Is isotropy a good proxy for generalization in transformer-based time series forecasting?

Anonymous authors
Paper under double-blind review

Abstract

Vector representations of contextual embeddings learned by transformer-based models are effective in various downstream tasks in *numerical domains* such as time series forecasting. The significant success of emergent transformer-based models in capturing long-range dependencies and contextual semantics has led to their widespread integration into modern large language models (LLMs). However, 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 provide prediction reliability in numerical domains, it is necessary to open the black box behind transformer-based models and develop explanatory tools that can serve as good proxies for performance. However, there is little theoretical understanding of when transformer-based models help solve generalization in time series forecasting tasks. This paper seeks to bridge this gap by understanding when the next-token prediction capability of transformer-based models can be adapted to time series forecasting through a novel analysis based on the concept of isotropy in the contextual embedding space. Specifically, a log-linear model is considered as a *simplified abstraction* for transformer-based models in which time series data can be predicted from its context through a network with softmax in the output layer (i.e., model head in self-attention). For this model, it is demonstrated that, in order to achieve state-of-the-art performance in time series forecasting, the hidden representations of the transformer-based model embeddings must possess a structure that accounts for the shift-invariance of the softmax function. By formulating a gradient structure of self-attention in transformer-based models, it is shown how the isotropic property of model embeddings in contextual embedding space preserves the underlying structure of representations, thereby resolving the shift-invariance problem and insights into model reliability and generalization. Experiments across 22 different numerical datasets and 5 different transformer-based 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

Transformer-based models have been proven to be 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. The significant success of emergent transformer-based models in capturing long-range dependencies and contextual semantics has led to their widespread integration into modern 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 transformer-based model 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 transformer-based models 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 the protection onset of sepsis or cardiac arrest in healthcare can result in patient deaths. Thus, to provide prediction reliability in numerical domains, it is necessary to open the black box behind transformer-based models and develop explanatory tools that can serve as good proxies for performance. 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 transformer-based models 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-token prediction capability of transformer-based models be effectively adapted to time series forecasting?"*

The main contribution of this paper is a novel approach for answering this question by exploiting the isotropic property of transformer-based model 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 transformer-based models must exhibit *a structured form* in contextual embedding space that accounts for the shift-invariance problem (Singla et al. (2021); Jacobsen et al. (2020); Rojas-Gomez et al. (2022)) 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 transformer-based models, we show how the isotropic property of transformer-based model 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 (Arora et al. (2016); Mu and Viswanath (2018), Andreas and Klein (2015); Peters and Klakow (2000); Nelakanti et al. (2013)) as a *simplified abstraction* for transformer-based models 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 transformer-based 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 transformer-based model 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 transformer-based models. Our experiments demonstrate that the isotropy of transformer-based model 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

Overview of Section 2. In this section, a general problem setup is presented for transformer-based modeling of numerical time series. Time series inputs are tokenized and processed through self-attention layers, leading to output distributions derived via softmax over inner products, an operation that aligns with the log-linear model. This abstraction enables a tractable foundation for isotropy analysis in the contextual embedding space. A numerical downstream task is also defined in terms of logits, casting time series forecasting as a categorical prediction problem. Together, these formulations establish the theoretical basis for studying the relationship between isotropy and forecasting performance.

Time Series Tokens and Similarity Measure. Similar to next-token prediction by transformer-based models, 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 transformer-based 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 transformer-based model 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 in numerical domains (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 computing the predictive distribution $p(k_l \,|\, \mathbf{k}_{-l})$, for $l \in \{1, \ldots, L\}$, where $\mathbf{k}_{-l}$ denotes the tokenized time series sequence and k_i be a time series token in time series vocabulary $\mathcal{V}$. In transformer-based forecasting models, this prediction is performed using self-attention layers that generate transformer-based model hidden representations in the contextual embedding space. These contextual representations are then combined with token embeddings via inner products and passed through a softmax function to yield the probability distribution over the time series vocabulary $\mathcal{V}$.

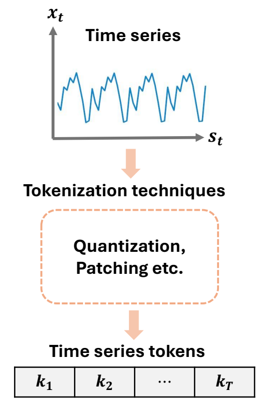


Figure 1: Time series tokenization.

Let $\tilde{\boldsymbol{\Psi}}_i = \{\psi_i^1, \psi_i^2, \ldots\}$ be the set of all transformer-based model contextual embedding instances of time series token k_i. Here, different contexts in the time series sequences yield different transformer-based model embeddings of k_i. By constructing $\sum_{k_i} |\tilde{\boldsymbol{\Psi}}_i| = |\mathcal{V}|$, we define the inter-token cosine similarity as:

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

where ψ_i and ψ_j are random samples from $\tilde{\boldsymbol{\Psi}}_i$. 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.

Model. We consider a general model for numerical data and open the black box of the transformer-based model. To this end, we adopt log-linear model (Arora et al. (2016); Mu and Viswanath (2018), Andreas and Klein (2015); Peters and Klakow (2000); Nelakanti et al. (2013)) as a simplified abstraction for transformer-based model that allows us to relate the geometry of contextual embeddings with output probabilities, which is essential for our theoretical analysis. Specifically, we assume that the observation probability of next time series token k_l given the input time series sequence $\mathbf{k}_{-l}$ satisfies the log-linear model as

$$p^*(k_l = x_t \mid \mathbf{k}_{-l}) \propto \exp(\langle \psi^*_{-l}(\mathbf{k}_{-l}), \psi^*_t \rangle), \tag{2}$$

where $\psi^*_t \in \mathbb{R}^D$ is a ime series token $k_{i\in\mathcal{V}}$ embedding (i.e., vector) that only depends on the time series observation x_t, $\psi^*_{-l}(\mathbf{k}_{-l})$ is a function that encodes the tokenized time series sequence $\mathbf{k}_{-l}$ into a vector (i.e., contextual represenation) in $\mathbb{R}^D$ and the notation $\langle . \rangle$ denotes the inner product between $\psi^*_{-l}(\mathbf{k}_{-l})$ and ψ^*_t. The inference mechanism of transformer-based models inherently follows a log-linear structure in the prediction step. The transformer-based models compute the probability of the next token using a softmax over inner products between the contextual representation$\psi^*_{-l}(\mathbf{k}_{-l})$ and the token embeddings ψ^*_t (i.e., *output of the self-attention layers*), which is similar to the form of the log-linear model in equation 2. Moreover, we do not consider any prior distribution for input, which makes our model more general than previous latent models in (Arora et al. (2016); Wei et al. (2021)).

