Is isotropy a good proxy for generalization in time series forecasting with transformers?

Rashed Shelim *rasheds@vt.edu*
Department of Electrical and Computer Engineering & Department of Computer Science
Virginia Tech

Shengzhe Xu *shengzx@vt.edu*
Department of Computer Science
Virginia Tech

Walid Saad *walids@vt.edu*
Department of Electrical and Computer Engineering
Virginia Tech

Naren Ramakrishnan *naren@cs.vt.edu*
Department of Computer Science
Virginia Tech

Reviewed on OpenReview: *https://openreview.net/forum?id=iUtDYVQzFq*

Abstract

Vector representations of contextual embeddings learned by transformer-based models have been shown to be effective even for downstream tasks in *numerical domains* such as time series forecasting. Their success in capturing long-range dependencies and contextual semantics has led to broad adoption across architectures. But at the same time, there is little theoretical understanding of when transformers, both autoregressive and non-autoregressive, generalize well to forecasting tasks. This paper addresses this gap through an analysis of isotropy in contextual embedding space. Specifically, we study a log-linear model as a simplified abstraction for studying hidden representations in transformer-based models. In this formulation, time series embeddings are mapped to predictive outputs through a softmax layer, providing a tractable lens for analyzing generalization. We show that state-of-the-art performance requires embeddings to possess a structure that accounts for the shift-invariance of the softmax function. By examining the gradient structure of self-attention, we demonstrate how isotropy preserves representation structure, resolves the shift-invariance problem, and provides insights into model reliability and generalization. Experiments across 22 different numerical datasets and 5 different transformer-based models show that data characteristics and architectural choices significantly affect isotropy, which in turn directly influences forecasting performance. This establishes isotropy as a theoretically grounded and empirically validated indicator of generalization and reliability in time series forecasting. The code for the isotropy analysis and all data are publicly available [1].

1 Introduction

Transformer-based models models have been proven effective across various downstream tasks in numerical domains, such as finance (Garza and Mergenthaler-Canseco (2023); Yu et al. (2023)), energy (Gao et al.

This work is supported in part by US National Science Foundation grants CNS-2225511, IIS-2509636, DBI-2412389, CMMI-2240402, IIS-2312794, and CCF-1918770. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the sponsor(s).

Corresponding author: Rashed Shelim (e-mail: rasheds@vt.edu).

[1] `https://github.com/ShengzheXu/stg_trust_indicator`

(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 across a wide variety of architectures. 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)) that extend or apply transformer-based models to numerical domains for 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 comprehend how models arrive at their predictions and raises fundamental challenges in assessing the reliability and generalization of model outputs in numerical domains.

Most scientific domains, e.g., finance, healthcare, rely on machine learning models to exhibit prediction reliability. Thus, in such domains, it is necessary to open up 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 embeddings learned by transformer-based models in various numerical downstream tasks, there is little theoretical understanding of their empirical success. This motivates the need for simplified yet expressive abstractions that enable the study of hidden representation geometry independent of specific architectural details. Thus, a fundamental question arises: *"When (or how) do the hidden representations of transformer-based models exhibit structural properties that enable reliable generalization in 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 reliable generalization 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 isotropy 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* to analyze the hidden representations of transformer-based models and demonstrate theoretically why such 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 contextual embeddings that are mapped to predictive outputs. While the precise output head may vary across architectures (e.g., regression or categorical distributions), we adopt a log-linear abstraction as a simplified model to study embedding geometry. 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 prediction problem on tokenized sequences. Together, these formulations establish the theoretical basis for studying the relationship between isotropy and forecasting performance.

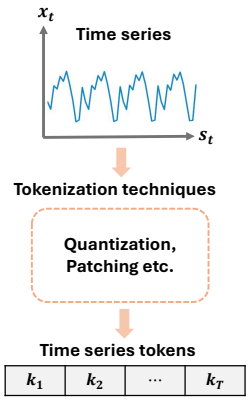


Figure 1: Time series tokenization.

Time Series Tokens and Similarity Measure. In transformer-based forecasting models, both autoregressive and non-autoregressive architectures can be viewed as learning predictive distributions over future values from historical context. 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 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 predictive distributions $p(k_l \,|\, \mathbf{k}_{-l})$ for $l \in \{1, \ldots, L\}$, where $\mathbf{k}_{-l}$ denotes the tokenized input sequence and k_i is a time series token in vocabulary $\mathcal{V}$.

Let $\tilde{\mathbf{\Psi}}_i = \{\psi_i^1, \psi_i^2, \ldots\}$ be the set of all contextual embedding instances produced by a transformer-based model for the time series token k_i. Here, different contexts in the time series sequences yield different self-attention-based embeddings of k_i. By constructing $\sum_{k_i \in \mathcal{V}} |\tilde{\mathbf{\Psi}}_i|$, 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 $\tilde{\mathbf{\Psi}}_i$. The expectation is taken over all pairs of different tokens. This metric quantifies how distinct or overlapping contextual embeddings are across tokens, thereby providing a measure of isotropy in the embedding space.

Model. We consider a general model for numerical data and open the black box of the transformer-based model. To analyze representation geometry, we adopt the log-linear model commonly used in prior theoretical work (Arora et al. (2016); Mu and Viswanath (2018); Andreas and Klein (2015); Peters and Klakow (2000); Nelakanti et al. (2013)). This adaptation is not intended to replicate the full transformer inference process, which also involves multi-head attention, feedforward layers, normalization, and residual connections. Instead, it serves as a tractable abstraction of the softmax parameterization commonly used in prediction heads. In this formulation, contextual embeddings (generated through self-attention and subsequent transformations) interact with token embeddings via inner products, and probabilities are obtained through normalization. This provides a mathematical framework for studying isotropy in contextual embedding spaces. Formally, we define the conditional distribution of a future time series token k_l given a tokenized input sequence $\mathbf{k}_{-l}$ using a softmax-parameterized log-linear form, typical of many prediction heads:

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

where $\psi^*_t \in \mathbb{R}^D$ is a token embedding corresponding to the observation x_t, and $\psi^*_{-l}(\mathbf{k}_{-l})$ encodes the input sequence $\mathbf{k}_{-l}$ into a contextual representation in $\mathbb{R}^D$. The notation $\langle \cdot, \cdot \rangle$ denotes the inner product between context and token embeddings. 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. In this abstraction, the partition function normalizes the output probabilities. Then, the normalized ground-truth model $\forall i \in \mathcal{V}$ is then

