Geometric Analysis of Transformer Time Series Forecasting Latent Manifolds

Anonymous authors
Paper under double-blind review

Abstract

Transformer models have consistently achieved remarkable results in various domains such as natural language processing and computer vision. However, despite ongoing research efforts to better understand these models, the field still lacks a comprehensive understanding. This is particularly true for deep time series forecasting methods, where analysis and understanding work is relatively limited. Time series data, unlike image and text information, can be more challenging to interpret and analyze. To address this, we approach the problem from a *manifold learning* perspective, assuming that the latent representations of time series forecasting models lie next to a low-dimensional manifold. In our study, we focus on analyzing the geometric features of these latent data manifolds, including intrinsic dimension and principal curvatures. Our findings reveal that deep transformer models exhibit similar geometric behavior across layers, and these geometric features are correlated with model performance. Additionally, we observe that untrained models initially have different structures, but they rapidly converge during training. By leveraging our geometric analysis and differentiable tools, we can potentially design new and improved deep forecasting neural networks. This approach complements existing analysis studies and contributes to a better understanding of transformer models in the context of time series forecasting.

1 Introduction

Over the past decade, modern deep learning has shown remarkable results on multiple challenging tasks in computer vision (Krizhevsky et al., 2012), natural language processing (NLP) (Pennington et al., 2014), and speech recognition (Graves et al., 2013), among other domains (Goodfellow et al., 2016). Recently, the transformer (Vaswani et al., 2017) has revolutionized NLP by allowing neural networks to capture long-range dependencies and contextual information effectively. In addition, transformer-based architectures have been extended to non-NLP fields, and they are among the state-of-the-art (SOTA) models for vision (Dosovitskiy et al., 2020) as well as time series forecasting (TSF) (Wu et al., 2021; Zhou et al., 2022). Unfortunately, while previous works, e.g., Zeiler & Fergus (2014); Karpathy et al. (2015); Tsai et al. (2019) among others, attempted to explain the underlying mechanisms of neural networks, deep transformer models are still considered not well understood.

The majority of approaches analyzing the inner workings of vision and NLP transformer models investigate their attention modules (Bahdanau et al., 2015) and salient inputs (Wallace et al., 2019). Unfortunately, time series forecasting methods have received significantly less attention. This may be in part due to their relatively recent appearance as strong contenders on TSF in comparison to non-deep and hybrid techniques (Oreshkin et al., 2020). Further, while vision and NLP modalities may be "natural" to interpret and analyze, time series data requires analysis tools which may be challenging to develop for deep models. For instance, N-BEATS (Oreshkin et al., 2020) designed a method that promotes the learning of trend and seasonality parts, however, their model often recovers latent variables whose relation to trend and seasonality is unclear (Challu et al., 2022). Moreover, there is already a significant body of work of SOTA TSF that warrants analysis and understanding. Toward bridging this gap, we investigate in this work the geometric properties of latent representations of transformer-based TSF techniques via *manifold learning* tools.

Manifold learning is the study of complex data encodings under the *manifold hypothesis* where high-dimensional data is assumed to lie close to a low-dimensional manifold (Coifman & Lafon, 2006). This assumption is underlying the development of numerous machine learning techniques, akin to considering independent and identically distributed (i.i.d.) samples (Goodfellow et al., 2016). Recent examples include works on vision (Nguyen et al., 2019), NLP (Hashimoto et al., 2016), and time series forecasting (Papaioannou et al., 2022). However, to the best of our knowledge, there is no systematic work that analyzes transformer-based TSF deep neural networks from a manifold learning perspective. In what follows, we advocate the study of geometric features of Riemannian manifolds (Lee, 2006) including their *intrinsic dimension* (ID) and *mean absolute principal curvature* (MAPC). The ID is the minimal degrees of freedom needed for a lossless encoding of the data, and MAPC measures the deviation of a manifold from being flat.

Previously, geometric features of data manifolds were considered in the context of analyzing deep convolutional neural networks (CNN) (Ansuini et al., 2019; Kaufman & Azencot, 2023). Motivated by these recent works, we extend their analysis on image classification to the time series forecasting setting, focusing on SOTA TSF models (Wu et al., 2021; Zhou et al., 2022) evaluated on several multivariate time series datasets. We aim at characterizing the dimension and curvature profiles of latent representations along layers of deep transformer models. Our study addresses the following questions: (i) how do dimensionality and curvature change across layers? are the resulting profiles similar for different architectures and datasets? (ii) is there a correlation between geometric features of the data manifold to the performance of the model? (iii) how do untrained manifolds differ from trained ones? how do manifolds evolve during training?

Our results show that manifolds of transformer forecasting models undergo two phases: during encoding, curvature and dimensionality either drop or stay fixed, and then, during the decoding part, both dimensionality and curvature increase significantly. Further, this behavior is shared across several architectures, datasets and forecast horizons. In addition, we find that the MAPC estimate is correlated with the test mean squared error, allowing one to compare models without access to the test set. Moreover, this correlation is unlike the one found in deep neural networks for classification. Finally, untrained models show somewhat random dimension and curvature patterns, and moreover, geometric manifolds converge rapidly (within a few epochs) to their final geometric profiles. This finding may be related to studies on the neural tangent kernel (Li & Liang, 2018; Jacot et al., 2018) and linear models for forecasting (Zeng et al., 2023). We believe that our geometric insights, results and tools may be used to design new and improved deep forecasting tools.

2 Related Work

Our research lies at the intersection of understanding deep transformer-based models, and manifold learning for analysis and time series. We focus our discussion on these topics.

Analysis of transformers. Large transformer models have impacted the field of NLP and have led to works such as Vig (2019) that analyze the multi-head attention patterns and found that specific attention heads can be associated with various grammatical functions, such as co-reference and noun modifiers. Several works (Clark et al., 2019; Tenney et al., 2019; Rogers et al., 2021) study the BERT model (Devlin et al., 2019) and show that lower layers handle lexical and syntactic information such as part of speech, while the upper layers handle increasingly complex information such as semantic roles and co-reference. In Dosovitskiy et al. (2020), the authors inspect patch-based vision transformers (ViT) and find that the models globally attend to image regions that are semantically relevant for classification. Caron et al. (2021) show that a self-supervised trained ViT produces explicit representations of the semantic location of objects within natural images. Chefer et al. (2021) compute a relevancy score for self-attention layers that is propagated throughout the network, yielding a visualization that highlights class-specific salient image regions. Nie et al. (2023) recently studied the effectiveness of transformer in TSF in terms of their ability to extract temporal relations, role of self-attention, temporal order preservation, embedding strategies, and their dependency on train set size. While in general they question the effectivity of transformer for forecasting, new transformer-based approaches continue to appear (Zeng et al., 2023), consistently improving the state-of-the-art results on common forecasting benchmarks.

Manifold learning analysis. Motivated by the ubiquitous manifold hypothesis, several existing approaches investigate geometric features of data representations across different layers. In Hauser & Ray (2017), the authors formalize a Riemannian geometry theory of deep neural networks (DNN) and show that residual neural networks are finite difference approximations of dynamical systems. Yu et al. (2018) compare two neural networks by inspecting the Riemann curvature of the learned representations in fully connected layers. Cohen et al. (2020) examine the dimension, radius and capacity throughout the training process, and they suggest that manifolds become linearly separable towards the end of the layer's hierarchy. Doimo et al. (2020) analyzed DNNs trained on ImageNet and found that the probability density of neural representations across different layers exhibits a hierarchical clustering pattern that aligns with the semantic hierarchy of concepts. Stephenson et al. (2021) conclude that data memorization primarily occurs in deeper layers, due to decreasing object manifolds' radius and dimension, and that generalization can be restored by reverting the weights of the final layers to an earlier epoch. Perhaps closest to our approach are the works by Ansuini et al. (2019) and Kaufman & Azencot (2023), where the authors estimate the intrinsic dimension and Riemannian curvature, respectively, of popular deep convolutional neural networks. Both works showed characteristic profiles and a strong correlation between the estimated geometric measure and the generalization error. Recently, Valeriani et al. (2023) investigated the intrinsic dimension and probability density of large transformer models in the context of classification tasks on protein and genetic sequence datasets. Complementary to previous works, our study focuses on the setting of regression time series forecasting problems using multivariate real-world time series datasets.

Manifold learning in time series forecasting. Unfortunately, latent representations of deep TSF received less attention in the literature, and thus we discuss works that generally investigate TSF from a manifold learning perspective. Papaioannou et al. (2022) embed high-dimensional time series into a lower-dimensional space using nonlinear manifold learning techniques to improve forecasting. Similarly, Han et al. (2018) proposed a novel framework, which performs nonuniform embedding, dynamical system revealing, and time-series prediction. In Li et al. (2021), the authors exploit manifold learning to extract the low-dimensional intrinsic patterns of electricity loads, to be used as input to recurrent modules for predicting low-dimensional manifolds. Lin et al. (2006) employ a dynamic Bayesian network to learn the underlying nonlinear manifold of time series data, whereas Shnitzer et al. (2017) harness diffusion maps to recover the states of dynamical systems. Finally, manifold-based distance functions for time series were proposed in Rodrigues et al. (2018); O'Reilly et al. (2017).

