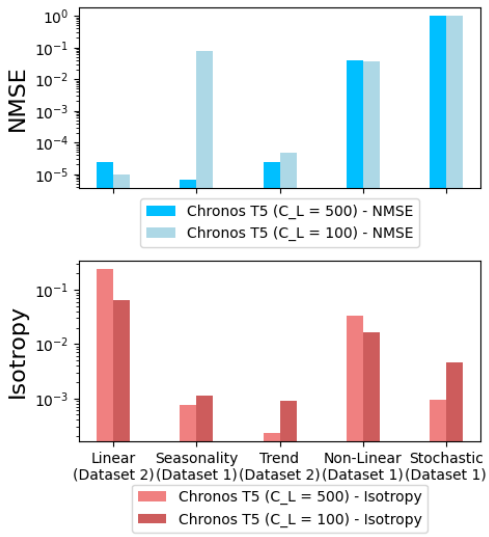


Figure 5: NMSE vs isotropy comparison across different input context lengths for synthetic datasets.

Isotropy in varying noise levels in datasets. Next, we focus on the second scenario to see the impact of noisy datasets on LLM's performance. Again, we use the Chornos-T5 as an example language model. Figure 6 compares the NMSE vs isotropy across two different cases, one without noise, and the other with Gaussian noise with a standard deviation $\sigma = 0.05$ standard deviation. From Figure 6, we can see consistently lower isotropy (i.e., inter-token cosine similarity far from 0) for all noisy synthetic datasets as compared to the datasets without noise. For instance, in trend-Dataset 2, we have (NMSE= 0.000024, cosine similarity= $|-0.00022|$) and (NMSE= 0.0012, cosine similarity= 0.0040) for $\sigma = 0$ and $\sigma = 0.05$, respectively. The decrease in isotropy significantly increases the NMSE for the noisy dataset. In practice, many real-world numerical domains—such as those in nature and energy—exhibit noisy and dynamic behavior. In these environments, it is often infeasible to measure noise in real time or to pre-process the input time series

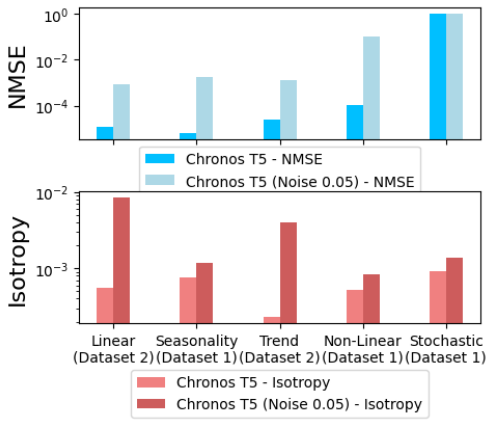


Figure 6: NMSE vs isotropy comparison across different noise levels in synthetic datasets.

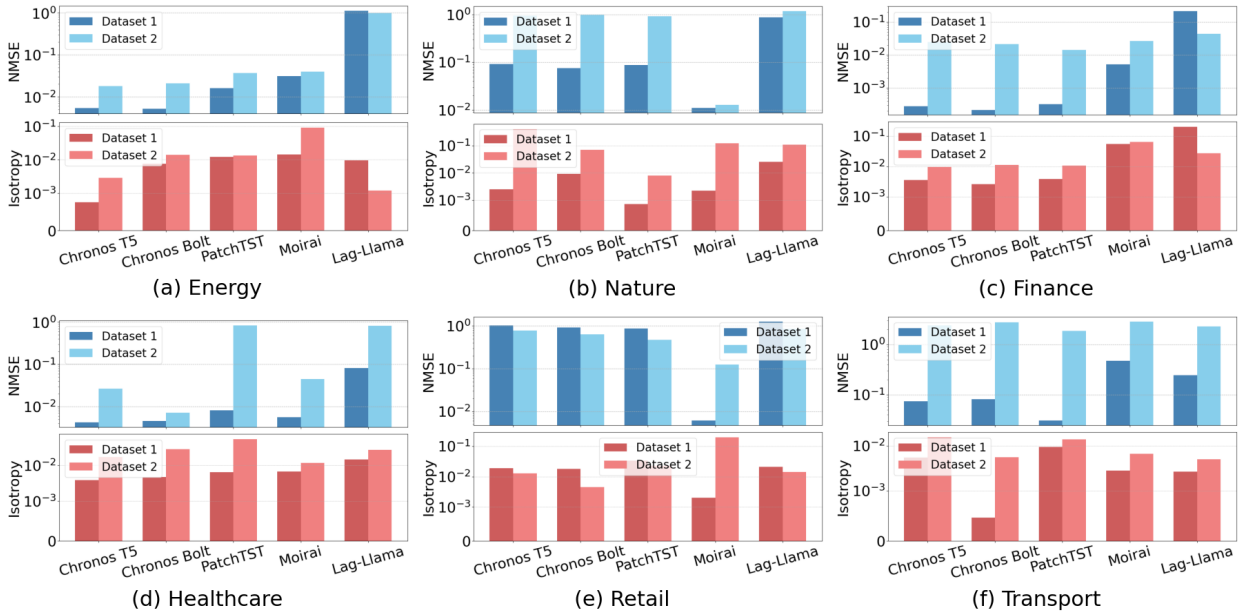


Figure 7: NMSE vs isotropy analysis for 12 different real datasets of 6 different domains.