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

Since we do not know the ground-truth model in reality, we do not have access to the ground-truth components ψ^*_t and $\psi^*_{-l}(\mathbf{k}_{-l})$. Instead, we only have access to the trained model embeddings ψ_t and $\psi_{-l}(\mathbf{k}_{-l})$ that aim to minimize training loss. We can then define the trained model's 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 trained transformer-based model during training. The downstream task is to learn a function $f(k,l)$. Finally, the output distribution of the trained model $\forall i \in \mathcal{V}$ is

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

Loss Function. While transformer-based forecasting models in practice are trained with a variety of objectives (e.g., mean squared error, likelihood-based losses, or cross-entropy), for theoretical analysis we adopt a cross-entropy formulation. This abstraction allows us to express the training objective compactly in expectation form and to connect it to KL divergence, thereby providing a tractable framework for studying how embedding representations relate to generalization and reliability. For a tokenized input sequence $\mathbf{k}_{-l}$, the loss is given by:

$$\begin{aligned} \mathcal{L} &= \mathbb{E}_k \Big[-p^*(k_l{=}x_t \,|\, \mathbf{k}_{-l}) \log p(k_l{=}x_t \,|\, \mathbf{k}_{-l}) \Big] \\ &= \mathbb{E}_k \Big[\mathcal{D}_{\mathrm{KL}} \Big(p^*(k_l \,|\, \mathbf{k}_{-l}) \,\|\, p(k_l \,|\, \mathbf{k}_{-l}) \Big) \Big] + \mathbb{E}_k \Big[\mathcal{H}\big(p^*(k_l \,|\, \mathbf{k}_{-l})\big) \Big], \end{aligned} \tag{4}$$

where $p(\cdot|\mathbf{k}_{-l})$ is the categorical distribution predicted by the trained model, $p^*(\cdot|\mathbf{k}_{-l})$ is the ground-truth distribution, $\mathcal{D}_{\mathrm{KL}}$ is the KL divergence, and $\mathcal{H}(\cdot)$ denotes entropy. This formulation highlights that minimizing cross-entropy is equivalent to minimizing KL divergence up to an additive constant given by the ground-truth entropy. Under large-scale training, the cross-entropy loss becomes small, implying that the KL divergence term is also small. This supports the common assumption that the trained model distribution approximates the ground-truth distribution in practice.

Downstream Numerical Task. We formulate time series forecasting as a downstream task, in which a trained transformer-based model is used to predict future values of a time series. While practical models often generate continuous outputs (e.g., through regression or likelihood-based heads), for theoretical analysis we adopt a categorical abstraction. Specifically, the continuous observations are mapped into a finite vocabulary of discrete tokens, so that forecasting can be cast as a conditional prediction problem over $\mathcal{V}$. The downstream task is then defined as a function of the logits $\mathbf{z}^*(k,l)$ generated by the ground truth model, where each logit corresponds to a token in $\mathcal{V}$ given the input sequence $\mathbf{k}_{-l}$. For interpretability, one can consider a linear predictor that operates directly on the contextual representation $\psi^*_{-l}(\mathbf{k}_{-l})$: $f^*(k,l) = \langle \psi^*_{-l}(\mathbf{k}_{-l}), u^* \rangle = \sum_{i=1}^{|\mathcal{V}|} a^*_i z^*_i(k,l)$, where $u^* = \sum_{i=1}^{|\mathcal{V}|} a^*_i \psi^*_t \in \mathbb{R}^D$ and a^*_i are coefficients. Although this linear form provides intuition about how logits contribute to predictions, it does not guarantee reliable generalization, as KL divergence is less sensitive to sign changes in $f^*(k,l)$ when logits have small magnitude Wu et al. (2023). To improve robustness, we instead model the downstream task using a nonlinear activation on the logits:

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

where σ is the ReLU function and b_i^* is a threshold parameter. This abstraction suppresses low-confidence logits that contribute minimally to KL divergence, thereby focusing the task on high-relevance predictions. While not intended as a literal description of forecasting architectures, this formulation provides a theoretical device for analyzing robustness and highlights how isotropy in the embedding space supports reliable generalization in time series forecasting.

Despite the empirical success of transformer-based approaches in numerical domains, there remains a fundamental gap in understanding when these models generalize reliably. A key challenge lies in the mismatch between training and inference performance, i.e., *good training performance does not necessarily translate to robustness at test time*. To address this, we introduce a theoretical framework based on isotropy in the contextual embedding space. Within this framework, strong isotropy in hidden representations stabilizes the partition function and mitigates the shift-invariance issue observed in softmax-based abstractions, thereby leading to more reliable inference. The next section formalizes this insight and provides a theoretical justification for using isotropy as a key indicator of model reliability in time series forecasting.

3 The Role of Isotropy in Transformer-Based Models for Time Series Forecasting

Overview of Section 3. This section develops a theoretical perspective on why isotropy in contextual embeddings is critical for reliable generalization of transformer-based models in numerical forecasting. Although strong performance is often observed during training, it does not always transfer to unseen scenarios, particularly across heterogeneous numerical domains. To explain this mismatch, we introduce a framework that links isotropy of hidden representations to the stability of the partition function. We show that isotropy mitigates the shift-invariance issue in softmax-based abstractions, thereby enabling more robust generalization. This establishes a formal connection between isotropy and inference reliability, motivating its use as an indicator of model performance in numerical domains.

As discussed in Section 2, we consider transformer-based forecasting models where outputs are often parameterized through task-specific heads (e.g., regression, likelihood, or classification). For theoretical analysis, we adopt a softmax-based formulation as an abstraction to connect logits and probabilities, enabling isotropy analysis. The relationship between logits and this softmax-based abstraction directly impacts downstream forecasting performance. However, the softmax function is inherently shift-invariant, i.e., its output remains unchanged if all logits are shifted by the same 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 time series forecasting, this implies that unless hidden representations enforce structural stability, outputs under the softmax-based abstraction may remain unaffected by meaningful representational changes. Since we cannot control the logit shifts of a trained model on unseen data, strong predictive performance during training does not necessarily imply reliable generalization in out-of-distribution scenarios. This lack of reliability due to uncontrolled logit shifts is 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 $\hat{p}\left(k_l \,|\, \mathbf{k}_{-l}\right)$ matches that of the ground-truth model $p^*(k_l \,|\, \mathbf{k}_{-l})$ while $\hat{f}(k,l) = 0$. In other words, there exists a trained model with the same training loss as the ground-truth model, but with logits ineffective for numerical downstream tasks.*

Proof. The proof is provided in Appendix A. □

Theorem 1 highlights that without additional structure in the hidden representations of transformer-based models, the logits can be arbitrarily shifted while preserving training loss, rendering them ineffective for forecasting tasks. Thus, theoretical guarantees for downstream performance require embedding spaces that enforce geometric stability.