3 Background and Method

Time series forecasting. Given a dataset of multivariate time series sequences $\mathcal{D} := \{x^j_{1:T+h}\}_{j=1}^N$ where $x_{1:T+h} = x_1, \ldots, x_{T+h} \subset \mathbb{R}^D$, the goal in time series forecasting (TSF) is to accurately forecast the series $x_{T+1:T+h}$, based on the sequence $x_{1:T}$, where we omit j for brevity. The values T and h are typically referred to as lookback and horizon, respectively. The forecast accuracy can be measured in several ways of which the mean squared error (MSE) is the most common. We denote by $\tilde{x}_{T+1:T+h} = f(x_{1:T})$ the output of a certain forecast model, e.g., a neural network, then $e_{\text{MSE}} := \frac{1}{h}\sum_{t=T+1}^{T+h} \|x_t - \tilde{x}_t\|_2^2$ is the forecast error. In our study, we consider $T = 96$, and $h = 96, 192, 336$ and 720, and standard benchmark datasets including Electricity, Traffic, ETTm1, ETTm2, ETTh1, ETTh2, and weather (Wu et al., 2021). In App. A, we provide a detailed description of the datasets and their properties.

Transformer-based TSF deep neural networks. State-of-the-art (SOTA) deep time series forecasting models appeared only recently (Oreshkin et al., 2020), enjoying a rapid development of transformer-based architectures, e.g., Zhou et al. (2021); Wu et al. (2021); Liu et al. (2021); Zhou et al. (2022); Nie et al. (2023), among many others. In what follows, we will focus on Autoformer (Wu et al., 2021) and FEDformer (Zhou et al., 2022) as they are established architectures that are still considered SOTA. In App. C, we also mention additional TSF models and their analysis. Please see Fig. 1 for a schematic illustration of the architecture we investigate. The network is composed of two encoder blocks and a single decoder block, where the encoder and decoder blocks include two and three sequence decomposition layers, respectively. Both Autoformer and FEDformer utilize these decomposition layers to extract trend and seasonality information. The network

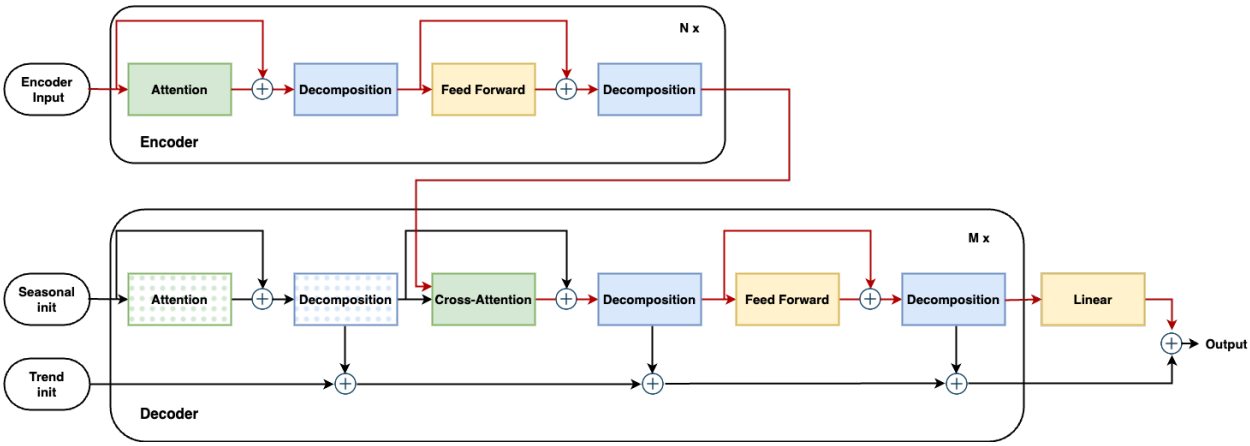

Figure 1: We study Transformer-based architectures (Wu et al., 2021; Zhou et al., 2022) that include two encoders and one decoder, and an output linear layer. We sample geometric features in the output of sequence decomposition layers, depicted as solid blue blocks.

includes two inputs, the input signal X which is fed to the encoder and the seasonality information X_s of X that serves as the input to the decoder. Note that the cross-attention module in the decoder receives two independent data streams, one is the output of the encoder while the other is a function of X_s. Our goal is to analyze the geometrical features of the data propagation along the layers of the network in a serial manner (with respect to depth), therefore, we focus on the path from the encoder through the decoder to the output. Namely, we follow data along the red trajectory, while ignoring the black trajectory. In particular, our analysis is based on sampling geometric properties of the data manifold after every decomposition module and after the final linear layer of the network. We chose the output of the decomposition blocks rather than the attention blocks since the Fourier Cross-correlation layer of the FEDformer model outputs almost identical values for all samples in the series, yielding zero curvature estimates.

Series decomposition. In time series analysis it is common to use the idea of decomposition to learn complex temporal patterns (Anderson & Kendall, 1976; Cleveland et al., 1990). The time series can be partitioned to their trend and seasonal parts, which in turn can be used to better capture global properties. In the time series forecasting domain, prior to Autoformer, time series decomposition was mainly used as a pre-processing tool, applied on the input data. Both Autoformer and FEDformer utilize decomposition as a core component, enabling it to progressively decompose the hidden time series throughout the entire forecasting process. This includes both historical (input) data and the predicted (output) intermediate results. The series decomposition block (noted as Decomposition in 1) receives a time series signal as an input and outputs the seasonality (right arrow) and the trend (bottom arrow) such that their summation results on the original signal.

[I'm not sure what do you mean by right arrow and bottom arrow in the paragraph above]

Encoder and decoder. The encoder is utilized for modeling the seasonal parts as it discards the trend output from the decomposition block. The encoder's output, which includes past seasonal information, will serve as cross-information to assist the decoder in refining the prediction results. The decoder serves two purposes, it accumulates the trend components and propagates the seasonal component. The final prediction is the sum of the seasonal and trend components.

Geometric properties of data manifolds. The fundamental assumption in our work is that data representations computed across layers of transformer-based models lie on Riemannian manifolds (Lee, 2006). We

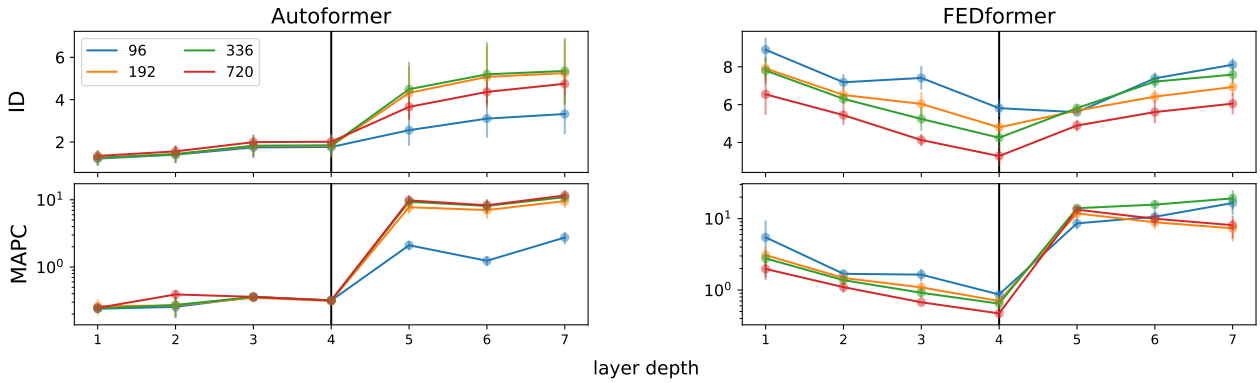

Figure 2: **Intrinsic dimension and mean absolute principal curvature along the layers of Autoformer and FEDformer on traffic dataset for multiple forecasting horizons.** Top) intrinsic dimension. Bottom) mean absolute principal curvature. For each model, both ID and MAPC share a similar profile across different forecasting horizons.

are interested in computing the intrinsic dimension (ID) and the mean absolute principal curvature (MAPC) of the manifold, following recent work on deep CNNs (Ansuini et al., 2019; Kaufman & Azencot, 2023). We compute the ID using the TwoNN method (Facco et al., 2017) that utilizes the Pareto distribution of the ratio between the distances to the two closest neighbors to estimate the dimension. For the MAPC, we employ the curvature aware manifold learning (CAML) technique (Li, 2018) that parametrizes the manifold via its second-order Taylor expansion, allowing to estimate curvatures via the eigenvalues of local Hessian matrices. More details related to ID and MAPC are provided in App. D.

Data collection. In this study, every architecture is trained on all datasets and horizons, using 10 different seed numbers. For every combination of model, dataset, horizon and seed, we extract the latent data representations across layers, and we compute the ID and MAPC. The intrinsic dimension is estimated on 500k point samples from $\mathcal{D}$, resulting in a single scalar value d. The estimated ID is used as an input to the CAML algorithm that uses 100k samples, and it returns $d(D-d)$ principal curvatures per point, where D is the extrinsic dimension. The MAPC of a manifold is calculated by computing the mean absolute value for each point and taking the mean over all points.

4 Results

4.1 Data manifolds share similar geometric profiles

In our first empirical result, we compute the intrinsic dimension (ID) and mean absolute principal curvature (MAPC) across the layers of Autoformer and FEDformer models on the traffic dataset. In Fig. 2, we plot the ID (top row) and MAPC (bottom row) for Autoformer (left column) and FEDformer (right column) on multiple forecast horizons $= 96, 192, 336, 720$. The x-labels refer to the layers we sample, where labels 1 to 4 refer to two sequence decomposition layers per encoder block (and thus four in total), labels 5 to 6 denote the decoder decomposition layers, and label 7 is the linear output layer, see Fig. 1 for the network scheme. Our results indicate that during the encoding phase, the ID and the MAPC are relatively fixed for Autoformer and decrease for FEDformer, and during the decoder module, these values generally increase with depth. Specifically, the ID values change from $\min(\text{ID}) = 1.2$ to $\max(\text{ID}) = 8.1$, showing a relatively small variation across layers. In comparison, the mean absolute principal curvature values present a larger deviation as they range from $\min(\text{MAPC}) = 0.2$ to $\max(\text{MAPC}) = 19.2$.