for improved performance. However, the isotropy in the hidden representations of LLMs can be readily measured, and thus, can be leveraged to enhance forecasting performance by identifying and mitigating the effects of noisy inputs in contextual embedding space.

5.2 Quantitative Analysis

Next, we present our main results on 12 real datasets which belong to 6 different numerical domains including energy, nature, finance, healthcare, retail, and transportation. As our qualitative analysis in Section 5.1, we select two different datasets from each numerical domain and the isotropy measure from LLM's last layer to show the impact of isotropy on NMSE performance for different language models.

In Figure 7, we analyze the time series forecasting performance of different baselines and its relation with isotropy for different real datasets. For instance, in Figure 7 e, we have (NMSE = 0.0061 and cosine similarity = 0.0020) for retail-Dataset 1 and (NMSE = 0.1255 and cosine similarity = 0.1931) for retail-Dataset 2 for Moirai. This indicates the existence of stronger isotropy in Moirai's embedding space for retail-Dataset 1 which preserves the structure in its hidden representations and causes good downstream task performance. On the other hand, a weaker isotropy exists in Moirai's embedding space for retail-Dataset 2, which yields a lack of structure in its hidden representations and, consequently, bad downstream task performance as compared to retail-Dataset 1. The NMSE and inter-type cosine similarity can vary across different real datasets and language models. For example, in Figure 7a, the NMSE for energy-Dataset 1 is lower for Chronos-T5, Chronos-Bolt, PatchTST, and Moirai, but higher for Lag-Llama, compared to their respective NMSE on energy-Dataset 2. Conversely, the NMSE for energy-Dataset 2 is lower for Moirai but higher for the other baselines, compared to their respective NMSE on energy-Dataset 1. A similar analysis can also be observed for other synthetic datasets and baselines in Figure 7 c and 7 e. This again shows that datasets from the same numerical domain can cause varying forecasting performance across different baselines, as they generate distinct hidden representations in contextual embedding spaces, and hence, different isotropy measures, depending on the language model architecture and tokenization strategy.

Finally, in Figure 8, we compare the NMSE vs isotropy for varying input context lengths to observe its impact on the real datasets. We select Lag-Llma as our example model. We compare the results for two different input context lengths: 1) the recommended input context length $L = 144$ and the reduced input context length $L = 96$. As can be seen from the figure, the inter-token cosine similarity values become far from 0, i.e., from 0.0097 to 0.0220 for energy-Dataset 1 and from 0.0026 to 0.0103 for transport-Dataset 1, which in turn decreases the NMSE performances.

On the other hand, the inter-token cosine similarity values become close to 0, i.e., from 0.1091 to 0.0112 for nature-Dataset 2 and from 0.2014 to 0.0133 for finance-Dataset 1, which in turn improves the NMSE performances. Thus, the variation in the recommended input context length may not only decrease the NMSE performances, but can also increase for some datasets.

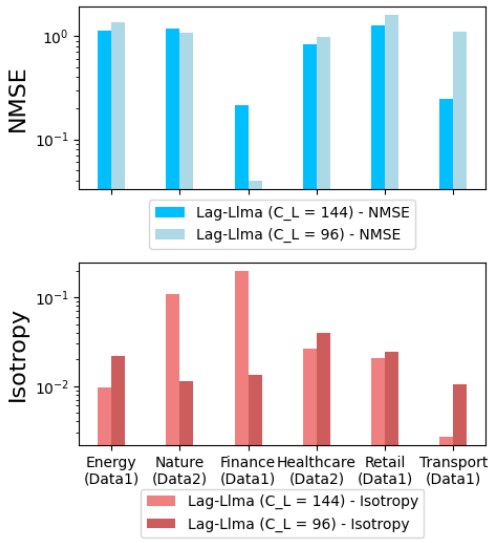


Figure 8: NMSE vs isotropy comparison across different input context lengths for real datasets.

6 Conclusion and Limitations

In this work, we introduced a novel approach to investigate the role of isotropy in LLM hidden representations for numerical downstream tasks. By deriving an upper bound for the Jacobian matrix which collects all first-order partial derivatives of self-attention with respect to the input pattern, we showed that the self-attention mechanism implicitly aligns with the dominant eigenvectors of the input correlation structure and induces isotropy in the contextual embedding space. The existence of isotropy in the contextual embedding space was found to stabilize the partition function and enable better generalization in numerical downstream tasks across different models and datasets. Our empirical analysis across 10 synthetic and 12 real numerical datasets, and 5 different language models further validated the consistent relationship between isotropy and forecasting performance, highlighting isotropy as a reliable indicator of structured representation learning. These insights open up a new interpretability frontier for LLMs in numerical domains.