To define the numerical downstream task, let $z_i^*(k,l) := \langle \psi^*_{-l}(\mathbf{k}_{-l}), \psi^*_t \rangle$ 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 $Z^*(k,l) := \sum_{i=1}^{|\mathcal{V}|} \exp(z_i^*(k,l))$ be the partition function (Arora et al. (2016)), i.e., normalization factor.

In transformer-based models, 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

$$p^*(k_l = x_t \mid \mathbf{k}_{-l}) = \frac{\exp(\langle \psi^*_{-l}(\mathbf{k}_{-l}), \psi^*_t \rangle)}{Z^*(k,l)} = \frac{\exp(z^*_i(k,l))}{Z^*(k,l)}.$$

Since we do not know the ground-truth model in reality, we do not have access to the ground-truth model components ψ^*_t and $\psi^*_{-l}(\mathbf{k}_{-l})$. Instead, we only have access to the trained model ψ_t and $\psi_{-l}(\mathbf{k}_{-l})$ that aims to achieve low pre-training loss. We can define the student logits as $\mathbf{z}(k,l) := \{\langle \psi_{-l}(\mathbf{k}_{-l}), \psi_t \rangle\}_{i=1}^{|\mathcal{V}|}$. 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 trained model $\forall i \in \mathcal{V}$ can be defined as

$$p(k_l = x_t \mid \mathbf{k}_{-l}) = \frac{\exp(z_i(k,l))}{Z(k,l)}. \tag{3}$$

Loss Function. As typical in transformer-based models, we use the categorical distribution over the elements in the time series vocabulary $\mathcal{V}$ as the output distribution $p(k_l = x_t|\mathbf{k}_{-l})$, for $l \in \{1, \dots, L\}$, where $\mathbf{k}_{-l}$ is the tokenized time series. The transformer-based model is trained to minimize the cross entropy between the distribution of the tokenized ground truth label and the predicted distribution. The loss function we use in this work is cross-entropy loss (Ansari et al. (2024); Wu et al. (2023))

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

where $p(k_l = x_t \mid \mathbf{k}_{-l})$ is the categorical distribution predicted by the trained model, $p^*(k_l = x_t|\mathbf{k}_{-l})$ is the distribution of ground-truth model, $\mathcal{D}_{\mathrm{KL}}$ is the KL divergence, and $\mathcal{H}(p^*(k_l = x_t| \mathbf{k}_{-l}))$ is the entropy of distribution $p^*(k_l = x_t|\mathbf{k}_{-l})$ which is a constant. We assume that the transformer-based model undergoes large-scale training to achieve a small loss so that the KL-divergence term in equation 4 is also small. This aligns with standard assumptions in representation learning where a trained model approximates the ground-truth distribution, thereby making the KL-divergence small in practice. The use of KL-divergence and its connection to cross-entropy in equation 4 provide a theoretical foundation to analyze how learned representations in contextual embedding space approximate a true distribution, even when models are trained with different loss functions.

Downstream Numerical Task. We consider numerical forecasting as a downstream task, where a trained transformer-based model is adapted to predict the next token in a time series. To model this, the continuous time series values are mapped into a finite vocabulary of discrete tokens to cast the time series forecasting as a categorical prediction problem. The downstream task is then defined as a function of the logits $\mathbf{z}^*(k,l)$ generated by the trained model where each logit corresponds to a token in the time series vocabulary $\mathcal{V}$ given the input $\mathbf{k}_{-l}$. For simplicity, a linear model can be considered as a downstream task whose prediction on categorical distribution is linear in $\psi^*_{-l}(\mathbf{k}_{-l})$, that is, $f^*(k,l) = \langle \psi^*_{-l}(\mathbf{k}_{-l}), u^* \rangle = \sum_{i=1}^{|\mathcal{V}|} a^*_i z^*_i(k,l)$, where $u^* = \sum_{i=1}^{|\mathcal{V}|} a^*_i \psi^*_t \in \mathbb{R}^D$ and a_j is the coefficient. While this model helps interpret predictions, it is not sufficient to guarantee generalization in unseen scenarios. This is because the KL-divergence is less sensitive to sign changes in $f^*(k,l)$ when logits have low magnitude Wu et al. (2023)). Then, it is more reasonable to model the numerical downstream task as

$$f^*(k,l) = \sum_{i=1}^{|\mathcal{V}|} a^*_i \sigma(z^*_i(k,l) - b^*_i) = \sum_{i=1}^{|\mathcal{V}|} a^*_i \sigma\left(\langle \psi^*_{-l}(\mathbf{k}_{-l}), \psi^*_t \rangle - b^*_i\right),$$

where σ is the ReLU function and b^*_j denotes the threshold for the logits. This formulation suppresses low-confidence logits that contribute minimally to KL divergence, focusing the task on high-relevance predictions.

The downstream numerical task modeling reflects common transformer-based model strategies and improves robustness in time series forecasting tasks.

Despite the empirical success of transformer-based 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 transformer-based model 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 transformer-based models to Numerical Data

Overview of Section 3. This section addresses a key gap in understanding when and why transformer-based models generalize reliably to numerical downstream tasks such as time series forecasting. While strong training performance is often observed, it does not always translate to reliable inference, particularly across diverse numerical domains. To address this mismatch, a theoretical framework is introduced based on the isotropic property of the contextual embedding space. It is shown that strong isotropy stabilizes the partition function and resolves softmax shift-invariance, thereby enabling more robust generalization. The formal connection between isotropy and inference reliability is established to motivate its use as an indicator of model performance in numerical settings.