Remarkably, it can be observed from Fig. 2 that both Autoformer and FEDformer feature similar ID and MAPC profiles in terms of values. Further, a strong similarity in trend can be viewed across different forecast horizons per method. Moreover, Autoformer and FEDformer differ during the encoding phase (layers 1 to 4), but match quite well during the decoding and output phases (layers 5 to 7). Our intrinsic

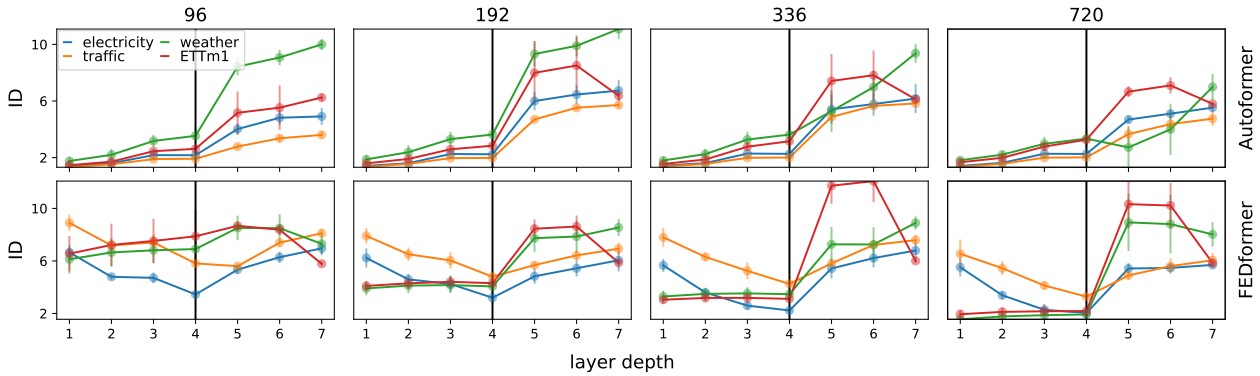

Figure 3: **ID profiles across layers of Autoformer and FEDformer on electricity, traffic, weather and ETTm1 datasets for multiple forecasting horizons.** Each panel includes ID profiles per dataset, for several horizons (left to right) and architectures (top to bottom).

dimension estimations stand in contrast to existing results on classification tasks with CNN and transformer architectures that observed a "hunchback" ID profile (Ansuini et al., 2019; Valeriani et al., 2023). That is, prior work found the intrinsic dimension to increase significantly at the first few layers, and then, it presented a sharp decrease with depth.

This qualitative difference in ID can be attributed to the differences between classification models as in Ansuini et al. (2019) in comparison to regression TSF networks we consider here. In particular, deep classification neural networks essentially recover the related low-dimensional manifold, to facilitate a linear separation of classes (Cohen et al., 2020), and thus one may expect a low ID toward the final layers of the network. On the other hand, forecast regression models aim to encode the statistical distribution of input data, to facilitate forecast of the horizon windows and it is typically of a higher dimension due to spurious variations. Namely, regression TSF models are expected to learn an underlying low-dimensional and simple representation while encoding. Then, a more complex manifold that better reflects the properties of the data is learned during decoding.

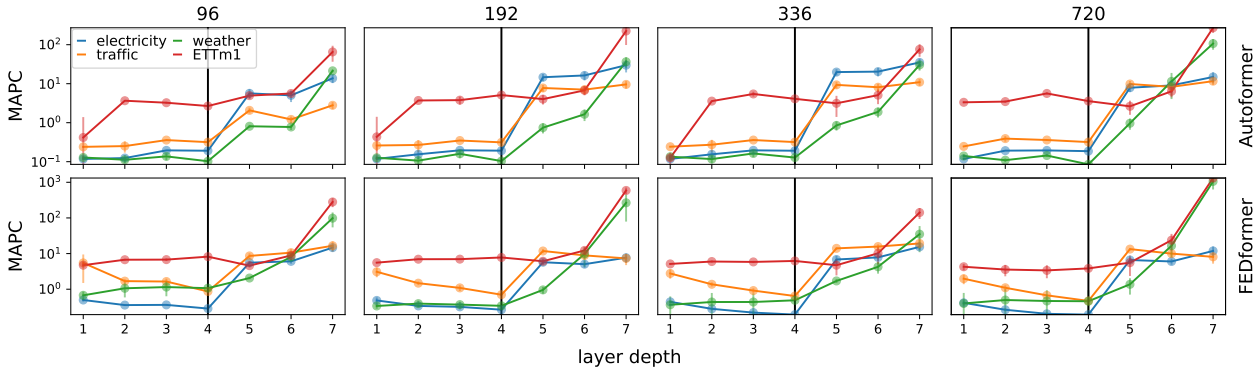

Figure 4: **MAPC profiles across layers of Autoformer and FEDformer on electricity, traffic, weather and ETTm1 for multiple horizons.** Each panel includes MAPC profiles per dataset, for several horizons (left to right) and architectures (top to bottom).

Importantly, while our intrinsic dimension profiles do not exhibit the "hunchback" shape identified in Ansuini et al. (2019), our estimated ID d is significantly smaller than the extrinsic dimension $D = 512$, in correspondence with existing work. Finally, our MAPC profiles attain a "step-like" appearance, similar to the results in Kaufman & Azencot (2023), where they identify a sharp increase in curvature in the final layer, and we observe such a jump in the decoder.

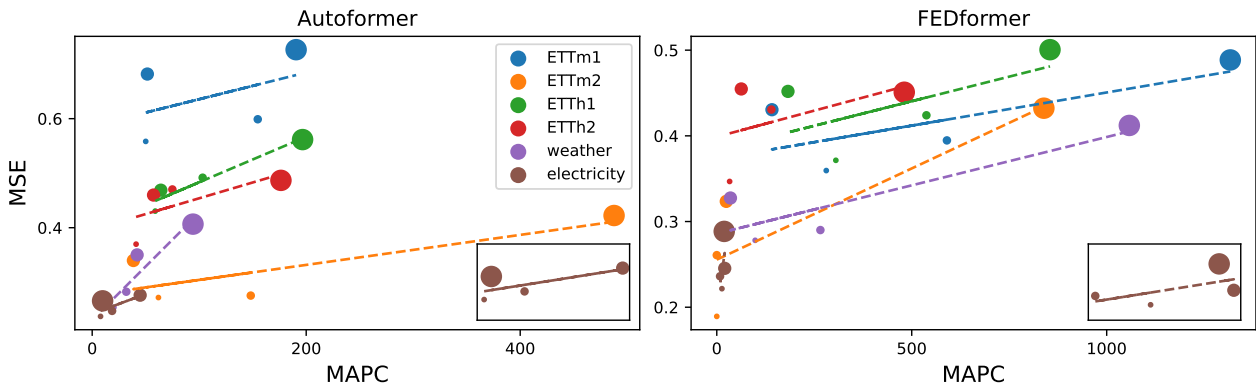

Figure 5: **MAPC is correlated with model performance.** The test mean squared error is proportional to the MAPC on multiple datasets. Each color represents a different data set while the size of the dot is determined by the forecast horizon (longer horizon results in a larger dot). The dashed line represents the best linear fit for each dataset. Due to different scales, an inlay of the electricity data set is added to the bottom right corner of each architecture.

To extend our analysis, we present in Fig. 3 and 4 the ID and MAPC profiles, respectively, for Autoformer (top) and FEDformer (bottom) for several horizons using multiple different datasets. For all Autoformer configurations, the IDs in Fig. 3 generally increase with depth, and the IDs of FEDformer present a "v"-shape for electricity and traffic and a "step"-like behavior for weather and ETTm1. Interestingly, ETTm1 (and other ETT* datasets, please see Fig. 8) shows a hunchback trend, however, the drop of ID in the final layer is due to ETT* datasets consisting of a total of seven features, and thus we do not consider this behavior to be characteristic to the network. As in Fig. 2 and existing work (Ansuini et al., 2019), the intrinsic dimension d is much lower than its extrinsic counterpart D. Our MAPC results in Fig. 4 indicate a shared step-like behavior in general for all models, horizons, and datasets, where the main difference is where the curvature increase occurs. For electricity and traffic, we observe a sharp increase at the beginning of the decoder block, whereas for weather and ETTm1, the increase often appears at the final layer. Additionally, the maximal curvature values for weather and ETTm1 tend to be higher than those of electricity and traffic. Overall, our results suggest that weather and ETTm1 are associated with manifolds whose geometric features match. This observation can be justified by the known correlation between electricity transformer temperature (ETT) and climate change (Hashmi et al., 2013; Gao et al., 2018). Similarly, electricity consumption (electricity) and road occupancy (traffic) attain a shared behavior that may be explained due to the strong seasonality component in these datasets (Zeng et al., 2023).

