Scaling Law for Time Series Forecasting

Jingzhe Shi[1]*, **Qinwei Ma**[1]*, **Huan Ma**[2], **Lei Li**[3,4] ✉

[1] Institute for Interdisciplinary Information Sciences, Tsinghua University
[2] Zhili College, Tsinghua University
[3] University of Copenhagen [4] University of Washington
[1,2] {shi-jz21,mqw21,mah21}@mails.tsinghua.edu.cn, [3,4] lilei@di.ku.dk

Abstract

Scaling law that rewards large datasets, complex models and enhanced data granularity has been observed in various fields of deep learning. Yet, studies on time series forecasting have cast doubt on scaling behaviors of deep learning methods for time series forecasting: while more training data improves performance, more capable models do not always outperform less capable models, and longer input horizons may hurt performance for some models. We propose a theory for scaling law for time series forecasting that can explain these seemingly abnormal behaviors. We take into account the impact of dataset size and model complexity, as well as time series data granularity, particularly focusing on the look-back horizon, an aspect that has been unexplored in previous theories. Furthermore, we empirically evaluate various models using a diverse set of time series forecasting datasets, which (1) verifies the validity of scaling law on dataset size and model complexity within the realm of time series forecasting, and (2) validates our theoretical framework, particularly regarding the influence of look back horizon. We hope our findings may inspire new models targeting time series forecasting datasets of limited size, as well as large foundational datasets and models for time series forecasting in future work.[2]

1 Introduction

Because of the practical value of time series forecasting, past years have seen rapid development for time series forecasting methods using the paradigm of neural network training. Neural Nets utilize different model architectures, including FFN-based [1, 2, 3], Transformer-based [4, 5, 6, 7, 8] and Convolution-based [9, 10] neural nets have been proposed. Starting from around 2022, some previous work [1, 11, 7, 10] proposed that powerful models could be enhanced by extending the look-back horizon because more historical information can be utilized. However, our experiments (Figure 3) show that **this claim may not hold for datasets in practice with a certain amount of training data**: optimal horizon does exist, and it will increase if the amount of available training data increases. This calls for a more thorough understanding of the impact of horizon and dataset size on forecasting loss.

In Natural Language Processing (NLP) [12, 13], Computer Vision (CV) [14] and other fields in deep learning, the impact of dataset size, model size and data granularity on performance is sometimes summarized as the Scaling Law: larger dataset, larger models and more detailed data granularity improves performance in these cases, and theories [15, 16] have been proposed to explain these behaviors. **However, these theories do not lay emphasis on the horizon of time series**, hence cannot be used directly to explain the impact of the horizon.

---

*Equal Contribution.

[2]Code for our experiments has been made public at: `https://github.com/JingzheShi/ScalingLawForTimeSeriesForecasting`.

In this work, we introduce a comprehensive scaling law theory for time series forecasting, with a particular emphasis on the impact of the horizon. This theory integrates dataset size and model size to optimize predictive performance based on the look-back horizon. We further conduct experiments to (1) verify that scaling behaviors for dataset size and model size do exist in time series forecasting and (2) validate our theory, especially about the influence of different horizons.

Our main contribution regarding time series forecasting includes:

1. We introduce a novel theoretical framework that elucidates scaling behaviors from an intrinsic space perspective, highlighting the critical influence of the look-back horizon on model performance. Our theory identifies an optimal horizon, demonstrating that beyond this point, performance degrades due to the inherent limitations of dataset size.

2. We conduct a comprehensive empirical investigation into the scaling behaviors of dataset size, model size, and look-back horizon across various models and datasets. Our research establishes a robust scaling law for time series forecasting, providing a foundational framework that adapts to diverse modeling contexts.

As a corollary to our conclusions, we point out that different models might have different optimal look-back horizon for the same dataset (Figure 1); therefore, **we call for future work to compare different models using the optimal look-back horizon for each model correspondingly rather than using a fixed look-back horizon**.

As a further result of our findings, though widely used in previous work, **to show a model benefits from longer horizon compared to baseline models is unnecessary for proving its superiority over these baseline models**.

We hope our work may inspire future work when designing forecasting models for specific datasets of limited size, as well as future work proposing large foundational datasets and models in the field of time series.

2 Related Work

2.1 Time Series Forecasting

The task of time series forecasting is to predict a time series with N variables in its next S frames (denoted as $Y = \{y_1, y_2, \ldots, y_N\} \in \mathbb{R}^{N \times S}$) given its previous observations with H frames (denoted as $X = \{x_1, x_2, \ldots, x_N\} \in \mathbb{R}^{N \times H}$). H is called look back horizon in some scenarios.

Channel-Independent model means to predict y_i by $\hat{y}_i = f(x_i)$. Linear models and MLPs have been proven to be effective learners for time series forecasting. A series of work [1, 3, 17] utilizes linear layers and methods like low-pass-filter and Ordinary Least Squares regression to learn linear relationships in the time series. Reversible MLP [18] proposes to use linear layers and MLPs with reversible Instance Norm [19] and obtains satisfying results. PatchTST [7] proposes to use patch embedding for time series.