As discussed in Section 2, we consider transformer-based model 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. Shift invariance has been extensively studied in the deep learning literature (e.g., Singla et al. (2021); Jacobsen et al. (2020); Rojas-Gomez et al. (2022)) and is recognized as a practical concern across various domains such as NLP and vision. For instance, (Singla et al. (2021)) shows that transformers can be sensitive to input shifts, and (Jacobsen et al. (2020)) shows that such invariance can distort representation geometry and affect generalization. In our context, softmax's shift invariance to uniform logit shifts means representational changes may not impact output unless the structure of the embedding space is preserved. Since we do not have any control over the logit shift of the trained model on unseen data, good prediction in time series forecasting during training does not necessarily provide good prediction in time series forecasting on unseen scenarios. The lack of reliability on prediction due to uncontrolled logit shifts on unseen 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 trained model $\hat{p}\,(k_l \,|\, \mathbf{k}_{-l})$ matches that of ground-truth model $p^*(k_l \,|\, \mathbf{k}_{-l})$ and $\hat{f}(k,l) = 0$. In other words, there exists a trained 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 transformer-based model embeddings, the trained 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 transformer-based model representations learned by the transformer-based 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 $\mathbf{\Psi} = (\psi_1, \ldots, \psi_{|\mathcal{V}|})^\top \in \mathbb{R}^{|\mathcal{V}| \times D}$ 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

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

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, $I(\{\psi_i\})$ becomes close to 1 which indicates that the contextual embedding space 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 1 and the need for structural constraints in transformer-based model representations, we analyze the effective dimensionality and cluster organization of transformer-based model's hidden representations in the contextual embedding space. These analyses reveal how isotropy manifests in transformer-based models and how its presence correlates with the model's ability to generalize to time series forecasting across different numerical domains, and hence, provides a reliability on predictions. 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 to empirical generalizability.

4 Study of isotropy in transformer-based models hidden representations

Overview of Section 4 This section builds on the theoretical foundation of isotropy to address the empirical question of how isotropy can be measured and interpreted in practice, and how it relates to generalization across numerical domains. It introduces techniques such as spectral alignment, principal component analysis (PCA), and cluster-based isotropy metrics to examine the structural geometry of hidden representations in transformer-based models. These methods uncover how isotropy manifests in model embeddings and how its presence correlates with forecasting performance. Through this analysis, the section establishes a crucial link between theoretical reliability and empirical generalizability.

Analysis Settings. For this study, we consider five different transformer-based 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-Llama	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

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

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 transformer-based models still perform well in numerical domains. This counterintuitive result motivate us to look deeper into the contextual embedding space.

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., $g_i(\mathbf{\Psi}) = \mathrm{softmax}(\mathbf{\Psi}\mathbf{\Lambda}\mathbf{\Psi}^\top)\mathbf{\Psi}$, 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 transformer-based models enables generalization in numerical domains. The proof of this lemma follows the analysis in (Kim et al. (2021)) and is provided in Appendix B for completeness.

Lemma 1. *Consider the Jacobian matrix* $\mathbf{J}$[2] *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* $\|\mathbf{J}\|_2 \leq |\mathbf{\Lambda}|_2 \sum_{i=1}^{|\mathcal{V}|}\left(p_{i,i} + \frac{1}{2}\right)\left|\psi_i - \sum_{j=1}^{|\mathcal{V}|} p_{i,j}\psi_j\right|^2 + \Delta$.

The residual term Δ and the station weight $p_{i,j}$ is defined in Appendix B. For notatioanl simplicity, we express the term $\sum_{i=1}^{|\mathcal{V}|}\left|\psi_i - \sum_{j=1}^{|\mathcal{V}|} p_{i,j}\psi_j\right|^2$ in Lemma 1 as Γ. From Lemma 1, we can see that, in order to minimize the norm of the gradient $\|\mathbf{J}\|_2$, we essentially need to make Γ 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 $\Gamma \approx \sum_{i=1}^{|\mathcal{V}|}\left|\psi_i - \mathbf{\Psi}^\top\mathbf{\Psi}\mathbf{\Lambda}\psi_i\right|^2$ (see Appendix B).

Next, we prove that $\mathbf{\Lambda}$ minimizes the objective Γ 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 $\mathbf{\Lambda}$.

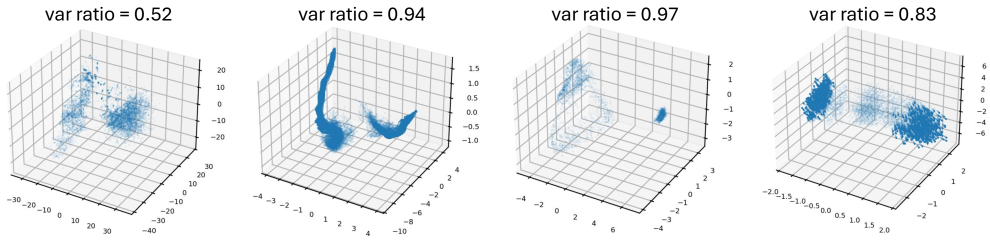
Theorem 2. *Let the eigenvalues of the correlation matrix* $\mathbf{\Psi}^\top\mathbf{\Psi}$ *be ordered as* $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_D$, *and let* $\gamma_i \in \mathbb{R}^D$ *for* $i = 1, \ldots, D$ *denote their associated eigenvectors. Then, the matrix* $\mathbf{\Lambda}^*$ *that minimizes the quantity* Γ *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 transformer-based models 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.

[2]The Jacobian matrix $\mathbf{J} = \left[\frac{\partial g_i(\mathbf{\Psi})}{\partial \psi_j}\right]_{i,j=1}^{|\mathcal{V}|}$ represents the gradient of the self-attention mapping $G(\mathbf{\Psi})$ with respect to the input time series token embeddings.



(a) Chronos-T5 layer 8 (b) Lag-Llama layer (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.

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 transformer-based models' 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 transformer-based model 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 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 transformer-based model 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 transformer-based model contextual embedding instance ψ_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 $\Phi_{i_c} = \{\psi_{i_c}^1, \psi_{i_c}^2, \dots\}$ be the set of token k_i's contextual embeddings in cluster $c \in C$, and ψ_{i_c} be one random sample in Φ_{i_c}. We define the adjusted inter-token cosine similarity as

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

where $\bar{\psi}_{i_c} = \psi_{i_c} - \mathbb{E}_{\psi_{i_c}}[\psi_{i_c}]$. Here $\mathbb{E}_c$ is the average over different clusters, and $\bar{\psi}_{i_c}$ 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 transformer-based model representations. The stability of the partition function $Z(k, l)$ depends on balanced inter-token similarities (i.e., strong isotropy) in the contextual embedding space. However, as shown in Section 4.2, hidden representations often form disconnected or weakly overlapping clusters where global isotropy can be misleading due to the variance being dominated by large distances between cluster centroids. By analyzing local isotropy within each cluster in equation 6, meaningful intra-cluster geometry can be captured which ensures that no cluster disproportionately skews the partition function normalization. This leads to more stable and interpretable model outputs.

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 transformer-based model 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 transformer-based models' hidden representations correlates with time series forecasting performances across a wide range of numerical datasets, varying context lengths, and noise levels.