4.2 Final MAPC is correlated with performance

In what follows, we investigate whether geometric properties of the learned manifold are associated with inherent features of the model. For instance, previous works find a strong correlation between the ID (Ansuini et al., 2019) and MAPC (Kaufman & Azencot, 2023) with model performance and generalization. Specifically, the intrinsic dimension in the last hidden layer is correlated with the top-5 score on image classification, i.e., lower ID is associated with lower error. Similarly, large normalized MAPC gap between the penultimate and final layers of CNNs is related to high classification accuracy. These correlations are important as they allow developers and practitioners to evaluate and compare deep neural networks based on statistics obtained directly from the train set. This is crucial in scenarios where, e.g., the test set is unavailable during model design.

We show in Fig. 5 plots of the test mean squared error (e_{MSE}) vs. the MAPC in the final layer of Autoformer and FEDformer models trained on ETTm1, ETTm2, ETTh1, ETTh2, weather, and electricity. For each dataset, we plot four colored circles corresponding to the four different horizons, where each circle is scaled proportionally to the horizon length. The dashed graphs are generated by plotting the e_{MSE} with respect to the MAPC. We find a positive slope in all Autoformer and FEDformer models with an average correlation

Table 1: Correlation coefficients of test MSE vs. final MAPC for several datasets and architectures.

	Autoformer	FEDformer
ETTm1	0.46	0.74
ETTm2	0.81	0.86
ETTh1	0.97	0.63
ETTh2	0.67	0.5
weather	0.93	0.58
electricity	0.71	0.7

coefficient of 0.76 and 0.7, respectively (see full results in Table. 1). In all cases, we observe a correlation between the test MSE and final MAPC, namely, the model performs *better* as curvature *decreases*, as shown in Fig. 5.

As in Sec. 4.1, we identify different characteristics for TSF models with respect to classification neural networks. While popular CNNs show better performance when the MAPC gap is high (Kaufman & Azencot, 2023), we report an opposite trend, namely, better models are associated with a *lower* MAPC. We believe this behavior may be attributed to classification networks requiring final high curvatures to properly classify the statistical distribution of the input information and its large variance. In addition, we note the relatively flat slope profiles presented across all datasets and architectures. Essentially, these results indicate that while the manifolds become more complex in terms of curvature, transformers yield a relatively fixed MSE, regardless of the underlying MAPC. Thus, our results may hint that Autoformer and FEDformer are *not* expressive enough. Indeed, the datasets we consider consist of many features (on the range of hundreds for electricity and traffic). Therefore, while TSF approaches may need highly expressive networks to model these datasets due to their complex statistics, it might be that current approaches can not achieve better representations in these cases, and they tend to get "stuck" on a local minimum. We hypothesize that significantly deeper and more expressive TSF models as in Nie et al. (2023) and is common in classification (He et al., 2016) may yield double descent forecasting architectures (Belkin et al., 2019).

maybe add later: We observe a relatively horizontal slope on electricity and traffic, where the MAPC changes more in comparison to the MSE

4.3 Manifold dynamics during training

Our analysis above focuses on fully trained deep neural networks and the geometric properties of the learned manifolds. In addition, we also investigate below the evolution of manifolds during training and their structure at initialization. In prior works, Ansuini et al. (2019) observed that randomly initialized architectures

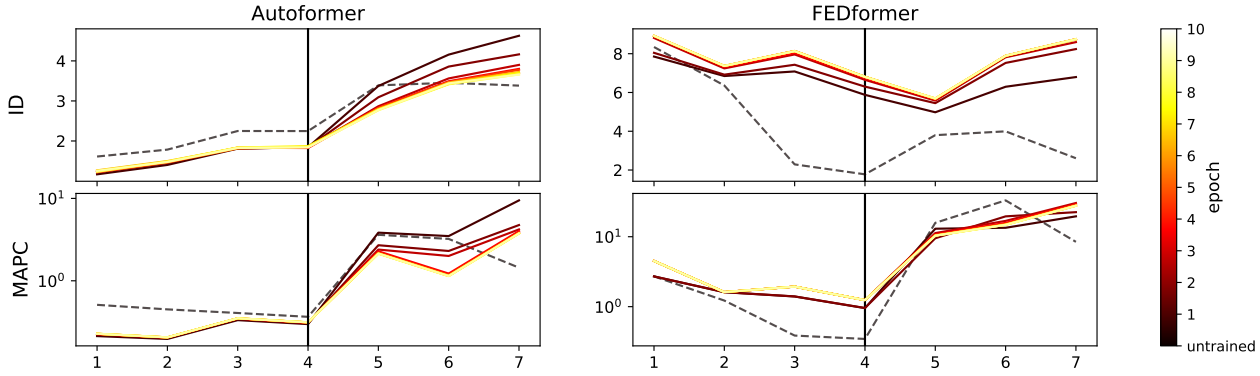

Figure 6: **Training dynamics of the ID and MAPC on traffic dataset.** The plot shows how the ID and MAPC change during training, colored by the training epoch. The black dashed line represents the untrained model.

exhibit a constant ID profile, and further, there is an opposite trend in ID in intermediate layers vs. final layers of convolutional neural networks during training. Kaufman & Azencot (2023) find that untrained models have different MAPC profiles than trained networks, and they observe that the normalized MAPC gap consistently increases with model performance during training. Moreover, ID and MAPC profiles converge consistently to their final configuration as training proceeds. Motivated by their analysis, we study the general convergence and trend of ID and MAPC of TSF models during training evolution.

We show in Fig. 6 the ID and MAPC profiles for Autoformer and FEDformer during training, where each plot is colored by its sampling epoch using the `hot` colormap. First, the untrained ID and MAPC profiles (dashed black) are somewhat random in comparison to the other geometric profiles. Second, the overall convergence to the final behavior is extremely fast, requiring approximately five epochs to converge in all the configurations which is consistent with the results of Bonheme & Grzes (2022) where they show that the ID does not change much after the first epoch. Moreover, the encoder in the Autoformer converges within two epochs, whereas the FEDformer model needs more epochs for the encoder to converge. Third, the decoder shows a slower convergence for both methods, suggesting that "most" learning takes place in the decoder component of transformer-based forecasting models. The previous observation aligns with the works of Bonheme & Grzes (2023); Raghu et al. (2017), showing that representations of layers closer to the input tend to stabilize quicker. More specifically, Bonheme & Grzes (2023) show that encoders' representations are generic while decoders' are specific, resulting in a slight change of the encoders' representations during training. Finally, except for the untrained profiles, the ID and MAPC curves during training are generally similar across different epochs. The latter observation may mean that Autoformer and FEDformer mainly perform fine-tuning training as their underlying manifolds do not change much during training.

4.4 Distribution of principal curvatures

The CAML algorithm (Li, 2018) we employ for estimating the principal curvatures produces $d(D-d)$ values per point, yielding a massive amount of curvature information for analysis. Following Kaufman & Azencot (2023), we compute and plot in Fig. 7 the distribution of principal curvatures for every layer, shown as a smooth histogram for Autoformer and FEDformer models on electricity and traffic datasets with horizons $= 192, 336$. The histogram plots are colored by the network depth using the `hot` colormap. These distributions strengthen our analysis in Sec. 4.1 where we observe a "step"-like pattern in MAPC, where the sharp jump in curvature occurs at the beginning of the decoder. Indeed, the curves in Fig. 7 related to layers 1 to 4 span a smaller range in comparison to the curves in layers 5 to 7. Further, the histograms show that the distribution of curvature is relatively fixed across the encoder blocks, and similarly, a different but rather fixed profile appears in the decoder.

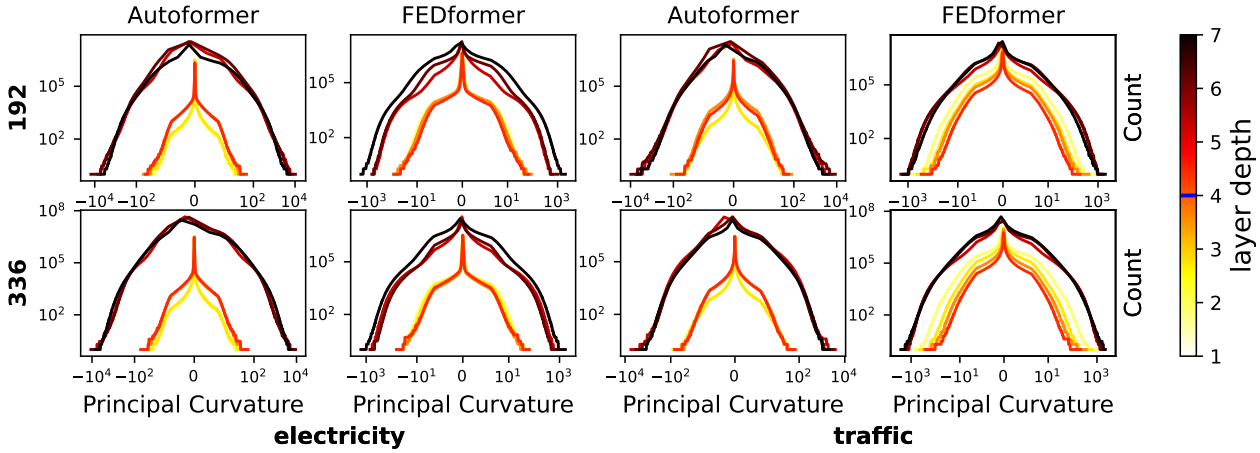

Figure 7: **Principal curvature distribution of Autoformer and FEDformer.** Each plot shows the histogram profiles of principal curvatures per layer, colored by their relative depth.