To prevent the shift-invariance problem from undermining forecasting reliability, it is necessary to stabilize the partition function. Let $\mathbf{\Psi} = (\psi_1, \ldots, \psi_{|\mathcal{V}|})^\top \in \mathbb{R}^{|\mathcal{V}| \times D}$ denote the contextual embeddings of the tokenized

time series. Within our softmax-based abstraction, the stability of the partition function can be assessed through the isotropy of the embedding space (Arora et al. (2016); Mu and Viswanath (2018)):

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

where $\mathcal{C} = \mathbf{\Psi}^\top \mathbf{\Psi}$ is the input correlation matrix of input patterns 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 closer to isotropic and thus geometrically stable (Arora et al. (2016); Mu and Viswanath (2018)). In this abstraction, probabilities depend consistently on logits without being distorted by global shifts, thereby mitigating the softmax shift-invariance problem.

Building on this theoretical foundation, we now turn to the following empirical question: *"How can isotropy be quantified and interpreted in practice, and how does it relate to forecasting generalization across numerical domains?"* Motivated by Theorem 1, we analyze structural properties of hidden representations, such as effective dimensionality and cluster organization, in transformer-based models under our softmax-based abstraction. These analyses reveal how isotropy manifests in contextual embeddings and how its presence correlates with generalization ability across diverse numerical domains, thereby providing a practical indicator of model reliability. Section 4 introduces quantitative methods, including spectral alignment, principal component analysis (PCA), and cluster-based isotropy metrics, which link theoretical reliability to empirical generalization.

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, 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[2], 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

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

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				

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

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., $g_i(\mathbf{\Psi}) = \text{softmax}(\mathbf{\Psi}\mathbf{\Lambda}\mathbf{\Psi}^\top)\mathbf{\Psi}$, 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 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* $\mathbf{J}$[3] *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 $\mathbf{\Lambda}$.

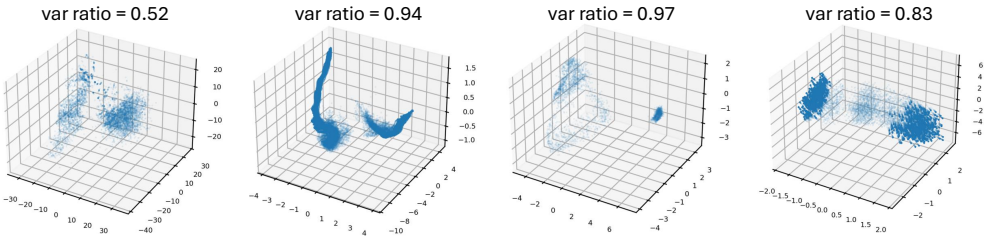
Theorem 2. *Let the eigenvalues of the correlation matrix* $\mathbf{\Psi}^\top\mathbf{\Psi}$ *be ordered as* $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_D$*, and let* $\gamma_i \in \mathbb{R}^D$ *for* $i = 1, \ldots, D$ *denote their associated eigenvectors. Then, the matrix* $\mathbf{\Lambda}^*$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

[3]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 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.

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 through a combination of controlled and broad-based experiments. We begin with a *matching controlled experiment* that implements a simple transformer-based model with a softmax output head trained under cross-entropy loss which reflects the assumptions of our theoretical framework in Section 2. This serves as a sanity check that bridges theory and practice, confirming that isotropy is tightly linked to forecasting reliability when the model setup matches the analytical formulation. We then extend the evaluation to diverse transformer-based models and datasets, analyzing both synthetic (qualitative) and real-world (quantitative) domains. These broader experiments investigate how factors such as model architecture, tokenization strategies, context length, and noise influence the isotropic structure of learned representations. Across all settings, we find that higher isotropy consistently correlates with better generalization, reinforcing its utility as a reliable, label-free diagnostic for the robustness of transformer-based models in time series forecasting.

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. We take the released weights of of the baselines and apply them directly to our experimental datasets without fine-tuning. This setup allows us to study the generalization of transformer-based models in time series forecasting, since models trained once are evaluated on unseen time series datasets. In this context, the isotropy of embeddings reflects how well transformer-based models generalize their forecasting capability to new numerical domains.

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	$L = 32$

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

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)

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 Matching Controlled Experiment

To bridge our theoretical framework in Section 2 with the empirical evaluation in Section 5.2 and Section 5.3, in this section, we conduct a matching controlled experiment using a simple transformer-based time series forecasting model closely aligned with our problem formulation. Specifically, we adapt an open-source vanilla transformer (Vaswani et al. (2023)) implementation for time series forecasting by modifying its output layer and loss function to match our theoretical assumptions. This controlled setup allows us to evaluate isotropy under conditions that directly reflect the log-linear abstraction and KL divergence formulation introduced in Section 2.

Model Architecture. The controlled model consists of two standard transformer encoder layers, each comprising multi-head self-attention with 10 heads, residual connections, layer normalization, and a feed-forward network. The architectural details are chosen to align with our problem formulation in Section 2. We introduced the following adaptations:

- **Input tokenization:** Raw numerical time series data is discretized via uniform quantization into $|\mathcal{V}| = 512$ bins. Each quantized value represents a time series token, forming a categorical vocabulary consistent with the log-linear model in Section 2. These tokens are then embedded into 250-dimensional vectors, and sinusoidal positional encodings are added to preserve temporal order. This transformation converts the continuous sequence of values into a sequence of categorical tokens suitable for the transformer, thereby ensuring that the log-linear abstraction in equation 2 and KL-divergence formulation in equation 4 is realized in practice.
- **Transformer encoder:** A two-layer transformer encoder ($n_l = 2$), where each layer contains multi-head self-attention with 10 heads ($n_H = 10$) and a feed-forward block ($d_h = 250$). Softmax normalization is applied within each attention layer, matching the theoretical derivation in Section 2.
- **Output projection:** The contextual embeddings are mapped to the vocabulary size $|\mathcal{V}| = 512$ through a linear transformation, followed by a softmax layer to yield predictive distributions over quantized values.
- **De-tokenization:** At inference, predicted token distributions are mapped back to real values by taking the expectation over the quantization bins, enabling comparison with the ground-truth numerical series.