While isotropy offers a principled way to preserve useful structure, there may be alternative approaches to approximating the partition function and guiding numerical reasoning. Moreover, developing mechanisms to recover or enhance structure when isotropy is weak remains an important avenue for future work. Ultimately, we believe that incorporating structural insights like isotropy into the LLM design pipeline can significantly improve their reliability and adaptability to numerical domains.

References

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Syndar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, and Maddix et al. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*, 2024.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. A latent variable model approach to PMI-based word embeddings. volume 4, pages 385–399, Cambridge, MA, 2016. MIT Press. doi: 10.1162/tacl_a_00106.

Vadim Borisov, Kathrin Sessler, Tobias Leemann, Martin Pawelczyk, and Gjergji Kasneci. Language models are realistic tabular data generators. In *The Eleventh International Conference on Learning Representations*, 2023.

Xingyu Cai, Jiaji Huang, Yuchen Bian, and Kenneth Church. Isotropy in the contextual embedding space: Clusters and manifolds. In *International Conference on Learning Representations*, 2021.

Tuan Dinh, Yuchen Zeng, Ruisu Zhang, Ziqian Lin, Michael Gira, Shashank Rajput, Jy-yong Sohn, Dimitris Papailiopoulos, and Kangwook Lee. Lift: Language-interfaced fine-tuning for non-language machine learning tasks. *Advances in Neural Information Processing Systems*, 35:11763–11784, 2022.

Samuel Dooley, Gurnoor Singh Khurana, Chirag Mohapatra, Siddartha Naidu, and Colin White. Forecastpfn: Synthetically-trained zero-shot forecasting, 2023. URL `https://arxiv.org/abs/2311.01933`.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, page 226–231. AAAI Press, 1996.

Kawin Ethayarajh. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. In *Conference on Empirical Methods in Natural Language Processing*, 2019.

Kuofeng Gao, Yang Bai, Jindong Gu, Shu-Tao Xia, Philip Torr, Zhifeng Li, and Wei Liu. Inducing high energy-latency of large vision-language models with verbose images. In *ICLR*, 2024.

Azul Garza and Max Mergenthaler-Canseco. Timegpt-1. *arXiv preprint arXiv:2310.03589*, 2023.

Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew Gordon Wilson. Large language models are zero-shot time series forecasters, 2024. URL `https://arxiv.org/abs/2310.07820`.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-LLM: Time series forecasting by reprogramming large language models. 2024.

Hyunjik Kim, George Papamakarios, and Andriy Mnih. The lipschitz constant of self-attention. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5562–5571. PMLR, 18–24 Jul 2021.

Boxun Liu, Xuanyu Liu, Shijian Gao, Xiang Cheng, and Liuqing Yang. Llm4cp: Adapting large language models for channel prediction. *Journal of Communications and Information Networks*, 9(2):113–125, 2024. doi: 10.23919/JCIN.2024.10582829.

Jiaqi Mu and Pramod Viswanath. All-but-the-top: Simple and effective postprocessing for word representations. In *International Conference on Learning Representations*, 2018.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*, 2023.

Kashif Rasul, Arjun Ashok, Andrew Robert Williams, Hena Ghonia, Rishika Bhagwatkar, Arian Khorasani, Mohammad Javad Darvishi Bayazi, George Adamopoulos, Roland Riachi, Nadhir Hassen, Marin Biloš, Sahil Garg, Anderson Schneider, Nicolas Chapados, Alexandre Drouin, Valentina Zantedeschi, Yuriy Nevmyvaka, and Irina Rish. Lag-llama: Towards foundation models for probabilistic time series forecasting, 2024. URL https://arxiv.org/abs/2310.08278.

Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987. ISSN 0377-0427.

Dandan Wang and Shiqing Zhang. Large language models in medical and healthcare fields: applications, advances, and challenges. *Artificial Intelligence Review*, 57(299):1–27, 2024.

Colin Wei, Sang Michael Xie, and Tengyu Ma. Why do pretrained language models help in downstream tasks? an analysis of head and prompt tuning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 16158–16170. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/86b3e165b8154656a71ffe8a327ded7d-Paper.pdf.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers, 2024. URL https://arxiv.org/abs/2402.02592.

Chenwei Wu, Holden Lee, and Rong Ge. Connecting pre-trained language model and downstream task via properties of representation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Shengzhe Xu, Christo Kurisummoottil Thomas, Omar Hashash, Nikhil Muralidhar, Walid Saad, and Naren Ramakrishnan. Large multi-modal models (lmms) as universal foundation models for ai-native wireless systems. *Netwrk. Mag. of Global Internetwkg.*, 38(5):10–20, July 2024. ISSN 0890-8044. doi: 10.1109/MNET.2024.3427313.