Channel-Dependent model means to predict y by $\hat{y}_i = [f(x_1, x_2, \ldots, x_N)]_i$. A series of works based on Transformers and its variants have been proposed[4, 5, 6], as well as a series of convolution-based methods based on TCN[20, 21]. More recently, iTransformer [8] proposes to use attention to capture the relationship between different variables. ModernTCN [10] proposes new convolutional architectures enabling a wide Effective Receptive Field.

There have been many works analyzing on mathematical properties of time series before machine learning exists. [22, 23] give good summaries of the early works from different perspectives.

Recently, there have been works proposing large foundational datasets and models for time series. Some propose foundational models that can do zero-shot forecasting[24]. Some propose open-source foundational datasets and verify the ability for transfer learning of foundation models trained on the datasets[25, 26].[3] There are also works utilizing LLMs to do zero-shot prediction[28], or use LLM backbones pretrained on text contexts to perform time series forecasting[29, 30, 31].

---

[3]Released in close timing with ours, [27] verifies scaling law experimentally for the classic transformer architecture on a large mixed dataset for time series forecasting.

2.2 Scaling Law and related Theory

A plethora of research has been conducted to investigate the scaling law in various domains of deep learning, encompassing Natural Language Processing, Computer Vision tasks, and Graph-based Neural Networks [12, 13, 14, 32]. These studies have not only observed the existence of scaling law but also proposed theories to elucidate them. For instance, certain work has interpreted the scaling behavior from the standpoint of intrinsic space or data manifold, providing a deeper understanding of the underlying mechanisms [15, 16]. In parallel, the properties of time series have been the subject of extensive theoretical and empirical studies. Notably, some research has established bounds for the quantization error of time series, contributing to the body of knowledge on time series analysis [33].

3 Theory for Scaling Law in Time Series Forecasting

3.1 Forecasting in Intrinsic Space

To represent the amount of information carried by a time series slice of length L, we consult the concept of intrinsic dimension and intrinsic space. Consider all time series slices of length L in a particular scenario, these slices are hypothesized to reside within an intrinsic space that encapsulates the fundamental characteristics of the data, effectively representing its inherent features. Denote this intrinsic space as $\mathcal{M}(L)$ and its intrinsic dimension as $d_I(L)$.

It immediately follows that **time series forecasting is equivalent to predicting a vector in $\mathcal{M}(S)$ given its previous H frames, which can be represented by a vector in the space $\mathcal{M}(H)$.**

3.2 Formulation

Before studying the impact of horizon, dataset size and model on loss, we first formulate the intrinsic space, the data distribution and the form of the loss.

3.2.1 Intrinsic Space

For a time series $s_{0,1,\dots,L}$ of length L (each s_i is a unit element in the sequence that may contain single or multiple variables, dependent or independent), we represent it using a vector x in $\mathcal{M}(L)$.

We make these assumptions on the spaces $\{\mathcal{M}(1), \mathcal{M}(2), \dots\}$ and the distribution of data in the spaces:

1. **Information-preserving**: Intuitively speaking, we should be able to recover the real sequence (which might be a multivariable sequence or a singlevariable sequence) from its corresponding intrinsic vector with the error bounded by a small constant value. Formally we can state this as follows:

    Exists a mapping ϕ from the original length-L sequence space $\mathcal{O}(L)$ to $\mathcal{M}(L)$, an inverse mapping $\phi^{-1}: \mathcal{M}(L) \to \mathcal{O}(L)$ and a constant $e \ll 1$ related to L so that for any $\mathbb{E}_{x \sim \mathcal{O}(L)}$, $\|x - \phi^{-1}(\phi(x))\|_2^2 \le e(L)$.

2. **Inverse Lipschitz**: ϕ^{-1} should be K_I-Lipschitz under L2 norm. That is:

$$\forall x, y \in \mathcal{M}(L), \|\phi^{-1}(x) - \phi^{-1}(y)\|_2 \le K_I \|x - y\|_2$$

3. **Bounded**: For simplicity, we assume the values in all dimensions of the intrinsic space are bounded, and thus we can scale the intrinsic space to fit it into $\mathcal{M}(H) = [0, 1]^{d_I(H)}$.

4. **Isomorphism**: $\mathcal{M}(L_1)$ is isomorphic to a subspace of $\mathcal{M}(L_2)$ for $L_1 \le L_2$. Moreover, the isometry should also preserve the data distribution in the space. Formally, we can state it as:

    Let $P[H_1, H_2]$ denote the linear projection from $\mathcal{M}(H_1)$ to the subspace of it isomorphic to $\mathcal{M}(H_2)$, and let Cov_L denote the covariance matrix of data distribution in $\mathcal{M}(L)$, then Cov_{L_1} should be congruent to $P[L_2, L_1](Cov_{L_2})$ for any $L_1 < L_2$.