Training Setup. The model was trained for 200 epochs with batch size 250 using the AdamW optimizer (learning rate 5×10^{-4}). The loss function was categorical cross-entropy, which directly minimizes the KL divergence between predicted and ground-truth token distributions, as in equation 4. Input sequences of length $L = 128$ are used with a forecast horizon of 1 step. Here, we train a lightweight vanilla transformer from scratch on synthetic datasets generated from Gaussian process kernels (linear, trend, seasonality, non-linear, and stochastic; see Algorithm E in Appendix E). For each domain, a different dataset from Table 3 is used to evaluate the controlled model. This setup not only realizes the log-linear and KL divergence assumptions in

practice but also tests whether isotropy remains a reliable proxy for generalization when transferring across datasets within and across domains. To ensure robustness, each experiment was repeated across 6 random seeds.

Results. In Figure 3, we compare the NMSE vs isotropy across different synthetic datasets to evaluate the controlled model. For instance, in the linear-Dataset 2, strong isotropy (inter-token cosine similarity in equation 6 near 0, i.e., 0.0000023) corresponds to NMSE of 9.6×10^{-6}, while in the stochastic-Dataset 1, a weaker isotropy (inter-token cosine similarity far from 0, i.e., 0.12) corresponds to NMSE of 0.038. Similar patterns can be observed across all synthetic domains, where models with higher isotropy consistently stabilized the partition function and generalized better to unseen datasets. These results in Figure 3, therefore, validate our theoretical justification by showing that the isotropy of the contextual embedding space correlates strongly with time series forecasting performance.

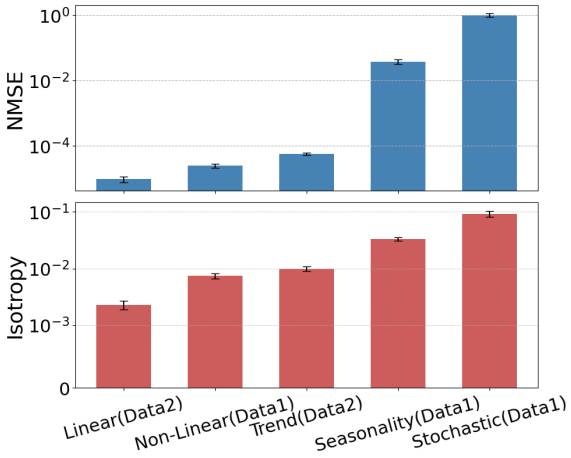


Figure 3: NMSE vs isotropy comparison across different synthetic datasets for the controlled model. All reported results are averaged over 6 independent trials, with error bars indicating the standard deviation across runs to capture variability.

Implications. The matching controlled experiment in this section validates our theoretical framework in Section 2 under assumptions precisely aligned with softmax outputs of self-attention and KL divergence in the loss function. It demonstrates that isotropy serves as a reliable proxy for generalization in practice, extending beyond the abstract log-linear formulation. Importantly, isotropy is computed solely from the embedding geometry of the input sequences, independent of ground-truth outputs. In this sense, the diagnostic does not require labels and can be applied in scenarios where only input data are available. In our study, outputs are used exclusively for validation, i.e., to empirically correlate isotropy with forecasting error through NMSE. Thus, these experiments already provide indirect empirical evidence that isotropy can function as a label-free reliability signal in time series forecasting tasks where outputs may not be accessible. Building on this foundation, the following sections extend the analysis to diverse transformer-based architectures and datasets, demonstrating the broader applicability of isotropy beyond the controlled setting.

5.2 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.

We begin by analyzing time series forecasting performance (i.e., NMSE) for different baselines and its relation with isotropy in Figure 4. For instance, in Figure 4 b, we have (NMSE = 0.0000066 and cosine similarity = $|-0.00076|$[4]) 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

[4]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.

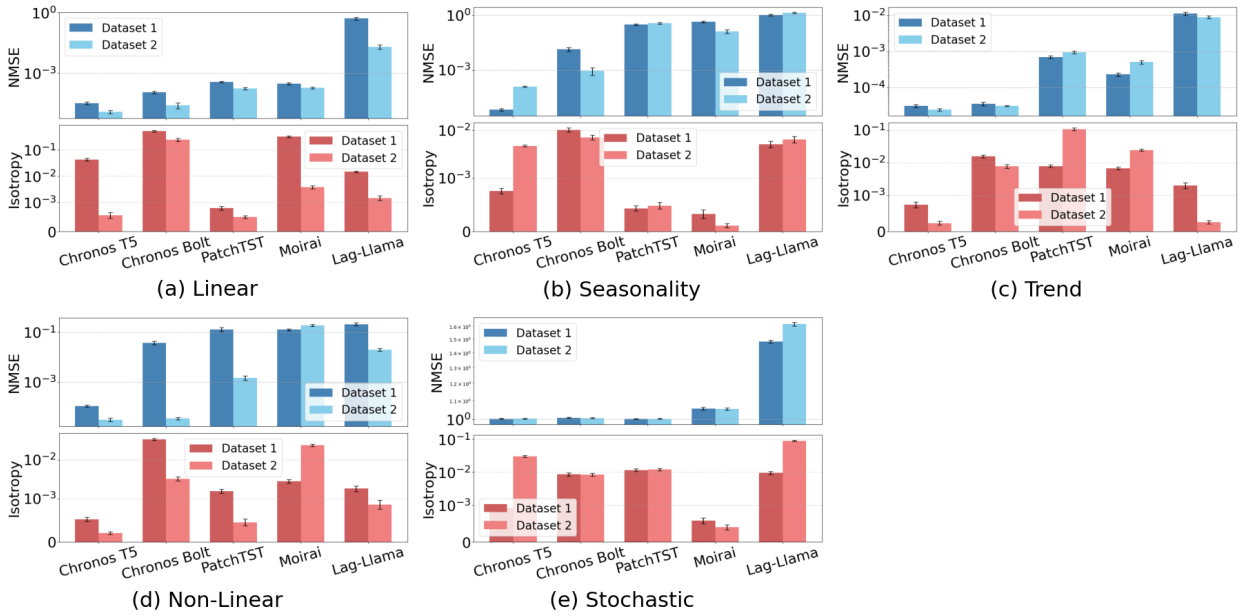


Figure 4: NMSE vs isotropy analysis for 10 different synthetic datasets of 5 different domains. All reported results are averaged over 6 independent trials, with error bars indicating the standard deviation across runs to capture variability.