Xinli Yu, Zheng Chen, Yuan Ling, Shujing Dong, Zongyi Liu, and Yanbin Lu. Temporal data meets llm–explainable financial time series forecasting. *arXiv preprint arXiv:2306.11025*, 2023.

A Proof of Theorem 1

Theorem 1. *Let the logits of the ground-truth model be bounded. Then for any $f^*(k,l)$, there exists a set of functions $\{\hat{z}_i(k,l)\}_{i=1}^{|\mathcal{V}|}$ such that for all k and T_{l+1}, the predictive distribution of the student model $\hat{p}\left(k_{T_{l+1}} \,|\, \mathbf{k}_{1:T_l}\right)$ matches that of ground-truth model $p^*(k_{T_{l+1}} \,|\, \mathbf{k}_{1:T_l})$ and $\hat{f}(k,l) = 0$. In other words, there exists a student model with the same pre-training loss as the ground-truth model, but its logits are ineffective for the numerical downstream tasks.*

Proof. We select $\tau \in \mathbb{R}$ such that $\forall k, T_{l+1}$, $\tau < \min_{j\in\mathcal{V}} b_j^* - \max_{j\in\mathcal{V}} z_i^*(k,l)$, and $\forall k, T_{l+1}, \forall j \in \mathcal{V}$. By setting $\hat{z}_j(k,l) := z_i^*(k,l) + \tau$, we get $\forall j \in \mathcal{V}$,

$$\hat{z}_j(k,l) - b_j^* < z_i^*(k,l) + \min_{j\in\mathcal{V}} b_j^* - \max_{j\in\mathcal{V}} z_i^*(k, T_{l+1}) - b_j^* \leq 0,$$

this implies that $\sigma(\hat{z}_j(k,l) - b_j^*) = 0$. Hence, $\forall k, T_{l+1}$ and we have $\hat{f}(k,l) = 0$. □

B Proof of Lemma 1

Lemma 1. *Consider the Jacobian matrix $\mathbf{J} = \left[\frac{\partial g_i(\mathbf{\Psi})}{\partial \psi_j}\right]_{i,j=1}^{|\mathcal{V}|}$, which represents the gradient of the self-attention mapping $G(\mathbf{\Psi})$ with respect to the input time series token embeddings. Then the spectral norm of $\mathbf{J}$ satisfies $\|\mathbf{J}\|_2 \leq |\mathbf{\Lambda}|_2 \sum_{i=1}^{|\mathcal{V}|} \left(p_{i,i} + \frac{1}{2}\right) \left|\psi_i - \sum_{j=1}^{|\mathcal{V}|} p_{i,j}\psi_j\right|^2 + \Delta$, where the residual term Δ is given by $\Delta = |\mathbf{\Lambda}|_2 \sum_{i\neq j}^{|\mathcal{V}|} p_{i,j} \left|\psi_j - \sum_{q=1}^{|\mathcal{V}|} p_{i,q}\psi_q\right|^2 + \frac{|\mathbf{\Lambda}|_2}{2} \sum_{j=1}^{|\mathcal{V}|} |\psi_j|^2$, and the attention weights $p_{i,j}$ are defined as $p_{i,j} = \frac{\exp(\psi_i^\top \mathbf{\Lambda}\psi_j)}{\sum_{k=1}^{|\mathcal{V}|} \exp(\psi_i^\top \mathbf{\Lambda}\psi_k)}$.*

Proof. According to the analysis, the gradient of $g_i(\mathbf{\Psi})$ with respect to the variable $\boldsymbol{\psi}_j$ is expressed as $J_{i,j} = \frac{\partial g_i(\mathbf{\Psi})}{\partial \psi_j} = p_{i,j} I + \mathbf{\Psi}^\top Q^i \left(\mathbf{\Psi}\mathbf{\Lambda}\delta_{i,j} + E_{j,i}\mathbf{\Psi}\mathbf{\Lambda}^\top\right)$ where the matrix Q^i is defined by $Q^i = \mathrm{diag}(p_{i,:}) - p_{i,:}p_{i,:}^\top$. Here, $p_{i,:} \in \mathbb{R}_+^{|\mathcal{V}|}$ corresponds to the i-th row of the probability matrix $\boldsymbol{P}$, $E_{j,i} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ denotes a matrix with a single entry at the (j,i)-th position and zeros elsewhere, and $\delta_{i,j} \in \{0,1\}$ is the Kronecker delta. We thus have