5 Experiments

Overview of Section 5 This section empirically validates the theoretical insights developed in the preceding sections by evaluating the relationship between isotropy and forecasting performance across a diverse set of transformer-based models and numerical time series datasets. It examines how factors such as model architecture, input tokenization, context length, and noise influence the isotropic structure of learned representations. By analyzing performance across multiple domains and settings, the experiments demonstrate that higher isotropy consistently correlates with better generalization, reinforcing its utility as a reliable, label-free diagnostic for transformer-based model robustness in time series forecasting tasks.

Baselines. We consider popular transformer-based models as the baselines for numerical downstream tasks, including Chronos-T5 (Ansari et al. (2024)) and Chronos-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-Llama use decoder only transformer, PatchTST and Moirai-1.0-R use vanilla Transformer encoder, while Chronos-T5 and Chronos-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 Chronos-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: transformer-based 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 transformer-based model 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 transformer-based models 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 (through equation 6) in the transformer-based model'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.

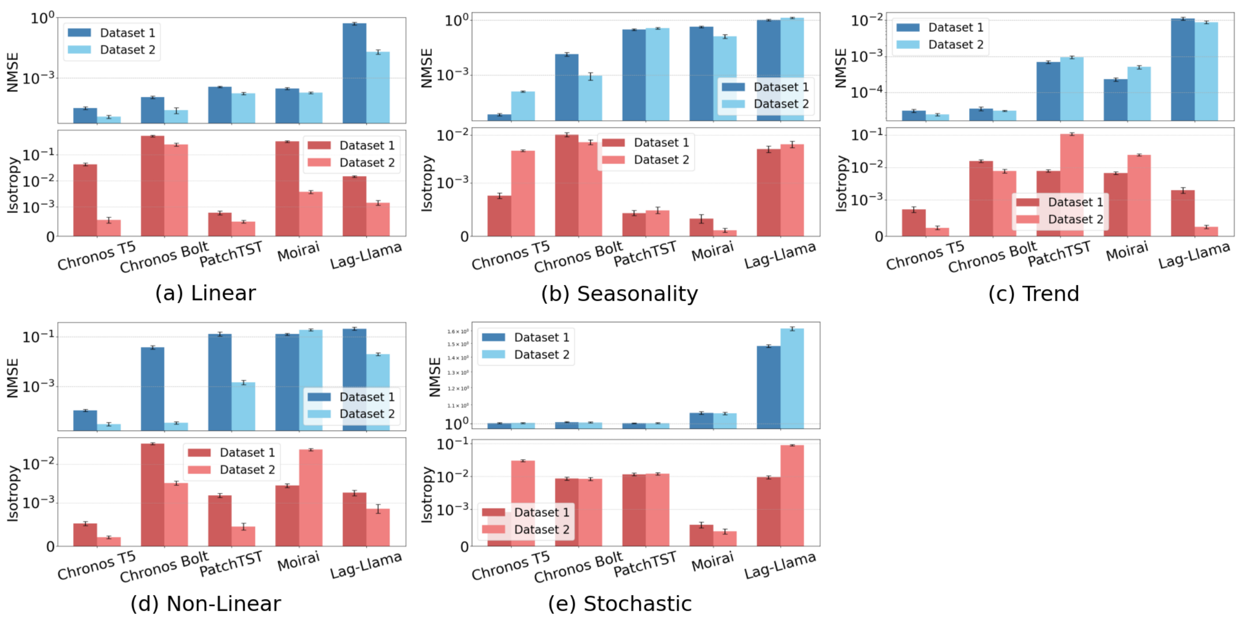


Figure 3: NMSE vs isotropy analysis for 10 different synthetic datasets of 5 different domains.

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 = $|-0.00076|$[3]) 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-token cosine similarity can also vary across different transformer-based models and datasets. For example, in Figure 3c, the 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

[3] The inter-token cosine similarity value close to zero indicates strong isotropy, with zero representing perfect isotropy. Since both positive and negative deviations from the origin reduce isotropy, we report absolute values (e.g., $|-0.x|$) to emphasize the distance from zero, which is the quantity of interest.

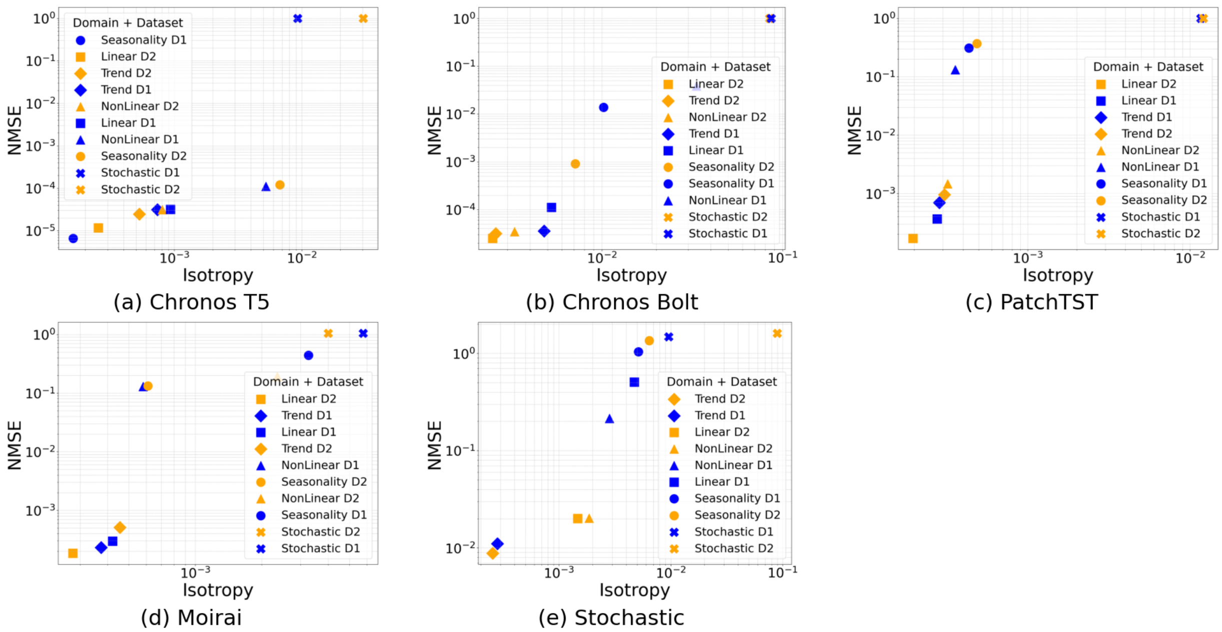


Figure 4: NMSE vs isotropy across numerical domains. The relationship between NMSE and isotropy is consistent across domains, highlighting that stronger isotropy (i.e., inter-token cosine similarity in equation 6 close to 0) leads to lower NMSE and vice versa.