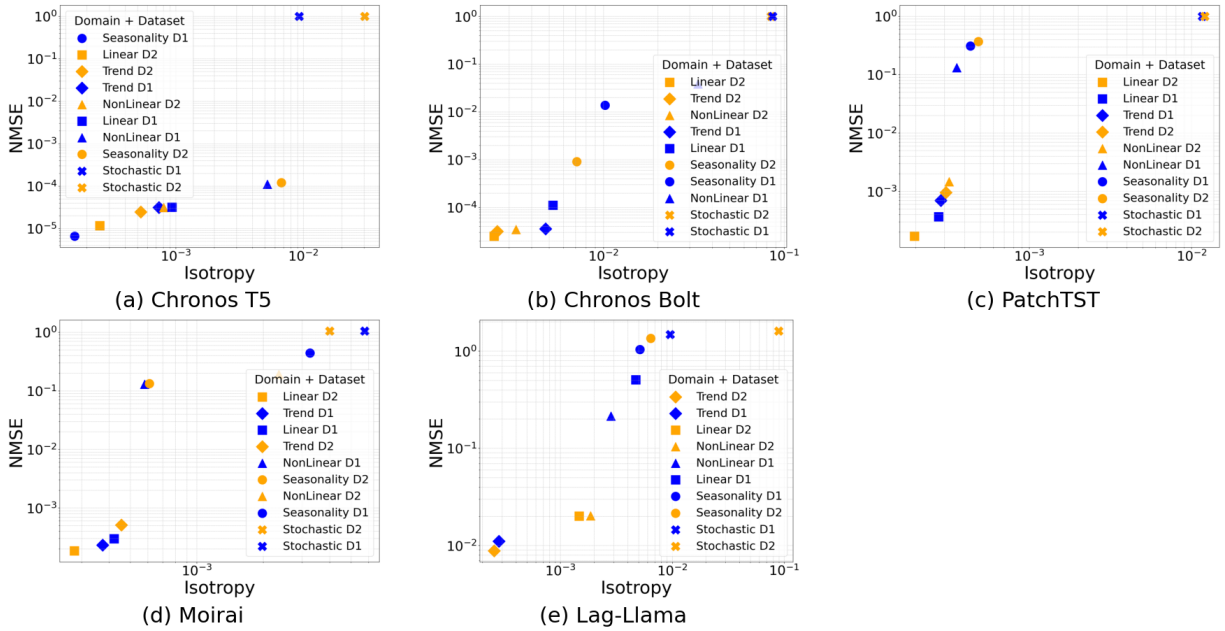


Figure 5: NMSE vs isotropy across numerical domains. The relationship between NMSE and isotropy is consistent across domains, highlighting that stronger isotropy (i.e., inter-token cosine similarity in equation 6 close to 0) leads to lower NMSE and vice versa.

cosine similarity can also vary across different transformer-based models and datasets. For example, in Figure 4c, 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 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 4 b, 4 d, and 4 e. This shows that any dataset from 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. We note that in some cases (e.g., Chronos Bolt in Figures 4d), isotropy exhibits higher variability than NMSE. This arises because isotropy captures sensitivity in the embedding geometry, which can fluctuate across runs, whereas the prediction head can compensate to keep forecasting error relatively stable. Such discrepancies highlight the complementary role of isotropy as a diagnostic tool beyond error metrics. In Figure 5, we show NMSE vs isotropy comparison across numerical domains. From

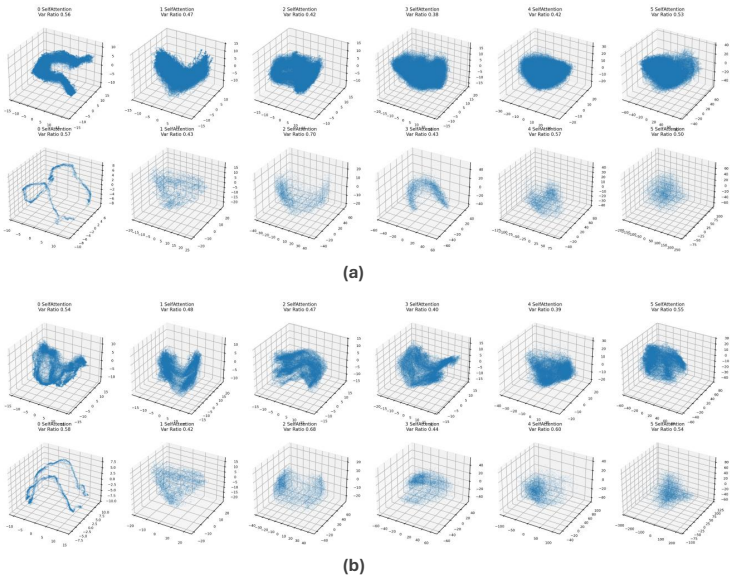


Figure 6: Variations in Chronos-T5's hidden representations for different input context lengths for the same synthetic dataset non-linear-Dataset 1 : (a) Contextual embedding space for input context length $L = 500$. (b) Contextual embedding space for input context length $L = 100$.

the figure, it can be seen that a consistent relationship exists between NMSE and isotropy, which justifies our findings, i.e., *stronger isotropy leads to lower NMSE and vice versa.*

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 6 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 7, 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 7 b, 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 Figure 7 a, we have (NMSE= 0.000025, cosine similarity= 0.2474) and (NMSE= 0.000009, cosine similarity= 0.0644) for $L = 500$ and $L = 100$, respectively, for Chronos-T5 with linear-Dataset 2. 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