5. **Linear Truncation**: Truncation in time series space is close to linear projection mapping in $\mathcal{M}(L)$. Formally, we can state it as:

Define a truncating function $t_p[H_1, H_2](x_{0:H_1})$, so that for any sequence s, let $s_{h_1:h_2}$ denote the intrinsic vector for the subsequence from $s[h_1]$ to $s[h_2 - 1]$, then $t_p[H_1, H_2](s_{0:H_1}) = s_{0:H_2}$, then $t_p[H_1, H_2] \approx P[H_1, H_2]$.

6. **Causality**: There should be an optimal model to predict the next S frames given the previous $h \to \infty$ frames, so that the error only originates from the intrinsic noise. That is:

$$\exists F[S] : \mathcal{M} \to \mathcal{M}(S), s.t. \lim_{h \to \infty} \mathbb{P}(y \mid x_{-h:0}) = (1-\eta)\delta(F[S](x_{-h:0})) + \eta\mathcal{N}(F[S](x_{-h:0}), \Sigma_S),$$

where η stands for the noise ratio in the system, $\delta(\cdot)$ stands for the Dirichlet function and $\mathcal{N}(\mu, \Sigma)$ stands for a normal distribution with mean μ and covariance Σ. (Notice that the noise distribution is not necessarily a normal distribution and our result holds for any noise distribution with mean μ and covariance σ. However we only consider the normal distribution case here for simplicity.)

Moreover, we assume $F[S]$ is first-order K_1-Lipschitz and second-order K_2-Lipschitz in L_2 metric (as we assume S is fixed we don't discuss how the Lipschitz coefficients vary with S and simply take them as constant).

7. **Uniform Sampling Noise**: When drawing a length-L sample, we assume the sampling noise is uniform in each direction in $\mathcal{M}(L)$.

8. **Zip-f Distribution**: We assume the data distribution in the intrinsic space follows a Zip-f law on different dimensions of the intrinsic space. That is, the eigenvalue spectrum of Cov_L satisfies $\lambda_i \approx \lambda_0 i^{-\alpha_Z}$ where λ_i represents the i-th largest eigenvalue. This is shown by other work [34, 35] and also verified in our experiments.

This result is asymptotic and does not suggest uniform intrinsic dimensions across sequence elements. Specifically, for a sequence element with v variables, $d_I(H) \propto v$ approximately in channel-independent scenarios. In contrast, for channel-dependent cases, which are more common, the total intrinsic dimension is typically less than the sum of dimensions for individual variables.

In the deduction part we do not assume the specific relationship between $d_I(H)$ and H. Some previous work[36] shows that in some cases $d_I(H) \approx \Theta(H)$. In Appendix A.1 we discuss more about their relationship.

We formulate the intrinsic space and these assumptions more strictly and provide a brief construction of $\{\mathcal{M}(1), \mathcal{M}(2), \ldots\}$ in Appendix A.2. Also, we discuss cases where these conditions are not strictly satisfied in Appendix A.3.

Moreover, under these assumptions made, the loss in the original space L_{ori} can be linearly bounded by the loss in the intrinsic space L_{ins}, details of which can be found at Appendix A.4.

In the following deduction, we use L to denote L_{ins}.

3.2.2 Loss: Overall

In the following sections, we consider the task to predict a vector in $\mathcal{M}(S)$ given the vector corresponding to its previous H frames in $\mathcal{M}(H)$ with a model m. For simplicity, we represent the operation of obtaining a vector in $\mathcal{M}(t_2 - t_1)$ by truncating the sequence s from time t_1 to t_2 as $x[t_1 : t_2]$, where x is a representation in $\mathcal{M}(|s|)$.

$$L = E_{x \sim \mathcal{M}(H+S)}[(x[H : H + S] - m(x[0 : H]))^2]$$

Let m^* denote the optimal Bayesian model, then it should satisfy:

$$m^*(x[0 : H]) = E_{x \sim \mathcal{M}(H+S)}[x[H : H + S]|x[0 : H]]$$

Thus:

$$L = E_{x \sim \mathcal{M}(H+S)}[(x[H : H + S] - m^*(x[0 : H]) + m^*(x[0 : H] - model(x[0 : H]))^2]$$
$$= E_{x \sim \mathcal{M}(H+S)}[(x[H : H + S] - m^*(x[0 : H])^2] + E_{x \sim \mathcal{M}(H)}[(m^*(x) - m(x))^2]$$
$$+ 2 * E_{x \sim \mathcal{M}(H+S)}[(x[H : H + S] - m^*(x[0 : H]) * (m^*(x) - m(x))]$$

4

Since $x[H:H+S]-m^*(x[0:H])$ and $m^*(x)-m(x)$ are independent with respect to $s[H:H+S]$ and $E_{x\sim\mathcal{M}(H+S)}[x[H:H+S]-m^*(x[0:H])]=0$, the loss is a sum of the previous two terms: one is decided by the capability of the optimal Bayesian model (or the Bayesian Error), and the other is the model's ability to approximate the Bayes Estimation: $L=L_{Bayesian}+L_{approx}$.