$$\begin{aligned}
\|\mathbf{J}\|_2 &\leq \sum_{i,j=1}^{|\mathcal{V}|} |J_{i,j}|_2 \\
&\leq \sum_{i,j=1}^{|\mathcal{V}|} p_{i,j} + \sum_{i=1}^{|\mathcal{V}|} |\mathbf{\Psi}^\top Q^i \mathbf{\Psi}|_2 |\mathbf{\Lambda}|_2 + \sum_{i,j=1}^{|\mathcal{V}|} |\mathbf{\Psi}^\top Q^i E_{j,i} \mathbf{\Psi}|_2 |\mathbf{\Lambda}|_2 \\
&\leq |\mathcal{V}| + |\mathbf{\Lambda}|_2 \sum_{i=1}^{|\mathcal{V}|} \left(\sum_{j=1}^{|\mathcal{V}|} p_{i,j} |\psi_j|^2 - \left| \sum_{j=1}^{|\mathcal{V}|} p_{i,j}\psi_j \right|^2 \right) + |\mathbf{\Lambda}|_2 \sum_{i,j=1}^{|\mathcal{V}|} |\mathbf{\Psi}^\top Q^i e_j \psi_i^\top| \\
&\leq |\mathcal{V}| + |\mathbf{\Lambda}|_2 \sum_{i=1}^{|\mathcal{V}|} \sum_{j=1}^{|\mathcal{V}|} p_{i,j} |\psi_j - \sum_{q=1}^{|\mathcal{V}|} p_{i,q}\psi_q|^2 + |\mathbf{\Lambda}|_2 \sum_{i,j=1}^{|\mathcal{V}|} p_{i,j} |\psi_i^\top (\psi_j - \mathbf{\Psi}^\top p_{i,:})| \\
&\leq |\mathbf{\Lambda}|_2 \sum_{i=1}^{|\mathcal{V}|} \left(p_{i,i} + \frac{1}{2}\right) |\psi_i - \mathbf{\Psi}^\top p_{i,:}|^2 + |\mathcal{V}| + |\mathbf{\Lambda}|_2 \sum_{i\neq j}^{|\mathcal{V}|} p_{i,j} |\psi_j - \mathbf{\Psi}^\top p_{i,:}|^2 + \frac{|\mathbf{\Lambda}|_2}{2} \sum_{j=1}^{|\mathcal{V}|} |\psi_i|^2 \\
&= |\mathbf{\Lambda}|_2 \sum_{i=1}^{|\mathcal{V}|} \left(p_{i,i} + \frac{1}{2}\right) |\psi_i - \mathbf{\Psi}^\top p_{i,:}|^2 + |\mathcal{V}| + |\mathbf{\Lambda}|_2 \sum_{i\neq j}^{|\mathcal{V}|} p_{i,j} \left|\psi_j - \sum_{q=1}^{|\mathcal{V}|} p_{i,q}\psi_q\right|^2 + \frac{|\mathbf{\Lambda}|_2}{2} \sum_{j=1}^{|\mathcal{V}|} |\psi_i|^2
\end{aligned}$$

□

Theorem 2 shows that $\mathbf{\Lambda}$ minimizing the objective $\sum_{i=1}^{|\mathcal{V}|} |\psi_i - \mathbf{\Psi}^\top \mathbf{\Psi}\mathbf{\Lambda}\psi_i|^2$ contains the largest m eigenvectors of the correlation matrix $\mathbf{\Psi}^\top\mathbf{\Psi}$ of input time series token embeddings where m is the rank of $\mathbf{\Lambda}$.

Lemma 1 implies that one of the key components in the Jacobian's upper bound takes the form $|\psi_i - \sum_{j=1}^{|\mathcal{V}|} p_{i,j}\psi_j|^2$. Consequently, during optimization, it is natural to aim for a reduction in the gradient magnitude, which motivates minimizing the expression $\sum_{i=1}^{|\mathcal{V}|} |\psi_i - \sum_{j=1}^{|\mathcal{V}|} p_{i,j}\psi_j|^2$. This leads to understand the choice of $\mathbf{W}^Q$ and $\mathbf{W}^K$ that minimize $\sum_{i=1}^{|\mathcal{V}|} |\psi_i - \sum_{j=1}^{|\mathcal{V}|} p_{i,j}\psi_j|^2$, which is equivalent to solving the optimization problem $\min_{|\mathbf{\Lambda}|_F \leq \rho} \sum_{i=1}^{|\mathcal{V}|} |\psi_i - \sum_{j=1}^{|\mathcal{V}|} p_{i,j}\psi_j|^2$, where the scalar constraint ρ regulates the size of $\mathbf{\Lambda}$.