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

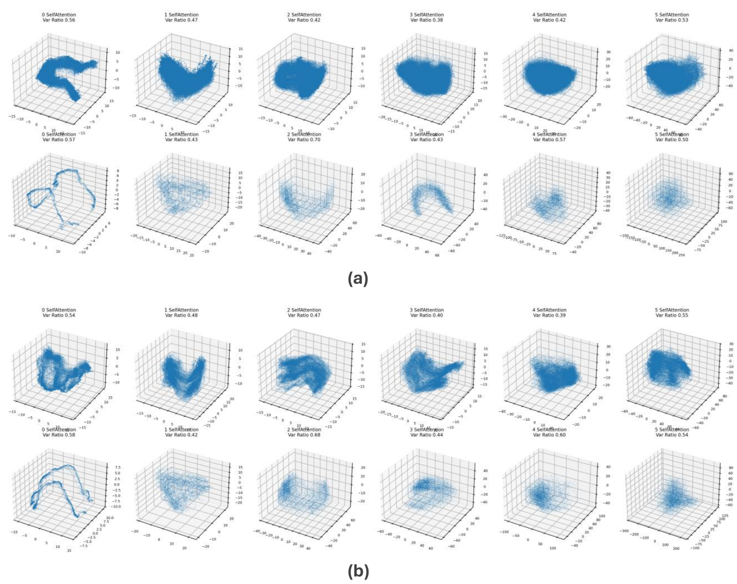


Figure 5: Variations in Chronos-T5's hidden representations for different input context lengths for the same synthetic dataset non-linear-Dataset 1 : (a) Contextual embedding space for input context length $L = 500$. (b) Contextual embedding space for input context length $L = 100$.

any particular 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. In Figure 4, we show NMSE vs isotropy comparison across numerical domains. From the figure, it can be seen that a consistent relationship exists between NMSE and isotropy, which justifies our findings: stronger isotropy leads to lower NMSE and vice versa.

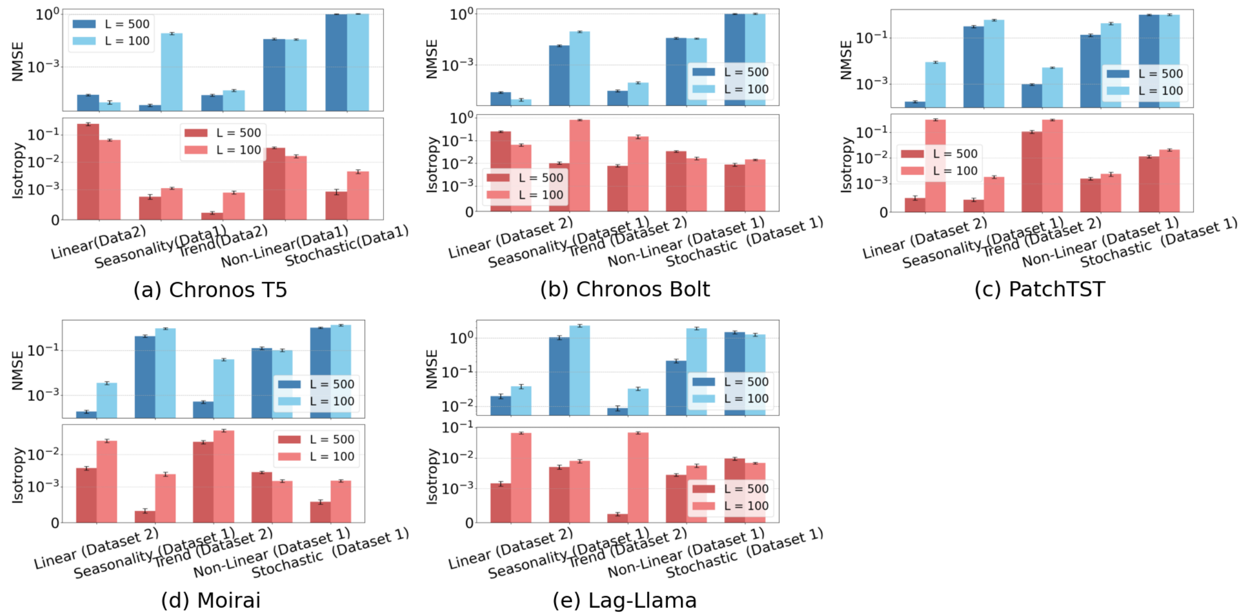


Figure 6: NMSE vs isotropy comparison across different input context lengths for synthetic datasets

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 5 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 6, we compare the NMSE vs isotropy across two different input context lengths, $L = 500$ and $L = 100$, for different synthetic datasets and transformer-based models. As can be seen from the figure, the isotropy values vary across different input context lengths and datasets. For instance, in Figure 6 a, we have (NMSE$=$ 0.0000066, cosine similarity$=$ $|-0.00076|$) and (NMSE$=$ 0.0793, cosine similarity$=$ 0.0011) for $L = 500$ and $L = 100$, respectively, for Chronos-T5 with seasonality-Dataset 1. 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 chronos-T5. A similar analysis can also be observed for other synthetic datasets and baselines in Figures 6 b, 6 c, 6 d, and 6 e. 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 transformer-based model. This insight helps guide improvements in forecasting performance by examining the isotropic properties of the contextual embedding space.