We then calculate each of the two terms corresponding to the horizon, dataset size and model size.

3.2.3 Bayesian Loss

We consider the Bayesian error for $\mathcal{M}(\infty)$, when the loss for predicting S frames originates from the inherent uncertainty of the system. That is, we evaluate the amount of information carried by $\mathcal{M}(H)$ compared to $\mathcal{M}(\infty)$. By assumption 5 and 6 in section 3.2.1, It can be verified that:

$$L_{Bayesian} \leq (1-\eta)K_1^2\mathbb{E}[var(P^{-1}[\infty,H](x))]+\eta\cdot tr(\Sigma_S)$$

According to assumption 7, the noise in the i-th predicted frame would be proportional to $\sqrt{i}$, and the total noise in the predicted S frames would be proportional to S. Let σ_M^2 denote the variance of the noise for a single frame in a single dimension, then $tr(\Sigma_S)$ should be equal to $\sigma_M^2 S^2 d_I(S)$.

According to assumptions 4 and 8, we can express the inverse projection term into:

$$\mathbb{E}[var(P^{-1}[H](x))] = \sum_{d_I(H)\leq i<d_I}\lambda_i \approx \frac{\lambda_0}{(\alpha_Z-1)d_I(H)^{\alpha_Z-1}}$$

which indicates that:

$$L_{Bayesian} \approx K_1^2(1-\eta)\frac{\lambda_0}{(\alpha_Z-1)d_I(H)^{\alpha_Z-1}}+\eta\sigma_M^2 S^2 d_I(S)$$

3.2.4 Approximation Loss: Two Cases

The training data is sampled from the distribution of $E_{x\sim\mathcal{M}(H+S)}[x[H:H+S]|x[0:H]]$. Following previous works [16, 37], we utilize the piece-wise linear assumption for deep learning models. That is, we assume the model partitions the intrinsic space into N subregions, and does a linear prediction for each subregion independently. Here we discuss two cases: the large-dataset limit and the small-dataset limit.

If the dataset is large and contains many samples in the piece of the model, the model tends to learn the averaged behavior of the region. Intuitively, a larger dataset size brings smaller noise averages, and a larger model brings smaller piece-wise linear pieces, reducing the error caused by deviation of the data distribution to the linear model in the small blocks. Roughly speaking, the loss consists of two terms: one caused by the uncertainly within the subregions partitioned by the model L_r, and one caused by the noise in the data that makes the model fail to learn the optimal model L_n. If we assume that the model partitions the space uniformly, then the loss should satisfy:

$$L_{approx} \approx K_2^2\frac{d_I(H)^2 N^{-\frac{4}{d_I(H)}}}{4\pi^2}+\frac{Nd_I(H)}{D}(\sigma_M^2 S^2 d_I(S)+\frac{K_1^2\lambda_0}{(\alpha_Z-1)d_I(H)^{\alpha_Z-1}}).$$

The uniform partitioning should be the most naive case for the model to learn, and we also analyze on the cases where more advanced partitionings are learned in Appendix D. Also, it is worth noticing that the noise consists of two term, one is the systematic uncertainly as shown in section 3.2.3, and the other is caused by the horizon limitation so that the effect of unseen dimension seems exactly like noise for the model. Please refer to Appendix B.2 for a more detailed derivation.

Otherwise if there are few data samples, or the model is sufficiently large and the model cannot learn to average the noises of closed samples, but rather learns to remember each sample in certain pieces. This would give a data-scaling loss determined by nearest-neighbor distance. In this case:

$$L_{approx} \approx \frac{K_1^2}{4\pi}d_I(H)D^{-\frac{2}{d_I(H)}}.$$

Please refer to Appendix B.3 for a more detailed derivation of this scenario.

Boundary of the two phases is decided by Dataset size and Model size, as well as horizon. It is worth noticing that in time series forecasting tasks, the sliding window method for constructing

data samples is usually used. [4] Therefore, the closest data points are always dependent on each other and are strongly correlated. The effective number of mutually independent data fragments is approximately proportional to D/H rather than D, and $\xi = D/NH$ would be a very approximate order parameter separating the previous two scenarios. Again, please refer to Appendix B for a more detailed derivation.

3.3 Optimal Horizon under Each Circumstance

As stated in section 3.2.2, the total loss would be simply the sum of the two components deduced above. Analyzing the total loss form, we may achieve an optimal horizon H^* (or a corresponding optimal intrinsic dimension d_I^* that minimizes the loss for different cases.