To proceed, we consider the objective in the scenario where ρ is small. In this case, we can approximate the attention weights by $p_{i,j} \approx \frac{1}{|\mathcal{V}|} + \frac{1}{|\mathcal{V}|}\psi_i^\top \mathbf{\Lambda}\psi_j$. Now, we define the average of embedding as $\bar{\psi} = \mathbf{\Psi}^\top 1/|\mathcal{V}|$. It then follows that $\sum_{i=1}^{|\mathcal{V}|} |\psi_i - \mathbf{\Psi}^\top p_{i,:}|^2 = \sum_{i=1}^{|\mathcal{V}|} |\psi_i - \bar{\psi} - \mathbf{\Psi}^\top\mathbf{\Psi}\mathbf{\Lambda}\psi_i|^2$. Assuming all input time series patterns are zero-centered, i.e., $\bar{\psi} = 0$, we have $\sum_{i=1}^{|\mathcal{V}|} |\psi_i - \mathbf{\Psi}^\top\mathbf{\Psi}\mathbf{\Lambda}\psi_i|^2 = \mathrm{tr}\left((I - \mathbf{\Psi}^\top\mathbf{\Psi}\mathbf{\Lambda})^2\mathbf{\Psi}^\top\mathbf{\Psi}\right)$. Theorem 2 establishes that the optimal $\mathbf{\Lambda}$ that minimizes $\sum_{i=1}^{|\mathcal{V}|} |\psi_i - \mathbf{\Psi}^\top\mathbf{\Psi}\mathbf{\Lambda}\psi_i|^2$ is spanned by the top m eigenvectors of $\mathbf{\Psi}^\top\mathbf{\Psi}$, where m equals the rank of $\mathbf{\Lambda}$.

C Proof of Theorem 2

Theorem 2. *Let the eigenvalues of the correlation matrix $\mathbf{\Psi}^\top\mathbf{\Psi}$ be ordered as $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_D$, and let $\gamma_i \in \mathbb{R}^D$ for $i = 1, \ldots, D$ denote their associated eigenvectors. Then, the matrix $\mathbf{\Lambda}^*$ that minimizes the quantity $\sum_{i=1}^{|\mathcal{V}|} \left|\psi_i - \mathbf{\Psi}^\top\mathbf{\Psi}\mathbf{\Lambda}\psi_i\right|^2$ has the optimal form $\mathbf{\Lambda} = \sum_{i=1}^{m} \frac{1}{\lambda_i}\gamma_i\gamma_i^\top$.*

Proof. Given that $\mathbf{W}_Q \in \mathbb{R}^{D\times m}$ and $\mathbf{W}_K \in \mathbb{R}^{D\times m}$, it follows that the matrix $\mathbf{\Lambda}$ has rank m. Hence, we know $\min_{\mathbf{\Lambda}} \sum_{i=1}^{|\mathcal{V}|} \|\psi_i - \mathbf{\Psi}^\top\mathbf{\Psi}\mathbf{\Lambda}\psi_i\|^2 \geq \sum_{q=m+1}^{|\mathcal{V}|} \lambda_q$. Now, if we set $\mathbf{\Lambda}$ to $\mathbf{\Lambda} = \sum_{i=1}^{m} \frac{1}{\lambda_i}\gamma_i\gamma_i^\top$, then we obtain $\sum_{i=1}^{|\mathcal{V}|} \|\psi_i - \mathbf{\Psi}^\top\mathbf{\Psi}\mathbf{\Lambda}\psi_i\|^2 = \mathrm{tr}\left(\left(I - \sum_{i=1}^{m}\gamma_i\gamma_i^\top\right)^2\mathbf{\Psi}^\top\mathbf{\Psi}\right) = \sum_{q=m+1}^{D}\lambda_q$.

Therefore, the optimal solution $\mathbf{\Lambda}$ for minimizing $\sum_{i=1}^{|\mathcal{V}|} \|\psi_i - \mathbf{\Psi}^\top\mathbf{\Psi}\mathbf{\Lambda}\psi_i\|^2$ is essentially characterized as a linear combination of the top m eigenvectors of $\mathbf{\Psi}^\top\mathbf{\Psi}$. Since a small gradient will prefer a small quantity of $\sum_{i=1}^{|\mathcal{V}|} \|\psi_i - \mathbf{\Psi}^\top\mathbf{\Psi}\mathbf{\Lambda}\psi_i\|^2$, the self-attention mechanism implicitly drives the weight matrices $\mathbf{W}_Q$ and $\mathbf{W}_K$ to align with the dominant eigen-directions of $\mathbf{\Psi}^\top\mathbf{\Psi}$. □