5 Discussion

Deep neural networks are composed of several computation layers. Each layer receives inputs from a preceding layer, it applies a (nonlinear) transformation, and it feeds the outputs to a subsequent layer. The overarching theme in this work is the investigation of data representations arising during the computation of deep models. To study these latent representations, we adopt a ubiquitous ansatz, commonly known as the *manifold hypothesis* (Coifman & Lafon, 2006): We assume that while such data may be given in a high-dimensional and complex format, it lies on or next to a low-dimensional manifold. The implications of this inductive bias are paramount; manifolds are rich mathematical objects that are actively studied in theory (Lee, 2006) and practice (Chaudhry et al., 2018), allowing one to harness the abundant classical tools and recent developments to study deep representations.

Our study aligns with the line of works that aims at better understanding the inner mechanisms of deep neural networks. Indeed, while modern machine learning has been dominating many scientific and engineering disciplines since the appearance of AlexNet (Krizhevsky et al., 2012), neural net architectures are still considered not well understood by many. In this context, the manifold ansatz is instrumental—existing analysis works investigate geometric features of latent manifolds and their relation to the underlying task and model performance. For instance, Ansuini et al. (2019) compute the intrinsic dimension of popular convolutional neural networks. Following their work, Kaufman & Azencot (2023) estimate the mean absolute principal curvatures. However, while CNNs are relatively studied from the manifold viewpoint, impactful sequential transformer models (Vaswani et al., 2017), received less attention (Valeriani et al., 2023). The lack of analysis is even more noticeable for the recent state-of-the-art transformer-based *time series forecasting* works, e.g., Wu et al. (2021). The main objective of our work is to help bridge this gap and study deep forecasting models trained on common challenging datasets from a manifold learning viewpoint.

We compute the ID and MAPC of data representations from several different deep architectures, forecasting tasks and datasets. To this end, we employ differentiable tools (Facco et al., 2017; Li, 2018) that produce a single scalar ID and many principle curvatures combined to a single scalar MAPC, per manifold. Our results raise several intriguing observations, many of them are in correspondence with existing work. First, the ID is much smaller than the extrinsic dimension, reflecting that learned manifolds are indeed low-dimensional. Second, the ID and MAPC profiles across layers are similar for many different architectures, tasks, and datasets. In particular, we identify two phases, where in the encoder, ID and MAPC are decreasing or stay fixed, and in the decoder, both geometric features increase with depth. Third, the MAPC in the final layer is strongly correlated with model performance, presenting a correlation, i.e., error is lower when MAPC is lower. Fourth, we observe that related but different datasets attain similar manifolds, whereas unrelated datasets are associated to manifolds with different characteristics. Finally, untrained models present random ID and MAPC profiles that converge to their final configuration within a few epochs.

Our analysis and observations lie at the heart of the differences between classification and regression tasks; a research avenue that only recently had started to be addressed more frequently (Muthukumar et al., 2021; Yao et al., 2022). Our results indicate a fundamental difference between image classification and time series forecasting regression models: while the former networks shrink the ID significantly to extract a meaningful representation that is amenable for linear separation, TSF models behave differently. Indeed, the ID generally increases with depth, perhaps to properly capture the large variance of the input domain where regression networks predict. Moreover, while a high MAPC gap seems to be important for classification, we find an opposite trend in TSF regression problems. In conclusion, we believe that our work sets the stage for a deeper investigation of classification vs. regression from a manifold learning and other perspectives. We believe that fundamental advancements on this front will lead to powerful machine learning models, better suited for solving the task at hand.

References

Oliver D. Anderson and M. G. Kendall. Time-series. 2nd edn. *The Statistician*, 25:308, 1976. URL `https://api.semanticscholar.org/CorpusID:134001785`.

Alessio Ansuini, Alessandro Laio, Jakob H Macke, and Davide Zoccolan. Intrinsic dimension of data representations in deep neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR*, 2015.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

Lisa Bonheme and Marek Grzes. Fondue: an algorithm to find the optimal dimensionality of the latent representations of variational autoencoders. *arXiv preprint arXiv:2209.12806*, 2022.

Lisa Bonheme and Marek Grzes. How good are variational autoencoders at transfer learning? *arXiv preprint arXiv:2304.10767*, 2023.

Pratik Prabhanjan Brahma, Dapeng Wu, and Yiyuan She. Why deep learning works: A manifold disentanglement perspective. *IEEE transactions on neural networks and learning systems*, 27(10):1997–2008, 2015.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021.

Cristian Challu, Kin G Olivares, Boris N Oreshkin, Federico Garza, Max Mergenthaler, and Artur Dubrawski. N-HiTS: Neural hierarchical interpolation for time series forecasting. *arXiv preprint arXiv:2201.12886*, 2022.

Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 532–547, 2018.

Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 782–791, 2021.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. What does bert look at? an analysis of bert's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Association for Computational Linguistics, 2019.

Robert B Cleveland, William S Cleveland, Jean E McRae, Irma Terpenning, et al. Stl: A seasonal-trend decomposition. *J. off. Stat*, 6(1):3–73, 1990.

Uri Cohen, SueYeon Chung, Daniel D Lee, and Haim Sompolinsky. Separability and geometry of object manifolds in deep neural networks. *Nature communications*, 11(1):746, 2020.

Ronald R Coifman and Stéphane Lafon. Diffusion maps. *Applied and computational harmonic analysis*, 21 (1):5–30, 2006.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*. Association for Computational Linguistics, 2019.

Diego Doimo, Aldo Glielmo, Alessio Ansuini, and Alessandro Laio. Hierarchical nucleation in deep neural networks. *Advances in Neural Information Processing Systems*, 33:7526–7536, 2020.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

Elena Facco, Maria d'Errico, Alex Rodriguez, and Alessandro Laio. Estimating the intrinsic dimension of datasets by a minimal neighborhood information. *Scientific reports*, 7(1):12140, 2017.

Xiang Gao, C Adam Schlosser, and Eric R Morgan. Potential impacts of climate warming and increased summer heat stress on the electric grid: a case study for a large power transformer (lpt) in the northeast united states. *Climatic change*, 147:107–118, 2018.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pp. 6645–6649. Ieee, 2013.

Min Han, Shoubo Feng, CL Philip Chen, Meiling Xu, and Tie Qiu. Structured manifold broad learning system: A manifold perspective for large-scale chaotic time series analysis and prediction. *IEEE Transactions on Knowledge and Data Engineering*, 31(9):1809–1821, 2018.

Tatsunori B Hashimoto, David Alvarez-Melis, and Tommi S Jaakkola. Word embeddings as metric recovery in semantic spaces. *Transactions of the Association for Computational Linguistics*, 4:273–286, 2016.

Murtaza Hashmi, Matti Lehtonen, H Seppo, et al. Effect of climate change on transformers loading conditions in the future smart grid environment. *Open Journal of Applied Sciences*, 3(02):24, 2013.

Michael Hauser and Asok Ray. Principles of riemannian geometry in neural networks. *Advances in neural information processing systems*, 30, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Andrej Karpathy, Justin Johnson, and Li Fei-Fei. Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078*, 2015.

Ilya Kaufman and Omri Azencot. Data representations' study of latent image manifolds. In *International Conference on Machine Learning, ICML*, 2023.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

John M Lee. *Riemannian manifolds: an introduction to curvature*, volume 176. Springer Science & Business Media, 2006.

Jinghua Li, Shanyang Wei, and Wei Dai. Combination of manifold learning and deep learning algorithms for mid-term electrical load forecasting. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

Yangyang Li. Curvature-aware manifold learning. *Pattern Recognition*, 83:273–286, 2018.

Yuanzhi Li and Yingyu Liang. Learning overparameterized neural networks via stochastic gradient descent on structured data. *Advances in neural information processing systems*, 31, 2018.

Ruei-Sung Lin, Che-Bin Liu, Ming-Hsuan Yang, Narendra Ahuja, and Stephen E. Levinson. Learning nonlinear manifolds from time series. In *Computer Vision - ECCV*, 2006.

Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X Liu, and Schahram Dustdar. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International conference on learning representations*, 2021.

Vidya Muthukumar, Adhyyan Narang, Vignesh Subramanian, Mikhail Belkin, Daniel Hsu, and Anant Sahai. Classification vs regression in overparameterized regimes: Does the loss function matter? *The Journal of Machine Learning Research*, 22(1):10104–10172, 2021.

Xuan Son Nguyen, Luc Brun, Olivier Lézoray, and Sébastien Bougleux. A neural network based on spd manifold learning for skeleton-based hand gesture recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12036–12045, 2019.

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, ICLR*, 2023.

Colin O'Reilly, Klaus Moessner, and Michele Nati. Univariate and multivariate time series manifold learning. *Knowl. Based Syst.*, 133:1–16, 2017.

Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-BEATS: neural basis expansion analysis for interpretable time series forecasting. In *8th International Conference on Learning Representations, ICLR*, 2020.

Panagiotis G Papaioannou, Ronen Talmon, Ioannis G Kevrekidis, and Constantinos Siettos. Time-series forecasting using manifold learning, radial basis function interpolation, and geometric harmonics. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 32(8), 2022.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. *Advances in neural information processing systems*, 30, 2017.

Pedro Luiz Coelho Rodrigues, Marco Congedo, and Christian Jutten. Multivariate time-series analysis via manifold learning. In *2018 IEEE Statistical Signal Processing Workshop, SSP*, pp. 573–577, 2018.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8:842–866, 2021.

Hang Shao, Abhishek Kumar, and P Thomas Fletcher. The riemannian geometry of deep generative models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 315–323, 2018.