Isotropy in varying noise levels in datasets. Next, we focus on the second scenario to see the impact of noisy datasets on transformer-based model's performance. Figure 7 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 7, 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,

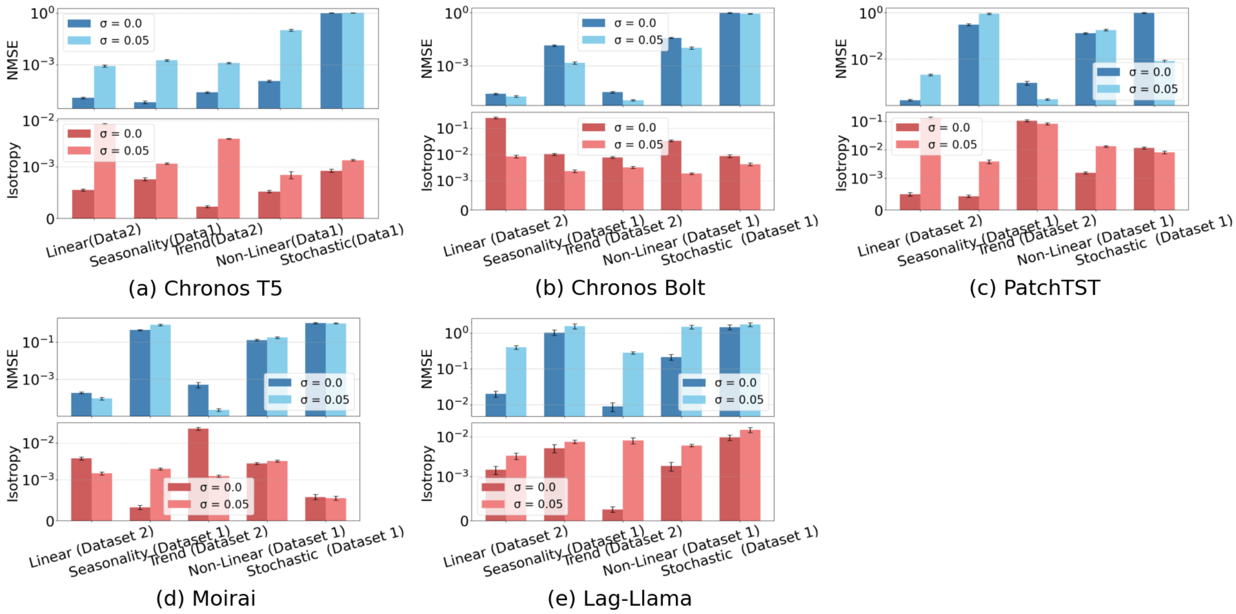


Figure 7: NMSE vs isotropy comparison across different noise levels in synthetic datasets.

in Figure 7 a, 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, for Chronos T5 with trend-Dataset 2. The decrease in isotropy significantly increases the NMSE for the noisy dataset. A similar analysis can also be observed for other synthetic datasets and baselines in Figures 7 b, 7 c, 7 d, and 7 e. 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 for improved performance. However, the isotropy in the hidden representations of transformer-based models 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.

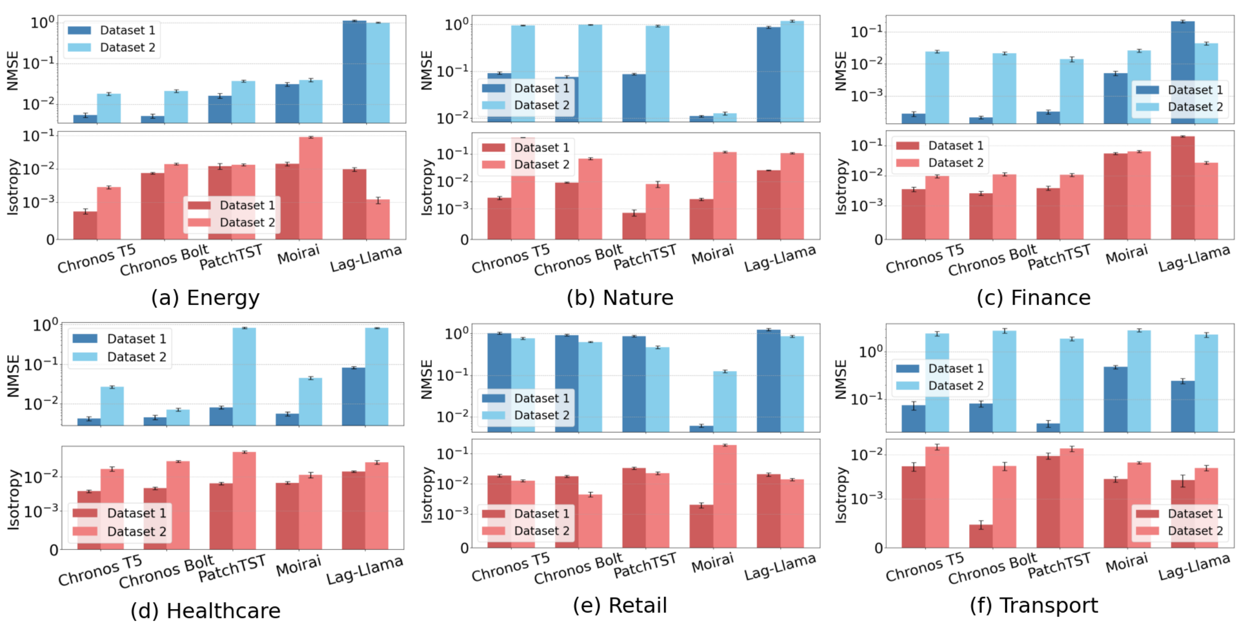


Figure 8: NMSE vs isotropy analysis for 12 different real datasets of 6 different domains.

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 transformer-based model's last layer to show the impact of isotropy on NMSE performance for different transformer-based models. As before, the isotropy measure in the figures of this section corresponds to equation 6)

In Figure 8, we analyze the time series forecasting performance of different baselines and its relation with isotropy for different real datasets. For instance, in Figure 8 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-token cosine similarity can vary across different real datasets and transformer-based models. For example, in Figure 8a, 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 8 c and 8 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 transformer-based model architecture and tokenization strategy.