D Clustering in the Contextual Embedding Space

Clustering. We begin with the isotropy assesmment by performing clustering on the LLM representations in the contextual embedding space. There are various methods for performing cultering, such as k-means, DBSCAN Ester et al. (1996). We select K-means clustering method because it is reasonably fast in high embedding dimensions (e.g., $d \geq 768$ for GPT2, ELMo, BERT etc.). We use the celebrated silhouette score analysis Rousseeuw (1987) to determine the number of clusters $|C|$ in the contextual embedding space. After performing K-means clustering, each observation p (i.e., one of the $\mathbf{J}$ vector representations in $\mathcal{V}$) is assigned to one of C clusters. For an observation p assigned to the cluster $c \in C$, we compute the silhouette score as follows

$$a(p) = \frac{1}{|C| - 1} \sum_{q\in C, p\neq q} \mathrm{dist}(p, q); \qquad b(p) = \min_{\bar{c}\neq c} \sum_{q\in\bar{c}} \mathrm{dist}(p, q); \qquad s(p) = \frac{b(p) - a(p)}{\max b(p), a(p)},$$

where $a(p)$ is the mean distance between an observation p and the rest in the same cluster class p, while $b(p)$ measures the smallest mean distance from p-th observation to all observations in the other cluster class. After computing the silhouette scores $s(p)$ of all observations, a global score is computed by averaging the individual silhouette values, and the partition (with a specific number of clusters) of the largest average score is pronounced superior to other partitions with a different number of clusters. We select the best $|C|$ that belongs to the partition that scores highest among the other partitions.

E Dataset Description

Real Datsets.

Table 4: The complete list of datasets used for our quantitative and qualitative analysis. The table is divided into three sections, representing how the datasets were used for baseline models.

Dataset	Domain	Freq.	Num. Series	Series Length			Prediction
				min	avg	max	Length (H)
Australian Electricity	Energy	30min	5	230736	231052	232272	48
Car Parts	Retail	1M	2674	51	51	51	12
Covid Deaths	Healthcare	1D	266	212	212	212	30
Dominick	Retail	1D	100014	201	296	399	8
Exchange Rate	Finance	1B	8	7588	7588	7588	30
FRED-MD	Economics	1M	107	728	728	728	12
Hospital	Healthcare	1M	767	84	84	84	12
NN5 (Weekly)	Finance	1W	111	113	113	113	8
Weather	Nature	1D	3010	1332	14296	65981	30
Transportaion Signal	Transport	1D	3010	1332	14296	65981	30
Synthetic (10 kernels)	Numerical	-	1000000	1024	1024	1024	64

Synthetic Datasets. We use KernelSynth Ansari et al. (2024), a method to generate synthetic dataset using Gaussian processes (GPs). KernelSynth allows generation of large, diverse datasets tailored to specific patterns or statistical properties, which is particularly useful when real-world data is scarce or incomplete. In this synthetic data generation process, the GPs are defined by a mean function, $\mu(t)$, and a positive definite kernel, $\kappa(x_i, x_j)$, which specifies a covariance function for variability across input pairs (x_i, x_j). A kernel bank $\mathcal{K}$ (which consists of linear, RBF, and periodic kernels) is used to define diverse time series patterns. The final kernel $\tilde{\kappa}(x_i, x_j)$ is constructed by sampling and combining kernels from $\mathcal{K}$ using binary operations like $+$ and $\times$. Synthetic time series are generated by sampling from the GP prior, $GP(\mu(t) = 0, \tilde{\kappa}(x_i, x_j))$. The following algorithm presents the pseudocode for KernelSynth which essentially follows the approach in Ansari et al. (2024).

Algorithm 1 KernelSynth: Generating Synthetic Sequences via Gaussian Process Kernels

Input: Kernel bank $\mathcal{K}$, maximum kernels per time series $J = 5$, and length of the time series $l_{\text{syn}} = 1024$.
Output: A synthetic time series $\mathbf{x}_{1:l_{\text{syn}}}$.

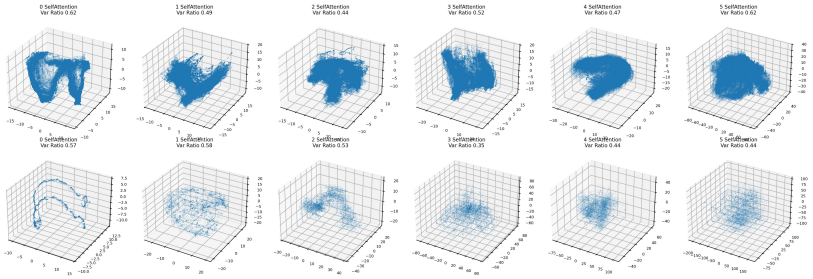
$j \sim \mathcal{U}\{1, J\}$ ▷ sample the number of kernels
$\{\kappa_1(t,t'), \ldots, \kappa_j(t,t')\} \overset{\text{i.i.d}}{\sim} \mathcal{K}$ ▷ sample j kernels from the Kernel bank $\mathcal{K}$
$\kappa^*(t,t') \leftarrow \kappa_1(t,t')$
for $i \leftarrow 2$ to j **do**
    $\star \sim \{+, \times\}$ ▷ pick a random operator (add or multiply)
    $\kappa^*(t,t') \leftarrow \kappa^*(t,t') \star \kappa_i(t,t')$ ▷ compose kernels
end for
$\mathbf{x}_{1:l_{\text{syn}}} \sim \mathcal{GP}(0, \kappa^*(t,t'))$ ▷ draw a sample from the GP prior
return $\mathbf{x}_{1:l_{\text{syn}}}$