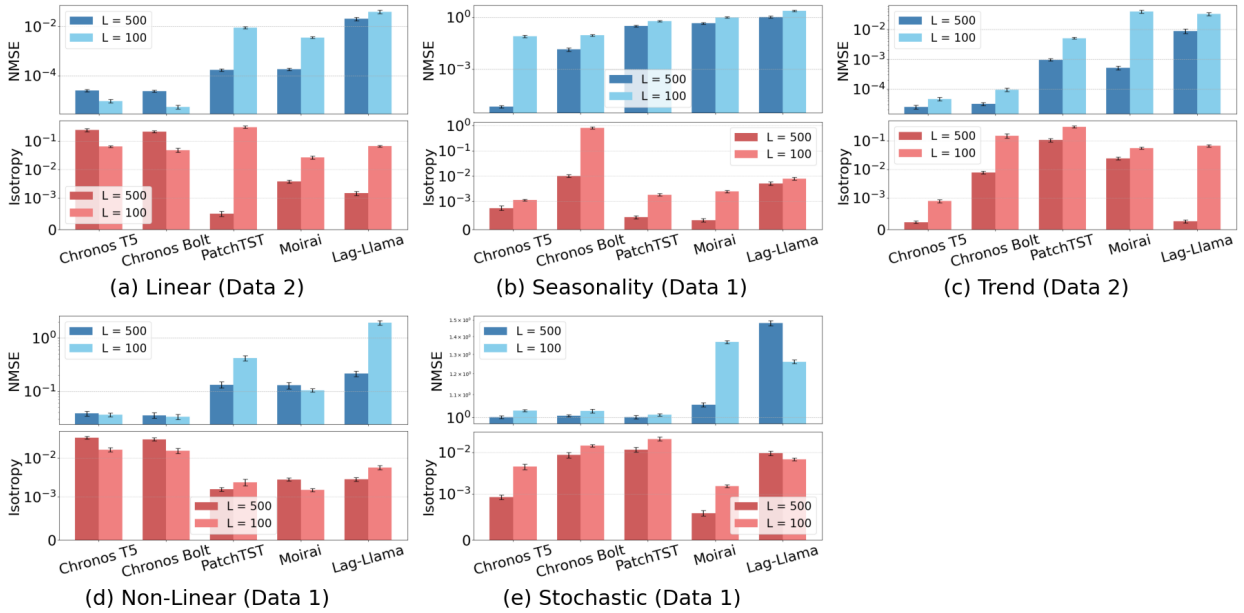


Figure 7: NMSE vs isotropy comparison across different input context lengths for synthetic datasets. All reported results are averaged over 6 independent trials, with error bars indicating the standard deviation across runs to capture variability.

also be observed for other synthetic datasets and baselines in Figures 7 c, 7 d, and 7 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.

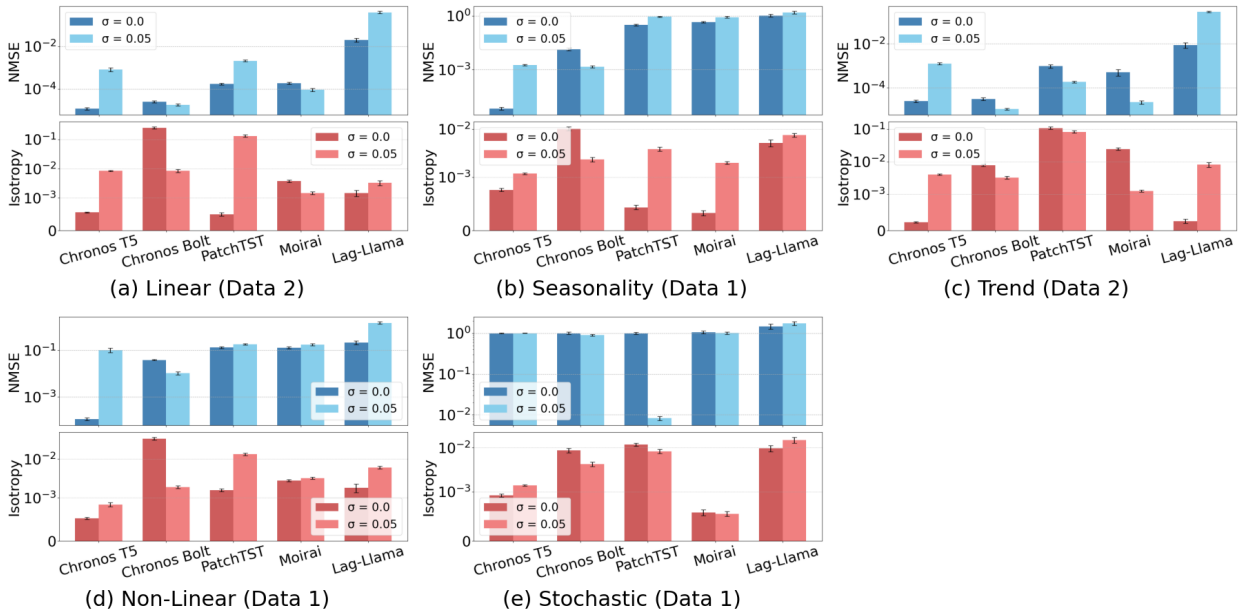


Figure 8: NMSE vs isotropy comparison across different noise levels in synthetic datasets. All reported results are averaged over 6 independent trials, with error bars indicating the standard deviation across runs to capture variability.

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 8 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. For instance, in Figure 8 c, 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 8 a, 8 b, 8 d, and 8 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.