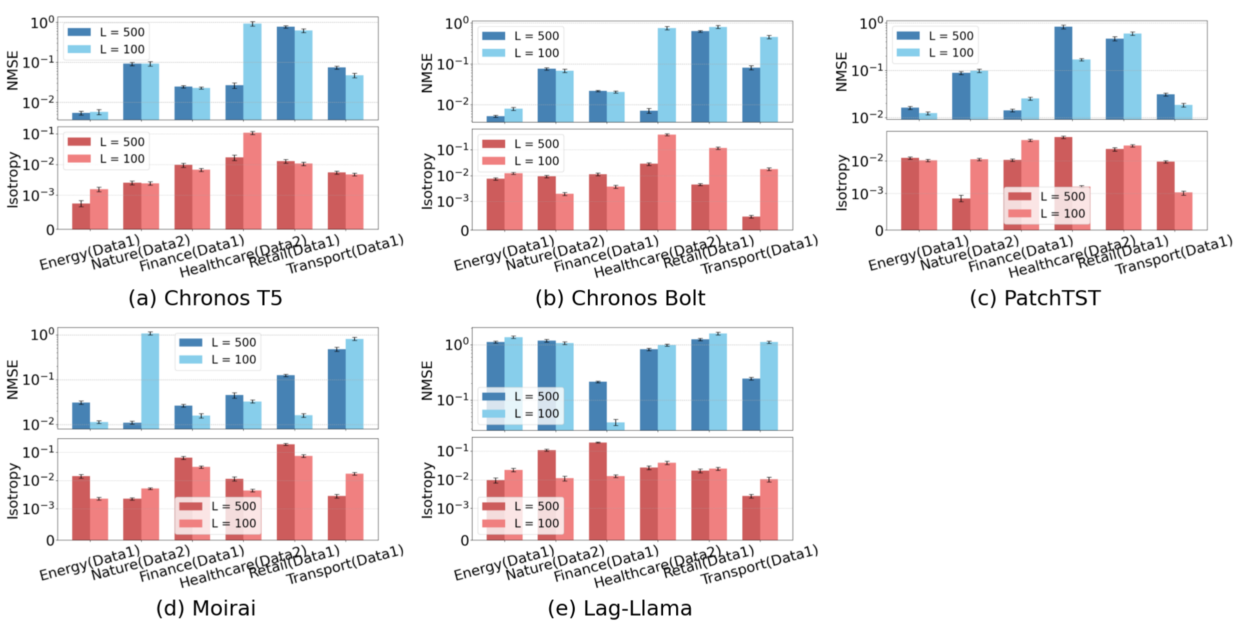


Figure 9: NMSE vs isotropy comparison across different input context lengths for real datasets.

Finally, in Figure 9, we compare the NMSE vs isotropy for varying input context lengths to observe its impact on the real datasets. 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 9 e, the inter-token cosine similarity values in Lag-Llama 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 in Lag-Llama 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. A similar analysis can also be observed for other synthetic datasets and baselines in Figures 9 a, 9 b, 9 c, and 9 d. Thus, the variation in the recommended input context length may not only decrease the NMSE performances, but can also increase for some datasets.

In Summary, while error metrics are essential, they are retrospective and require labeled data. In contrast, isotropy offers a label-free, structure-aware diagnostic of embedding quality that reflects a transformer-based model's generalization potential. Our results show that isotropy correlates with performance across diverse settings, capturing representational properties that error metrics overlook. This makes isotropy a necessary tool for assessing robustness and generalizability of transformer-based models.

6 Conclusion and Limitations

In this work, we introduced a novel approach to investigate the role of isotropy in transformer-based model 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 transformer-based 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 transformer-based models 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. In particular, promising directions include leveraging isotropy to guide fine-tuning strategies, inform inference-time decision-making (e.g., filtering low-quality predictions), and identify optimal representation depths in multi-layer transformers. Exploring these applications, alongside baseline performance comparisons, could translate our theoretical findings into practical tools for improving downstream performance. Ultimately, we believe that incorporating structural insights like isotropy into the transformer-based model design pipeline can significantly improve their reliability and adaptability to numerical domains.

References

Jacob Andreas and Dan Klein. When and why are log-linear models self-normalizing? In Rada Mihalcea, Joyce Chai, and Anoop Sarkar, editors, *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 244–249, Denver, Colorado, May–June 2015. Association for Computational Linguistics. doi: 10.3115/v1/N15-1027. URL https://aclanthology.org/N15-1027/.

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.

Jörn-Henrik Jacobsen, Jens Behrmann, Richard Zemel, and Matthias Bethge. Excessive invariance causes adversarial vulnerability, 2020. URL https://arxiv.org/abs/1811.00401.

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.

Anil Kumar Nelakanti, Cédric Archambeau, Julien Mairal, Francis Bach, and Guillaume Bouchard. Structured penalties for log-linear language models. In David Yarowsky, Timothy Baldwin, Anna Korhonen, Karen Livescu, and Steven Bethard, editors, *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 233–243, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://aclanthology.org/D13-1024/.

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.

Jochen Peters and Dietrich Klakow. Capturing long range correlations using log-linear language models. In *Verbmobil: Foundations of Speech-to-Speech Translation*, pages 79–94. Springer, 2000.

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.

Renan A Rojas-Gomez, Teck-Yian Lim, Alex Schwing, Minh Do, and Raymond A Yeh. Learnable polyphase sampling for shift invariant and equivariant convolutional networks. *Advances in Neural Information Processing Systems*, 35:35755–35768, 2022.

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.

Vasu Singla, Songwei Ge, Ronen Basri, and David Jacobs. Shift invariance can reduce adversarial robustness. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=tqi_45ApQzF.

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 trained model $\hat{p}(k_l \mid \mathbf{k}_{-l})$ matches that of ground-truth model $p^*(k_l \mid \mathbf{k}_{-l})$ and $\hat{f}(k,l) = 0$. In other words, there exists a trained 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 $\forall j \in \mathcal{V}$,

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

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

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 $\|\mathbf{J}\|_2 \le |\mathbf{\Lambda}|_2 \sum_{i=1}^{|\mathcal{V}|} \left(p_{i,i} + \frac{1}{2}\right) \left|\psi_i - \sum_{j=1}^{|\mathcal{V}|} p_{i,j}\psi_j\right|^2 + \Delta$.*

Proof. In Lemma 1, 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 $p_{i,j} = \frac{\exp(\psi_i^\top \mathbf{\Lambda} \psi_j)}{\sum_{k=1}^{|\mathcal{V}|} \exp(\psi_i^\top \mathbf{\Lambda} \psi_k)}$. 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

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

$\square$

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 $\sum_{i=1}^{|\mathcal{V}|} |\psi_i - \mathbf{\Psi}^\top \mathbf{\Psi} \mathbf{\Lambda} \psi_i|^2 = \operatorname{tr}\left((I - \mathbf{\Psi}^\top \mathbf{\Psi} \mathbf{\Lambda})^2 \mathbf{\Psi}^\top \mathbf{\Psi}\right)$. 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 $\sum_{i=1}^{|\mathcal{V}|} \|\psi_i - \mathbf{\Psi}^\top \mathbf{\Psi} \mathbf{\Lambda} \psi_i\|^2 = \operatorname{tr}\left(\left(I - \sum_{i=1}^m \gamma_i \gamma_i^\top\right)^2 \mathbf{\Psi}^\top \mathbf{\Psi}\right) = \sum_{q=m+1}^D \lambda_q$.