F Full Visualization of PCA plots for different models

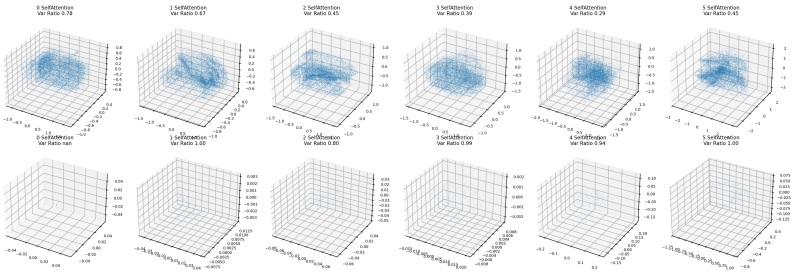
The full visualization of PCA plots of different models is provided below. We use the synthetic Dataset 1, and Dataset 2 from non-linear domain for illustration.

Non-Linear (Dataset 1):

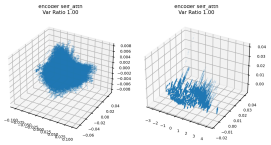
Chronos-T5



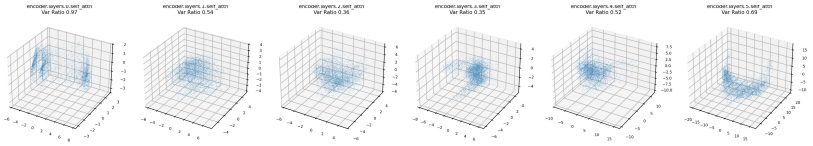
Chronos-Bolt



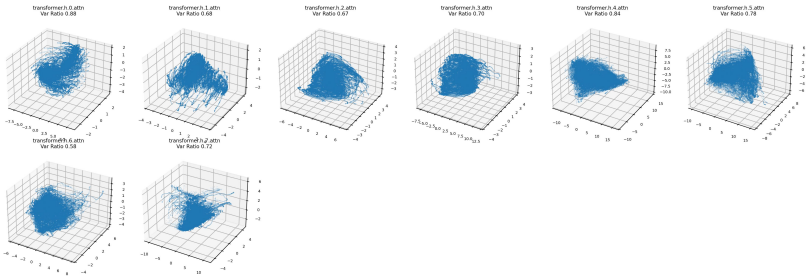
PatchTST



Morai

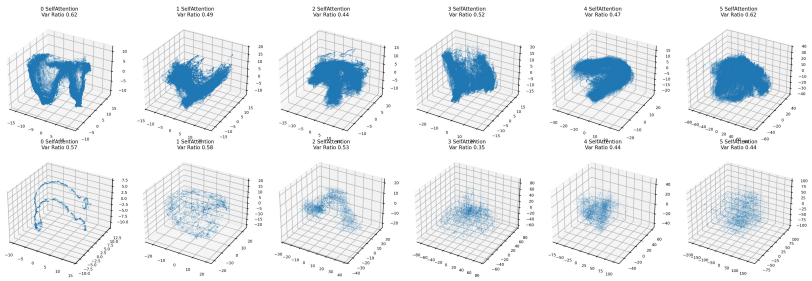


Lag-Llma

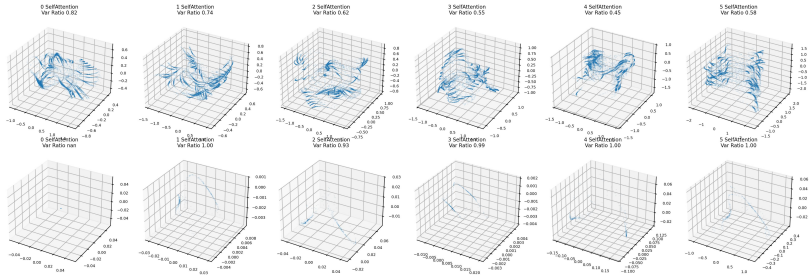


Non-Linear (Dataset 2):

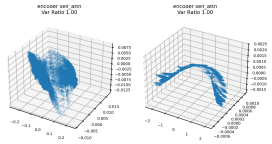
Chronos-T5



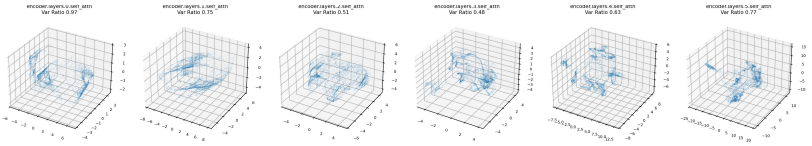
Chronos-Bolt



PatchTST



Morai



Lag-Llma