Tal Shnitzer, Ronen Talmon, and Jean-Jacques E. Slotine. Manifold learning with contracting observers for data-driven time-series analysis. *IEEE Trans. Signal Process.*, 65(4):904–918, 2017.

Cory Stephenson, Suchismita Padhy, Abhinav Ganesh, Yue Hui, Hanlin Tang, and Sue Yeon Chung. On the geometry of generalization and memorization in deep neural networks. In *9th International Conference on Learning Representations, ICLR 2021*, 2021.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. Bert rediscovers the classical nlp pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4593–4601, 2019.

Yao-Hung Hubert Tsai, Shaojie Bai, Makoto Yamada, Louis-Philippe Morency, and Ruslan Salakhutdinov. Transformer dissection: a unified understanding of transformer's attention via the lens of kernel. *arXiv preprint arXiv:1908.11775*, 2019.

Lucrezia Valeriani, Diego Doimo, Francesca Cuturello, Alessandro Laio, Alessio Ansuini, and Alberto Cazzaniga. The geometry of hidden representations of large transformer models. *arXiv preprint arXiv:2302.00294*, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Jesse Vig. A multiscale visualization of attention in the transformer model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 37–42, 2019.

Eric Wallace, Jens Tuyls, Junlin Wang, Sanjay Subramanian, Matt Gardner, and Sameer Singh. Allennlp interpret: A framework for explaining predictions of NLP models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP*, pp. 7–12, 2019.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34: 22419–22430, 2021.

Huaxiu Yao, Yiping Wang, Linjun Zhang, James Y Zou, and Chelsea Finn. C-mixup: Improving generalization in regression. *Advances in Neural Information Processing Systems*, 35:3361–3376, 2022.

Tao Yu, Huan Long, and John E Hopcroft. Curvature-based comparison of two neural networks. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pp. 441–447. IEEE, 2018.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pp. 818–833. Springer, 2014.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11121–11128, 2023.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11106–11115, 2021.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International Conference on Machine Learning*, pp. 27268–27286. PMLR, 2022.

A TSF datasets

Blow we provide a detailed description of the datasets used in the paper. A summery of the datasets can be found in Tab. 2.

Electricity Transformer Temperature (ETT) Zhou et al. (2021): The ETT contains electricity power load (six features) and oil temperature collected over a period of two years from two countries in China. The dataset is versatile and exhibits short-term periodical patterns, long-term periodical patterns, long-term trends, and irregular patterns. The dataset is further divided to two granularity levels: ETTh1, ETTh2 for one hour level and ETTm1, ETTm2 for 15 minutes level.

Weather[1]: The dataset contains 21 meteorological sensors for a range of 1 year in Germany.

Electricity Consuming Load (ECL)[2]: It contains the hourly electricity consumption (Kwh) of 321 clients.

Traffic[3]: The dataset consists of hourly data spanning 24 months (2016-2018) obtained from the California Department of Transportation. This data provides information on road occupancy rates, measured by 862 sensors on freeways in the San Francisco Bay area, ranging between 0 and 1.

Table 2: We detail several statistics regarding the datasets considered in this work.

Dataset	Number of features	Number of train samples	Granularity
ETTm1, ETTm2	7	34369	15 minutes
ETTh1, ETTh2	7	34369	1 hour
Weather	21	36696	1 hour
ECL	321	18221	1 hour
Traffic	862	12089	1 hour

B Additional results

B.1 ETT* datasets analysis

To complement the results in the main article we add a comparison of the ID and MAPC of three different ETT datasets: ETTm1, ETTh1 and ETTh2. We notice that ETTm1, ETTh1 and ETTh2 show a hunchback trend for Autoformer while for FEDformer the hunchback trend appears for larger forecast horizons as shown in Fig. 8. Our MAPC results in Fig. 9 show that MAPC are relatively fixed and start to rise at the decoder.

C TSF models

Here we will provide a supplementary analysis of additional TSF models: vanilla Transformer (Vaswani et al., 2017), Informer (Zhou et al., 2021) and PatchTST (Nie et al., 2023).

Transformer and Infromer: These models have a similar architecture to the Autoformer and FEDformer as shown in Fig. 1. The models are composed of two encoder layers and one decoder layer, however, in contrast to Autoformer and FEDformer, Transformer and Infromer do not contain series decomposition layers. The analysis of Transformer and Infromer inspects the output of the encoder layers, decoder layer and the last linear layer. In Fig. 10 we observe trends similar to ones shown for the Autoformer and FEDformer. When comparing different datasets, we see similar trends, a monotonic increase in ID for the Transformer and a saw-like behaviour for Informer (see Fig.11). The MAPC across datasets for the Transformer exhibits a monotonic increase trend while the Informer has a "v"-shape (see Fig.12)

[1] https://www.bgc-jena.mpg.de/wetter/
[2] https://archive.ics.uci.edu/dataset/321/electricityloaddiagrams20112014
[3] https://pems.dot.ca.gov/

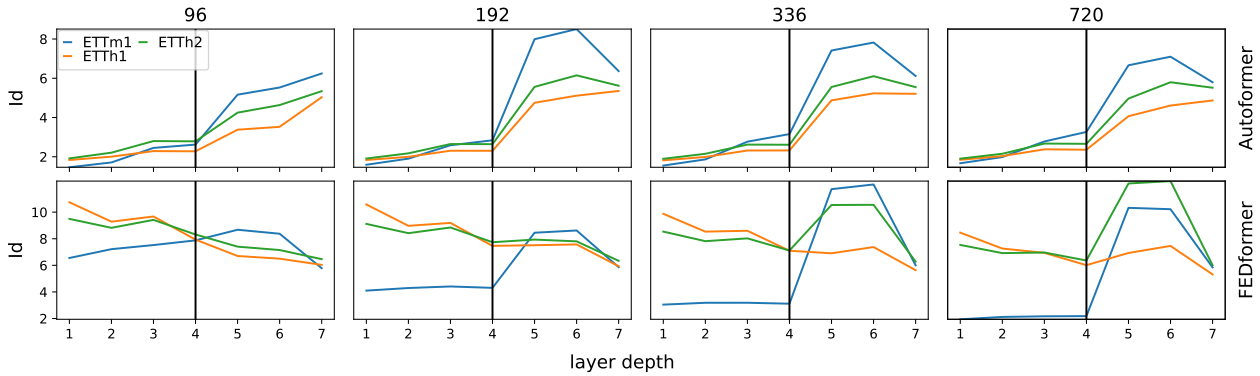

Figure 8: **ID profiles across layers of Autoformer and FEDformer on ETT datasets for multiple forecasting horizons.** Each panel includes separate ID profiles per dataset, for several horizons (left to right) and architectures (top to bottom).

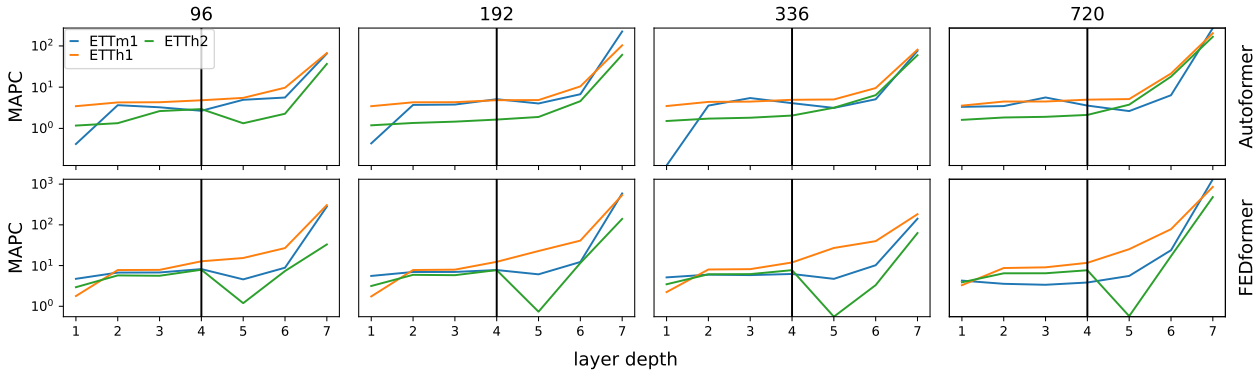

Figure 9: **MAPC profiles across layers of Autoformer and FEDformer on ETT datasets for multiple forecasting horizons.** Each panel includes separate MAPC profiles per dataset, for several horizons (left to right) and architectures (top to bottom).

PatchTST: PatchTST is a transformer-based model composed of vanilla Transformer encoders and a linear layer as a decoder. PatchTST differs from all other architecture mentioned in the paper by dividing a

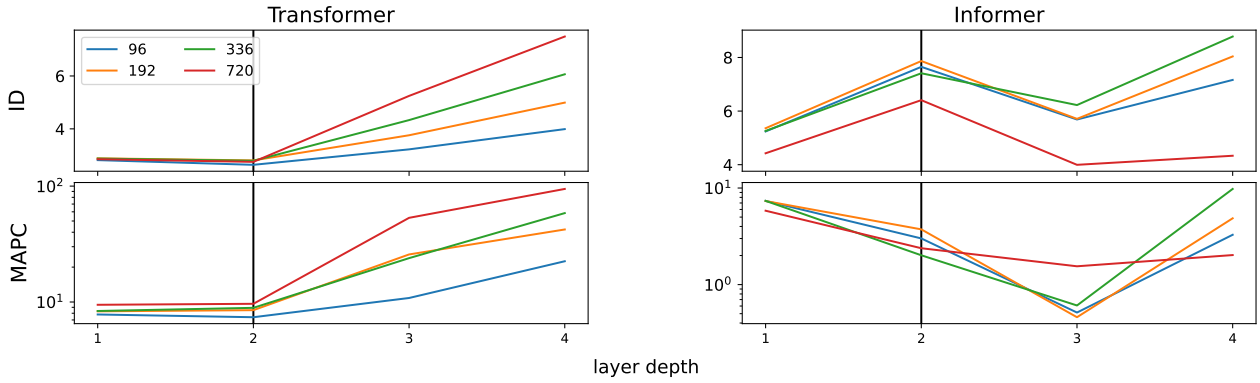

Figure 10: **Intrinsic dimension and mean absolute principal curvature along the layers of Transformer and Informer on traffic dataset for multiple forecasting horizons.** Top) intrinsic dimension. Bottom) mean absolute principal curvature. For each model, both ID and MAPC share a similar profile across different forecasting horizons.