3.3.1 Optimal Horizon for large amount of data

For the case with sufficient data, the total loss is:

$$
\begin{aligned}
L =& L_{Bayesian} + L_{Approx} \\
\approx& K_2^2 \frac{d_I(H)^2 N^{-\frac{4}{d_I(H)}}}{4\pi^2} + K_1^2(1-\eta)\frac{\lambda_0}{\alpha_Z-1}\frac{1}{d_I(H)^{\alpha_Z-1}} \\
&+ \frac{N d_I(H)}{D}(\sigma_M^2 S^2 d_I(S) + \frac{K_1^2 \lambda_0}{(\alpha_Z-1)d_I^{\alpha_Z-1}(H)}) \\
\approx& K_2^2 \frac{d_I(H)^2 N^{-\frac{4}{d_I(H)}}}{4\pi^2} + K_1^2(1-\eta)\frac{\lambda_0}{\alpha_Z-1}\frac{1}{d_I(H)^{\alpha_Z-1}} + \frac{N d_I(H)}{D}\sigma_M^2 S^2 d_I(S) \\
&\text{(since we always assume } N \ll D)
\end{aligned}
$$

We consider two cases. The detailed derivation can be found in Appendix C.1.

If model size is too small compared to dataset size such that $N = o(D^{\frac{d_I(H)}{d_I(H)+4}})$, then the effect of dataset size on picking optimal horizon could be neglected, thus:

$$
d_I^* = \mathcal{W}(\frac{4}{\alpha_Z C_0^{\frac{1}{\alpha_Z}}} \ln^{1+\frac{1}{\alpha_Z}} N) \approx \frac{4}{\alpha_Z C_0^{\frac{1}{\alpha_Z}}} \ln^{1+\frac{1}{\alpha_Z}} N.
$$

where $\mathcal{W}(\cdot)$ is the Lambert W function ($\mathcal{W}(x) \approx x$) and $C_0 = \frac{K_1^2 \pi^2 (1-\eta)\lambda_0}{K_2^2}$.

If N is not that small: $N = \omega(D^{\frac{d_I(H)}{d_I(H)+4}})$. Then the noise effect would be dominant in picking optimal H, and the optimal $d_I(H)$ would be:

$$
d_I^* = (\frac{K_1^2(1-\eta)\lambda_0 D}{N\sigma_M^2 S^2 d_I(S)})^{\frac{1}{\alpha_Z}}.
$$

In this case, d_I^* grows noticeably with the increment of D and the decrement of N.

3.3.2 Optimal Horizon for a relatively small amount of data

If data is scarce, the loss could be written as:

$$
\text{loss} \approx K_1^2(1-\eta)\frac{\lambda_0}{\alpha_Z-1}\frac{1}{d_I(H)^{\alpha_Z-1}} + \frac{K_1^2}{4\pi}d_I(H)D^{-\frac{2}{d_I(H)}}
$$

It is worth noticing that in this case, we assume the model is always large enough to find the nearest neighbor for a test sample, hence the loss is irrelevant with N. We can estimate the optimal d_I^* as:

$$
d_I^* = C_s \frac{\ln D}{\ln \ln D}
$$

Where C_s is a constant irrelevant with D, N or H. The exact form is provided in Appendix C.2. Compared to the first scenario, the optimal d_I changes much less in this scenario.

---

[4]Data points between time [t-H:t+S] is considered one sample, and [t-H+1:t+S+1] is considered another sample in the dataset. These two samples are strongly correlated with each other.

4 Experiment Results

4.1 Scaling Law for dataset size and model width do exist for Time Series Forecasting
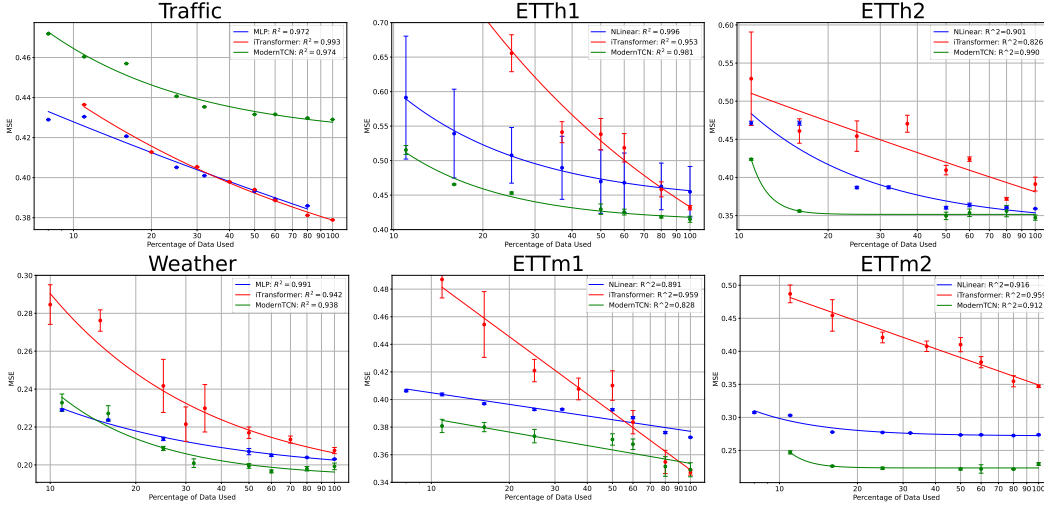