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 $\mathbf{\Psi}^\top \mathbf{\Psi}$. $\square$

D Clustering in the Contextual Embedding Space

Clustering. We begin with the isotropy assessment by performing clustering on the transformer-based model representations in the contextual embedding space. There are various methods for performing clustering, 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

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

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. One of our goals in this paper is to study how variations in time series characteristics affect isotropy and forecasting performance. For this, we selected real datasets from the Monash Time Series Forecasting Archive (https://forecastingdata.org/), a widely used benchmark covering diverse domains and structural properties. Additionally, we included a transportation dataset from a separate public source (https://github.com/phonism/llm4cp) to introduce greater variability.

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 $\mathbf{x}_{1:l_{\text{syn}}}$.

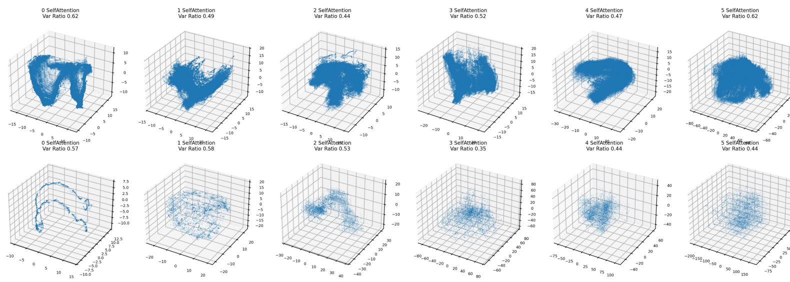
$j \sim \mathcal{U}\{1, J\}$ ▷ sample the number of kernels
$\{\kappa_1(t, t'), \ldots, \kappa_j(t, t')\} \overset{\text{i.i.d}}{\sim} \mathcal{K}$ ▷ sample j kernels from the Kernel bank $\mathcal{K}$
$\kappa^*(t, t') \leftarrow \kappa_1(t, t')$
for $i \leftarrow 2$ to j **do**
  $\star \sim \{+, \times\}$ ▷ 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

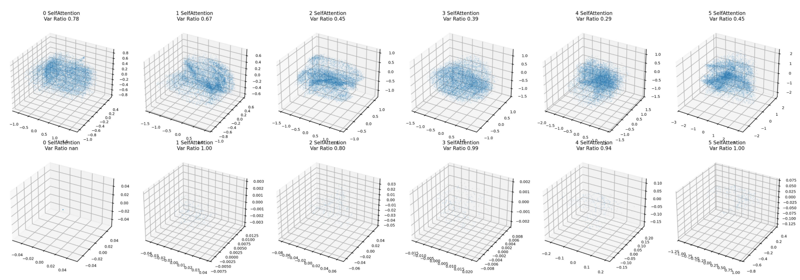
The full visualization of PCA plots of different models is provided below. We use the synthetic Dataset 1, and Dataset 2 from non-linear domain for illustration.

Non-Linear (Dataset 1):

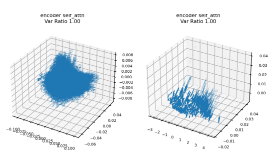
Chronos-T5



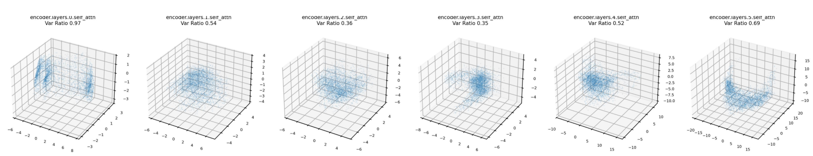
Chronos-Bolt



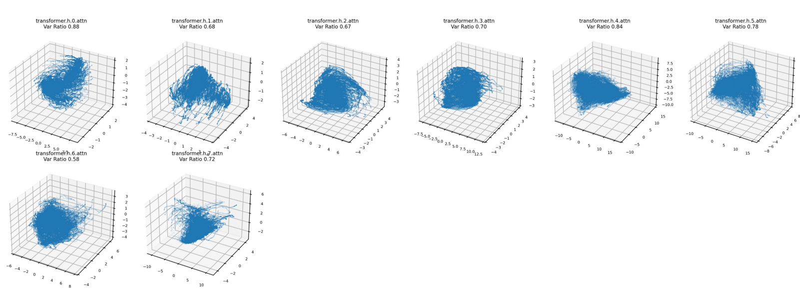
PatchTST



Morai

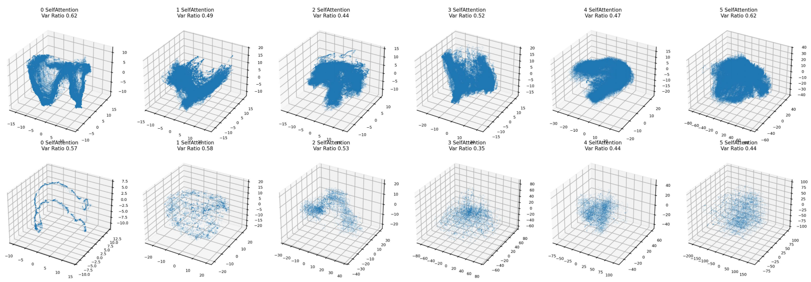


Lag-Llma

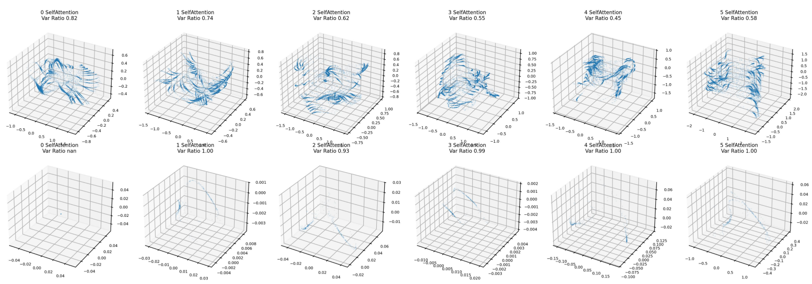


Non-Linear (Dataset 2):

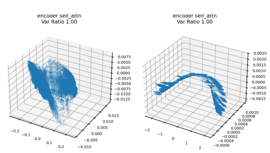
Chronos-T5



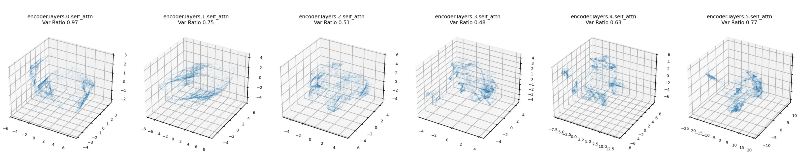
Chronos-Bolt



PatchTST



Morai



Lag-Llma