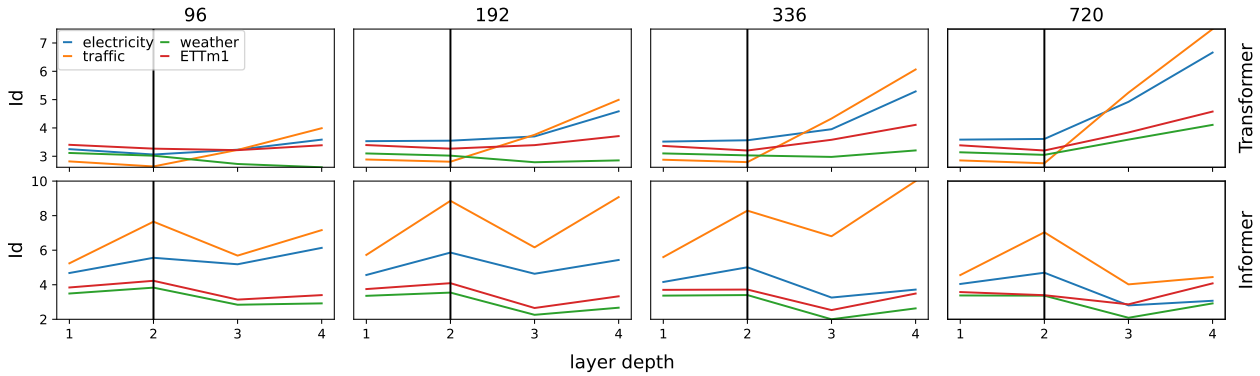

Figure 11: **ID profiles across layers of Transformer and Informer on electricity, traffic, weather and ETTm1 datasets for multiple forecasting horizons.** Each panel includes separate MAPC profiles per dataset, for several horizons (left to right) and architectures (top to bottom).

sequence into patches and using them as inputs to the network. Each patch is then embedded, and from that point on, all sequential information becomes inaccessible, i.e., we cannot manually extract temporal information for each time stamp in the patch. Our analysis focuses on time series manifolds, where each element represents a single point in time, while the latent representation of the PatchTST model lie on product manifolds where each element is a sequence of points in time. Similar to the analysis performed on the Transformer and Informer, we inspect the output of the encoder layers and the decoder layer which is the last linear layer. We notice that the ID remains relatively stable during the encoding phase and increases in the last layer, a behavior consistent across datasets and horizons. A similar trend is observed for the MAPC during the encoding phase where on some datasets the MAPC increases on the last layer and on other it decreases. See Figs. 13 and 14.

D Intrinsic dimension and mean absolute principal curvature

D.1 Intrinsic dimension

To estimate the ID of data representations in TSF neural networks, we use the TwoNN (Facco et al., 2017) global id estimator. The ID-estimator utilizes the distances only to the first two nearest neighbors of each point. This minimal selection helps reduce the impact of inconsistencies in the dataset during the estimation process.

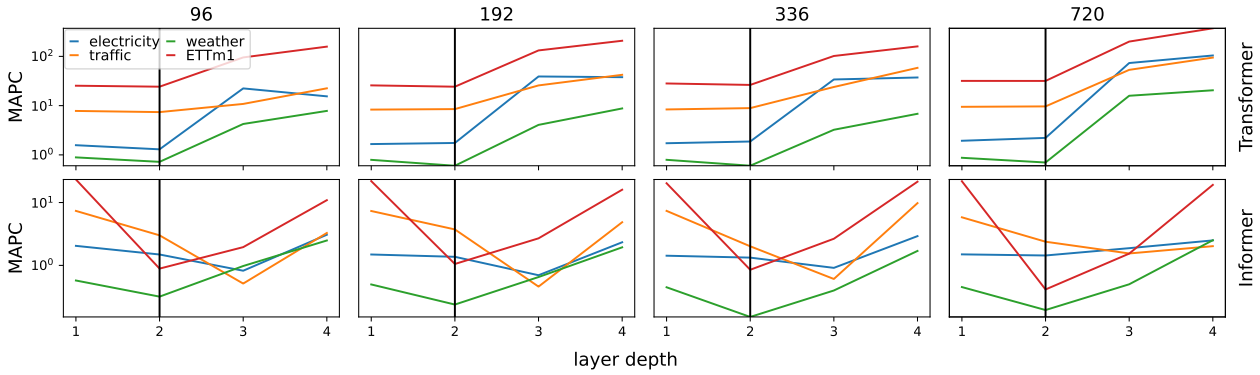

Figure 12: **MAPC profiles across layers of Transformer and Informer on electricity, traffic, weather and ETTm1 datasets for multiple forecasting horizons.** Each panel includes separate MAPC profiles per dataset, for several horizons (left to right) and architectures (top to bottom).

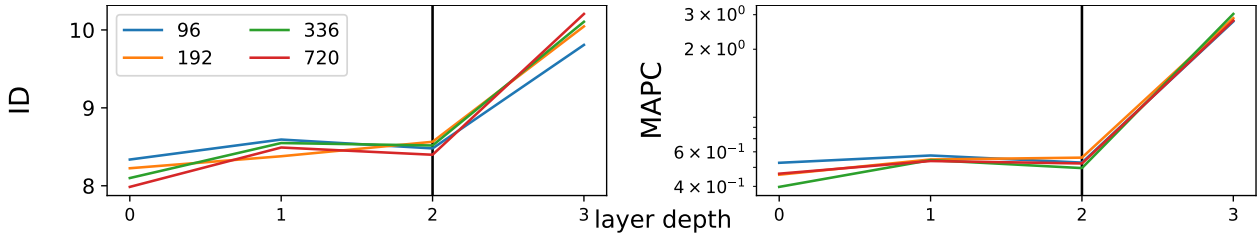

Figure 13: **ID and MAPC along the layers of PatchTST on weather dataset for multiple forecasting horizons.** Left) intrinsic dimension. Right) mean absolute principal curvature. ID and MAPC share a similar profile across different forecasting horizons.

Method Let $X = \{x_1, x_2, \cdots, x_N\}$ a set of points uniformly sampled on a manifold with intrinsic dimension d. For each point x_i, we find the two shortest distances r_1, r_2 from elements in $X \setminus \{x_i\}$ and compute the ratio $\mu_i = \frac{r_2}{r_1}$. It can be shown that $\mu_i, 1 \leq i \leq N$ follow a Pareto distribution with parameter $d+1$ on $[1, \infty)$, that is $f(\mu_i \mid d) = d\mu_i^{-(d+1)}$. While d can be estimated by maximizing the likelihood:

$$P(\mu_1, \mu_2, \cdots \mu_N \mid d) = d^N \prod_{i=1}^{N} \mu_i^{-(d+1)} \tag{1}$$

we follow the method proposed by Facco et al. (2017) based on the cumulative distribution $F(\mu) = 1 - \mu^{-d}$. The idea is to estimate d by a linear regression on the empirical estimate of $F(\mu)$. This is done by sorting the values of μ in ascending order and defining $F^{emp}(\mu_i) \doteq \frac{i}{N}$. A straight line is then fitted on the points of the plane $\{(\log \mu_i, -\log(1 - F_i^{emp}))\}_{i=1}^{N}$. The slope of the line is the estimated ID.

D.2 Data density

In comparison to the intrinsic dimension, which is a characteristic of the entire manifold, curvature information is local. Moreover, curvatures are calculated using second-order derivatives of the manifold. Consequently, our study assumes that the data is dense enough to compute curvatures. However, the latent representations of data, are both high-dimensional and sparse, which presents significant difficulties in calculating local differentiable values on such as curvature.