Figure 1: Data Scaling. The proposed formula $loss(D) = A + B/D^\alpha$ fits well. More comparison with other formulas can be found at Appendix I.
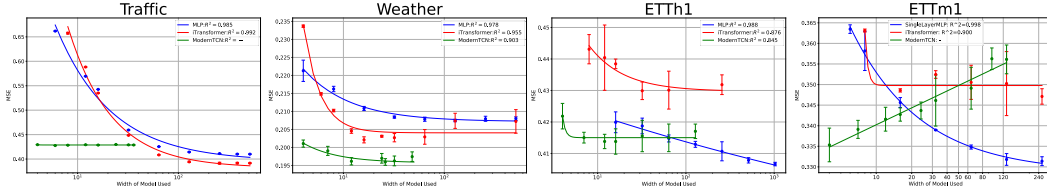

Figure 2: Width Scaling. When the model is not powerful enough, $loss(W) = A + B/W^\alpha$ fits well for these situations. When data is scarce, a large model may lead to overfitting, as observed with ModernTCN on ETTm1.

As depicted in Figure 1 and Figure 2, we corroborate the scaling behaviors pertaining to data scaling and model-size scaling across a diverse range of datasets and models. This validation underscores the robustness and versatility of our proposed theoretical framework in the context of time series forecasting.

Here we mainly include NLinear[1]/MLP, ModernTCN[10] and iTransformer[8] as our models, covering a scenario of Channel-Independent and Channel-Dependent, FFN-based, Transformer-based and Convolution-based methods. For datasets, we mainly use ETTh1, ETTh2, ETTm1, ETTm2, Traffic and Weather[4, 5] as our datasets. Detailed experiment settings can be found at Appendix E.

It can be seen that for all these models on all these datasets, for the **dataset-scaling** case where $loss = C_D + 1/D^{\alpha_D}$ and the **model width-scaling** with large amount of data: $loss = C_W + 1/W^{\alpha_W}$. The results fit well, thus verifying the existence of the original understanding of the scaling law.

However, in some special cases when the model is large enough to approximate the data, increasing model size would not gain performance, and may hurt performance (if regularization that is not strong enough is added) in some cases, like ModernTCN for ETTm1.

4.2 The Impact of Horizon to Final Loss

4.2.1 Optimal Horizon and Training Data Amount

Optimal Horizon grows with more available training data. We conduct experiments, fixing the available training data amount and model size while adjusting the horizon. As shown in Figure 3, an

optimal horizon is observed at each data amount, and this optimal value grows with more available training data.
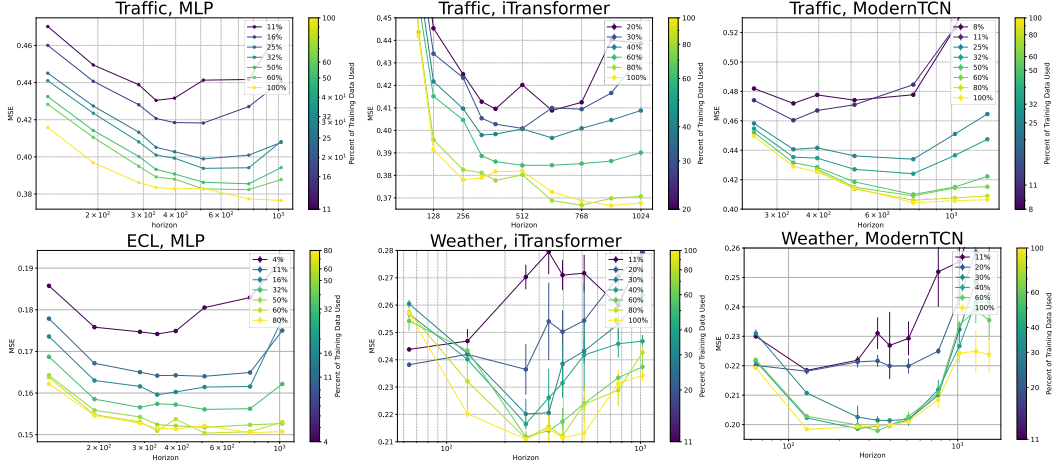

Figure 3: Loss v.s. Horizon for a certain amount of training data, for different datasets and different models.