5.3 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 5.2, 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)

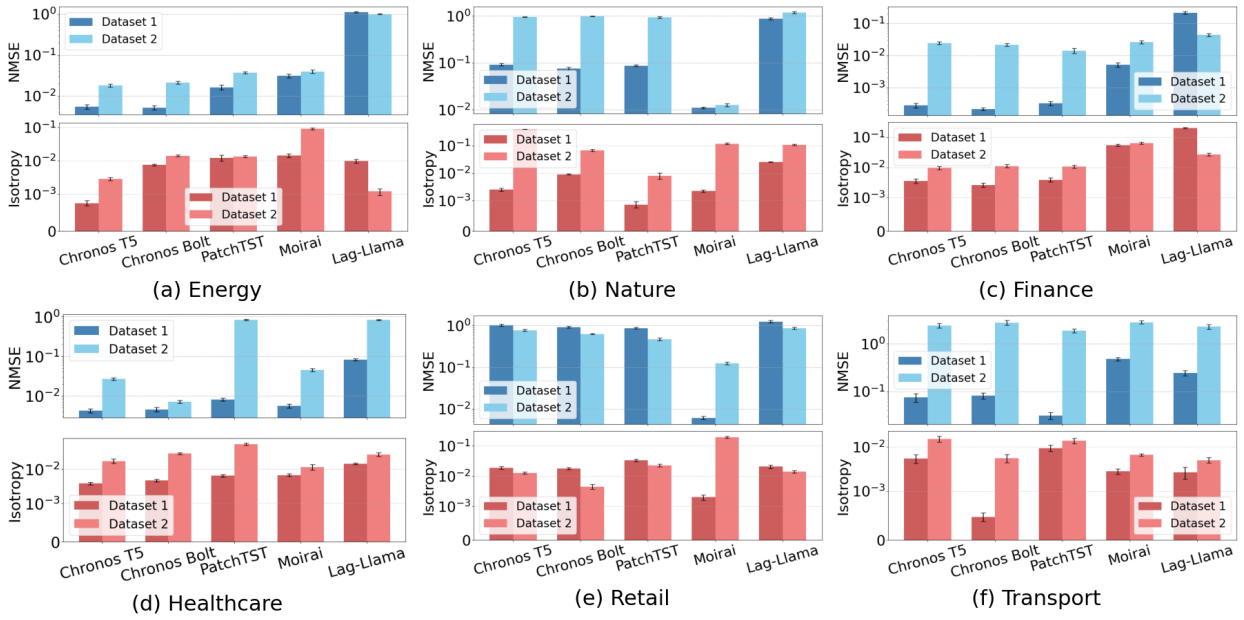


Figure 9: NMSE vs isotropy analysis for 12 different real datasets of 6 different domains. All reported results are averaged over 6 independent trials, with error bars indicating the standard deviation across runs to capture variability.

In Figure 9, we analyze the time series forecasting performance of different baselines and its relation with isotropy for different real datasets. For instance, in Figure 9 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

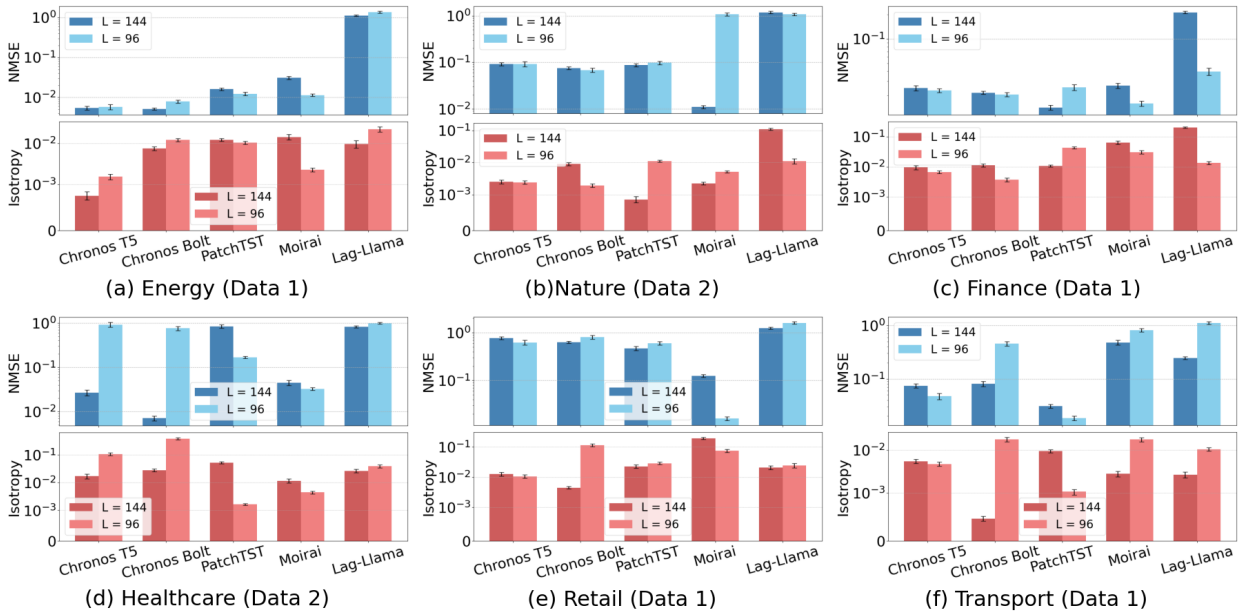


Figure 10: NMSE vs isotropy comparison across different input context lengths for real datasets. All reported results are averaged over 6 independent trials, with error bars indicating the standard deviation across runs to capture variability.

datasets and transformer-based models. For example, in Figure 9a, 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 9 c and 9 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.

Finally, in Figure 10, 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, 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 (Figure 10 a) and from 0.0026 to 0.0103 for transport-Dataset 1 (Figure 10 f), 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 (Figure 10 b) and from 0.2014 to 0.0133 for finance-Dataset 1 (Figure 10 c), which in turn improves the NMSE performances. A similar analysis can also be observed for other real datasets and baselines in Figures 10 a, 10 b, 10 c, 10 d, 10 e, and 10 f. 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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

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 $\hat{p}\left(k_l \,|\, \mathbf{k}_{-l}\right)$ matches that of the ground-truth model $p^*(k_l \,|\, \mathbf{k}_{-l})$ while $\hat{f}(k,l) = 0$.* In other words, there exists a trained model with the same training loss as the ground-truth model, but with logits ineffective for 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 $\|\mathbf{J}\|_2 \leq |\mathbf{\Lambda}|_2 \sum_{i=1}^{|\mathcal{V}|} \left(p_{i,i} + \frac{1}{2}\right) \left|\psi_i - \sum_{j=1}^{|\mathcal{V}|} p_{i,j}\psi_j\right|^2 + \Delta$.*