The typical characteristics of data used in machine learning require a large number of nearby points to create a stable neighborhood. One commonly used tool for this is k-Nearest-Neighbours (KNN). However,

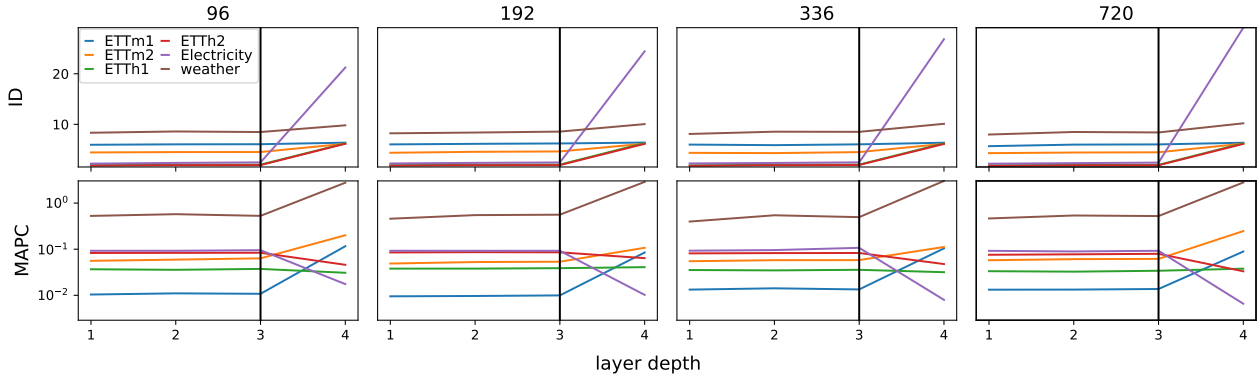

Figure 14: **ID and MAPC profiles across layers of PatchTST on ETT*, electricity and weather datasets for multiple forecasting horizons.** Each panel includes separate ID (top) and MAPC (bottom) profiles per dataset, for several horizons (left to right).

KNN can sometimes generate non-local and sparse neighborhoods, where the "neighbors" are effectively far apart in a Euclidean sense. Another approach is to use domain-specific augmentations, such as window cropping, window warping or slicing. However, this approach only explores a specific aspect of the data manifold and may overlook other important parts. A more effective approach, regardless of the domain, is to compute the Singular Value Decomposition (SVD) for each time series. This generates a close neighborhood by filtering out small amounts of noise in the data. This approach is well-motivated from a differential geometry standpoint, as it approximates the manifold at a point and samples the neighborhood.

Neighborhood generation. To improve the local density of time sereis samples, we use a procedure similar to Yu et al. (2018) to generate artificial new samples by reducing the "noise" levels of the original data. Specifically, given a d dimensional time series $x_{1:T} \in \mathbb{R}^{T\times d}$, let $x_{1:T} = U\Sigma V^T$ be its SVD, where $U \in \mathbb{R}^{T\times T}$, $V \in \mathbb{R}^{d\times d}$ and $\Sigma \in \mathbb{R}^{T\times d}$ a rectangular diagonal matrix with singular values $\{\sigma_1, \sigma_2, \cdots, \sigma_d\}$ on the diagonal in descending order such that r is the rank of $x_{1:T}$. Let m be the smallest index such that the explained varince $\frac{\sigma_m^2}{\sum_j \sigma_j^2}$ of the m'th mode is less than or equal to $1e^{-3}$. We define $\Sigma'=\{\sigma_1, \sigma_2, \cdots, u_1\sigma_m, u_2\sigma_{m+1}, \cdots u_{d-m+1}\sigma_d\}$ such that $u_i \overset{\text{i.i.d.}}{\sim} \mathbb{U}(0,1)$ and construct $x'_{1:T} = U\Sigma' V^T$. this process is repeated 64 times for each time series, generating 64 new time series.

D.3 Curvature estimation

There are several methods available for estimating curvature quantities of data representations, as discussed in papers such as Brahma et al. (2015); Shao et al. (2018). For our purposes, we have chosen to use the algorithm described in Li (2018), which is called Curvature Aware Manifold Learning (CAML). We opted for this algorithm because it is supported by theoretical foundations and is relatively efficient. In order to use CAML, we need to provide the neighborhood information of a sample and an estimate of the unknown ID. The ID is estimated using the TwoNN algorithm, as described in D.2, similarly to Ansuini et al. (2019); Kaufman & Azencot (2023).

In order to estimate the curvature of data $Y = \{y_1, y_2, \cdots, y_N\} \subset \mathbb{R}^D$, we make the assumption that the data lies on a d-dimensional manifold $\mathcal{M}$ embedded in $\mathbb{R}^D$, where d is much smaller than D. Consequently, $\mathcal{M}$ can be considered as a sub-manifold of $\mathbb{R}^D$. The main concept behind CAML is to compute a local approximation of the embedding map using second-order information.

$$f : \mathbb{R}^d \to \mathbb{R}^D \quad , y_i = f(x_i) + \epsilon_i \ , \quad i = 1, \ldots, N \ , \tag{2}$$

where $X = \{x_1, x_2, \cdots, x_N\} \subset \mathbb{R}^d$ are low-dimensional representations of Y, and $\{\epsilon_1, \epsilon_2, \cdots \epsilon_N\}$ are the noises. In the context of this paper, the embedding map f is the transformation that maps the low-dimensional dynamics to the sampled features for each time stamp t that might hold redundant information.

In order to estimate curvature information at a point $y_i \in Y$, we follow the procedure described above to define its neighborhood. This results in a set of nearby points $\{y_{i_1}, \ldots, y_{i_K}\}$, where K represents the number of neighbors. Using this set along with the point y_i, we utilize SVD to construct a local natural orthonormal coordinate frame $\left\{\frac{\partial}{\partial x^1}, \cdots, \frac{\partial}{\partial x^d}, \frac{\partial}{\partial y^1}, \cdots, \frac{\partial}{\partial y^{D-d}}\right\}$. This coordinate frame consists of a basis for the tangent space (first d elements) and a basis for the normal space. To be precise, we denote the projection of y_i and y_{i_j} for $j = 1, \ldots, K$ onto the tangent space spanned by $\partial/\partial x^1, \ldots, \partial/\partial x^d$ as x_i and u_{i_j} respectively. It is important to note that the neighborhood of y_i must have a rank of $r > d$. If the rank is less than d, then SVD cannot accurately encode the normal component at x_i, leading to poor approximations of f at x_i. Therefore, we verify that $\{y_{i_1}, \ldots, y_{i_K}\}$ has a rank of $d + 1$ or higher.

The map f can be expressed in the alternative coordinate frame as $f(x^1, \ldots, x^d) = [x^1, \ldots, x^d, f^1, \ldots, f^{D-d}]$. The second-order Taylor expansion of f^α at u_{i_j} with respect to x_i, with an error of $\mathcal{O}(|u_{i_j}|_2^2)$, is represented by

$$f^\alpha(u_{i_j}) \approx f^\alpha(x_i) + \Delta_{x_i}^T \nabla f^\alpha + \frac{1}{2}\Delta_{x_i}^T H^\alpha \Delta_{x_i} \ , \tag{3}$$

where $\alpha = 1, \ldots, D - d$, $\Delta_{x_i} = (u_{i_j} - x_i)$ and u_{i_j} is an elemnt in the neighborhood of x_i. The gradient of f^α is denoted by ∇f^α, and $H^\alpha = \left(\frac{\partial^2 f^\alpha}{\partial x^i \partial x^j}\right)$ is its Hessian. We have a neighborhood $\{y_{i_1}, \ldots, y_{i_K}\}$ of y_i,

and their corresponding tangent representations $\{u_{i_j}\}$. Using equation 3, we can form a system of linear equations, as explained in D.4. The principal curvatures are the eigenvalues of H^α, so estimating curvature information involves solving a linear regression problem followed by an eigendecomposition. Each Hessian has d eigenvalues, so each sample will have $(D-d)\times d$ principal curvatures. Additionally, one can compute the Riemannian curvature tensor using the principal curvatures, but this requires high computational resources due to its large number of elements. Moreover, as the Riemannian curvature tensor is fully determined by the principal curvatures, we focus our analysis on the eigenvalues of the Hessian. To evaluate the curvature of manifolds, we estimate the mean absolute principal curvature (MAPC) by taking the mean of the absolute values of the eigenvalues of the estimated Hessian matrices.

D.4 Estimating the Hessian Matrix

In order to estimate the Hessian of the embedding mapping f^α where $\alpha = 1, \ldots, D-d$, we build a set of linear equations that solves Eq. 3. We approximate f^α by solving the system $f^\alpha = \Psi X_i$, where X_i holds the unknown elements of the gradient ∇f^α and the hessian H^α. We define $f^\alpha = [f^\alpha(u_{i_1}), \cdots, f^\alpha(u_{i_K})]^T$, where u_{i_j} are points in the neighborhood of x_i, in the local natural orthogonal coordinates. The local natural orthogonal coordinates are a set of coordinates that are defined at a specific point p of the manifold. They are constructed by finding a basis for the tangent space and normal space at a point p by applying Principal Component Analysis, such that the first d coordinates (associated with the most significant modes, i.e., largest singular values) represent the tangent space, and the rest represent the normal space. We define $\Psi = [\Psi_{i_1}, \cdots, \Psi_{i_K}]$, where Ψ_{i_j} is given via

$$\Psi_{i_j} = \left[u^1_{i_j}, \cdots, u^d_{i_j}, \left(u^1_{i_j}\right)^2, \cdots, \left(u^d_{i_j}\right)^2, \left(u^1_{i_j} \times u^2_{i_j}\right), \cdots, \left(u^{d-1}_{i_j} \times u^d_{i_j}\right)\right] .$$

The set of linear equations $f^\alpha = \Psi X_i$ is solved by using the least square estimation resulting in $X_i = \Psi^\dagger f^\alpha$, where $X_i = \left[\nabla f^{\alpha 1}, \cdots, \nabla f^{\alpha d}, H^{\alpha 1,1}, \cdots, H^{\alpha d,d}, H^{\alpha 1,2}, \cdots, H^{\alpha d-1,d}\right]$. In practice, we estimate only the upper triangular part of H^α since it is a symmetric matrix. The gradient values ∇f^α are ignored since they are not required for the CAML algorithm. We refer the reader for a more comprehensive and detailed analysis in (Li, 2018).