A dataset with a larger feature degradation has a smaller optimal horizon. As our theory predicts, larger α_z leads to a smaller optimal horizon. In the experiment in Figure 4, we use the same Linear model on two different datasets with comparable amount of training data: the Exchange Dataset has 70% available training data compared to the ETTh1 Dataset. We use eigenvalues obtained by doing Principal component analysis on sampled series as an approximation to the feature variance in the intrinsic space. Detailed procedure and more PCA results can be found at Appendix G.
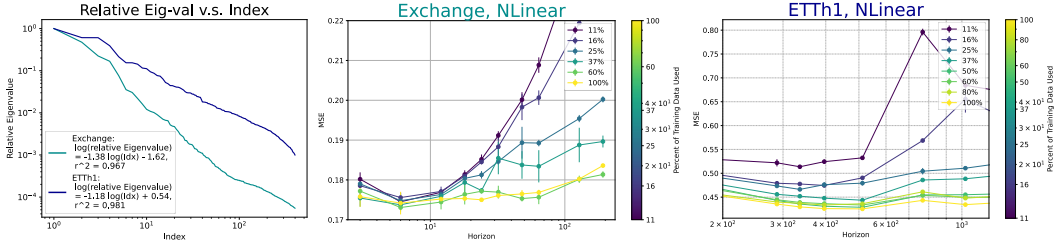

Figure 4: PCA results under Channel-Independent and Instance Normalization setting(left), Loss v.s. Horizon for certaim amount of training data on Exchange(middle) and ETTh1(right). Exchange dataset has 70% data points compared to ETTh1 for training. However, since its feature degradation is stronger, the optimal horizon (< 30) using 100% of Exchange dataset is much smaller than the optimal horizon of the ETTh1 dataset (> 300) with only 11% of available training data.

Channel-Dependent and Channel-Independent Models sometimes are in different states of the two cases. For CD models, $d_I(H)$ is larger and less training data is available, hence it tends to be in the few-data limit. For CI models, $d_I(H)$ is smaller and D is larger, hence it may reach the data-dense limit (where the scaling exponent for D is -1).

In the following Figure 5, iTransformer on Traffic dataset is in the data-dense limit. For MLP on Transformer, when the horizon is small it is in the few-data limit. For the Linear model, since it is simply linear (rather than piece-wise linear), we expect it to be within the data-dense limit even for long horizons and when the dataset is relatively small (like ETTh1).

The advantage of channel-independent and channel-dependent models can be explained from the perspective of our theory. For channel-dependent models, the horizon limitation is smaller hence it performs better with plenty of training data. For channel-independent models, more training data is available; moreover, d is smaller, making the scaling exponent larger.
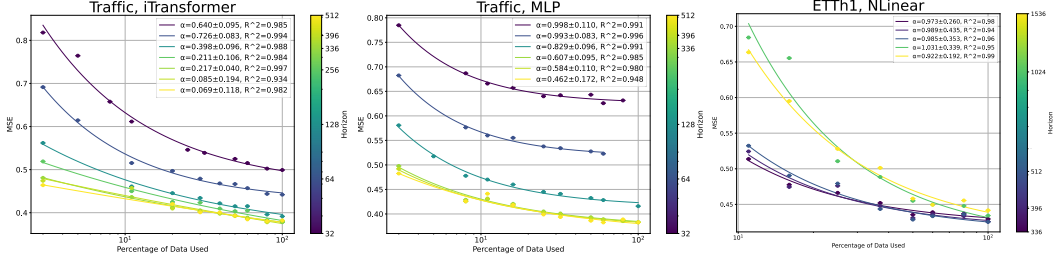
Figure 5: Data scaling behavior for iTransformer (Channel-Dependent model, left) and Norm-MLP(Channel-Independent model, middle) and NLinear (CI, right).

4.2.2 Optimal Horizon v.s. Model Size

As predicted by our theory there are **two cases**. **(1) When** N **is small**, the optimal horizon does not change much with N. **(2) When** N **is large**, the model size scaling term no longer dominates, the coefficient of the noise term $\frac{Nd_I}{D}$ dominants thus larger N leads to smaller optimal H.
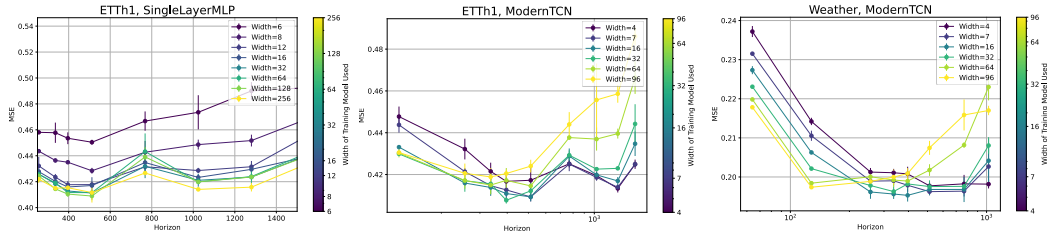

Figure 6: Loss v.s. Horizon for models of different widths. For MLP on ETTh1 (left), ModernTCN on ETTh1(middle) and ModernTCN on Weather (right).