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

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

□

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 $\sum_{i=1}^{|\mathcal{V}|} |\psi_i - \mathbf{\Psi}^\top \mathbf{\Psi} \mathbf{\Lambda} \psi_i|^2 = \operatorname{tr}\left((I - \mathbf{\Psi}^\top \mathbf{\Psi} \mathbf{\Lambda})^2 \mathbf{\Psi}^\top \mathbf{\Psi}\right)$. 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 $\mathbf{\Psi}^\top \mathbf{\Psi}$. □

D Clustering in the Contextual Embedding Space

Clustering. We begin with the isotropy assesmment 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

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

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

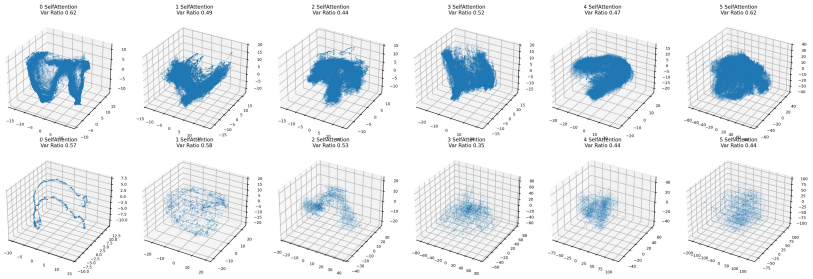
$j \sim \mathcal{U}\{1, J\}$ ▷ sample the number of kernels
$\{\kappa_1(t,t'), \ldots, \kappa_j(t,t')\} \overset{\text{i.i.d}}{\sim} \mathcal{K}$ ▷ sample j kernels from the Kernel bank $\mathcal{K}$
$\kappa^*(t,t') \leftarrow \kappa_1(t,t')$
for $i \leftarrow 2$ to j **do**
    $\star \sim \{+, \times\}$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

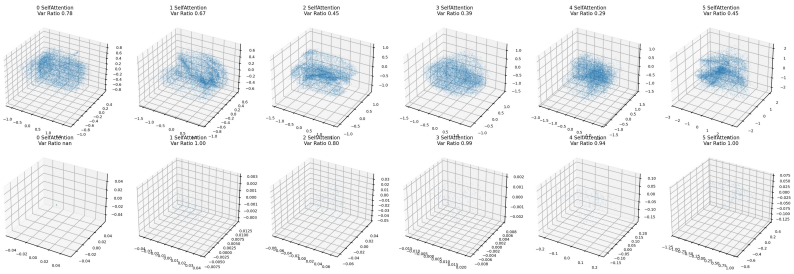
The full visualization of PCA plots of different models is provided below. We use the synthetic Dataset 1, and Dataset 2 from non-linear domain for illustration.

Non-Linear (Dataset 1):

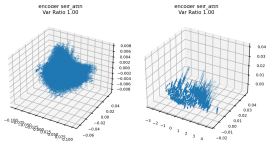
Chronos-T5



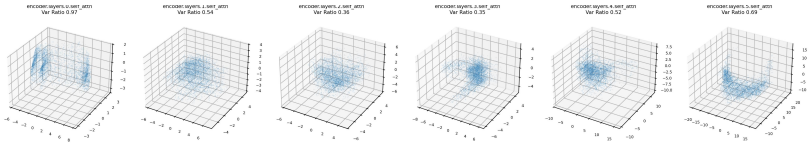
Chronos-Bolt



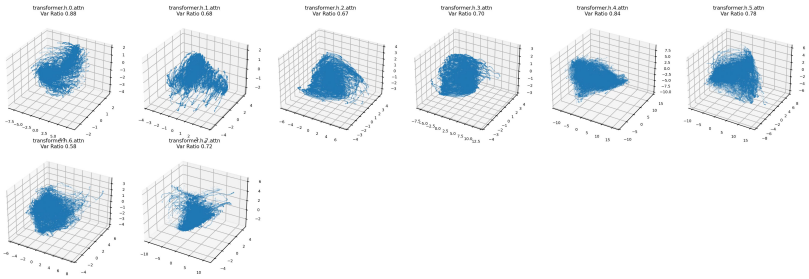
PatchTST



Morai

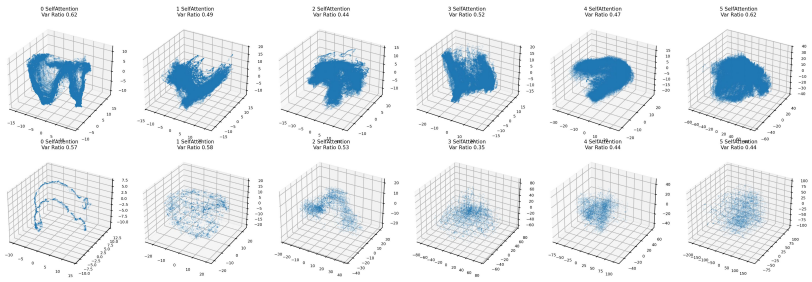


Lag-Llma

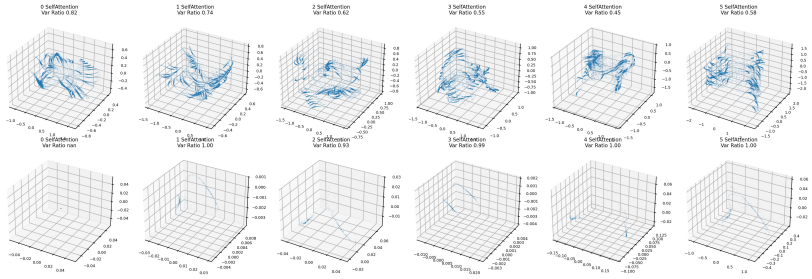


Non-Linear (Dataset 2):

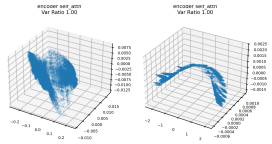
Chronos-T5



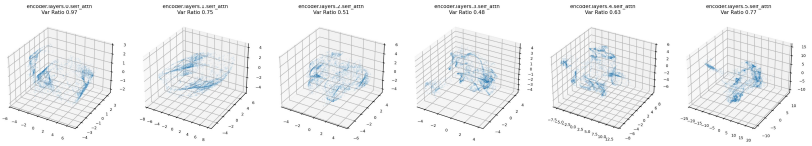
Chronos-Bolt



PatchTST



Morai



Lag-Llma