From the observations in Figure 6, it can be discerned that in the initial scenario, where an enhancement in N results in a performance augmentation, the optimal horizon remains relatively invariant with respect to N. This is exemplified in the case of the SingleLayerMLP model. However, in the second scenario, where N attains its optimal value (i.e., for a certain horizon, a smaller N surpasses a larger N in performance), a larger N will correspond to a reduced optimal horizon. This phenomenon is evident in the instances involving the ModernTCN model.

As predicted by our theory, we see that dataset size has an impact on optimal horizon, while model size has a less significant influence on it.

5 Discussion and Conclusion

5.1 Limitation and Discussion

Our theory mainly covers the part of time series forecasting, and our experiments verify our proposed theory on some of the well-used datasets of various sizes. However, these datasets are still small compared to some of the recently proposed large datasets[25]. The scaling behavior predicted in our work of the horizon on these large datasets remains to be experimentally verified and studied. Moreover, we mainly use popular time series forecasting models in recent work and these models might be over-designed for a few datasets for time series.

Moreover, our work mainly focus on the in-domain setting, rather for pretrained-then-finetuned models or foundation models trained on mixed datasets. We discuss more about whether the assumptions made and our theory deduction work for mixed datasets at Appendix H.

Since we focus on the impact of the horizon which is an adjustable hyper-parameter for forecasting tasks but a fixed hyper-parameter for other time series tasks, this work does not involve scaling behaviors for other tasks related to time series analysis.The theory for scaling law in other areas of time series forecasting as well as for foundation time series forecasting models remains to be studied.

Also, although our theory is compatible with analyzing the effect of the predict length S and current experiment for fixed S verifies the impact of H, it is still worth studying the impact of S on optimal H. Also, our theory only holds with assumptions made in Section 3. We leave these to future work. [5]

We call for future work to compare different models by 'performance at optimal look back horizon' rather than 'performance at a certain look back horizon' to improve robustness. This work further elucidates that, though used in a lot of previous work, **to show a model benefits from longer horizon compared to baseline models does not necessarily prove its superiority over these baseline models**.

5.2 Conclusion

In this work, we propose a theoretical framework that contemplates the influence of the horizon on scaling behaviors and the performance of time series forecasting models. We take into account the size of the dataset, the complexity of the model, and the impact of the horizon on the ultimate performance. An expanded horizon results in a diminished Bayesian Error, but it simultaneously complicates the task for a limited dataset to fully encompass the entire space and for a model of restricted size to learn effectively. Furthermore, our empirical investigations corroborate the existence of scaling behaviors in relation to dataset size and model size, and validate our proposed theory concerning the impact of the horizon.

Our theory provides some insight into the area of time series forecasting. For a certain dataset, it would be beneficial to design the models and hyperparameters according to the dataset size and feature degradation property of the particular dataset. Moreover, we think further experiments on larger foundational time series dataset about the optimal horizon with respect to pretraining loss and the loss for transferring to certain datasets may provide further insight as well.

In conclusion, we aim to provide insights to future work on time series forecasting, emphasizing the importance of the horizon and its potential impact on scaling behaviors in time series forecasting tasks.

---

[5]Meanwhile, NLP and CV tasks feature long-range dependency (low degradation with horizon), large datasets, hence more detailed data granularity (e.g., context length, image resolution) produces better results in nowadays' work. However, since the community is using larger and larger models and data granularity (context length, etc), but the amount of training data is limited, it is nature to raise such question: will we see similar patterns in time series forecasting that more detailed data granularity may hurt performance in NLP or CV tasks in the future? This is also an interesting question for future work.

References

[1] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?" 2022.

[2] S.-A. Chen, C.-L. Li, N. Yoder, S. O. Arik, and T. Pfister, "Tsmixer: An all-mlp architecture for time series forecasting," 2023.

[3] Z. Xu, A. Zeng, and Q. Xu, "Fits: Modeling time series with $10k$ parameters," 2024.

[4] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," 2021.

[5] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," 2022.

[6] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting," 2022.

[7] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," in *International Conference on Learning Representations*, 2023.

[8] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, and M. Long, "itransformer: Inverted transformers are effective for time series forecasting," 2024.

[9] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "Timesnet: Temporal 2d-variation modeling for general time series analysis," 2023.

[10] L. donghao and wang xue, "ModernTCN: A modern pure convolution structure for general time series analysis," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=vpJMJerXHU

[11] H. Wang, J. Peng, F. Huang, J. Wang, J. Chen, and Y. Xiao, "Micn: Multi-scale local and global context modeling for long-term series forecasting," 2023.

[12] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, "Scaling laws for neural language models," 2020.

[13] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. de Las Casas, L. A. Hendricks, J. Welbl, A. Clark, T. Hennigan, E. Noland, K. Millican, G. van den Driessche, B. Damoc, A. Guy, S. Osindero, K. Simonyan, E. Elsen, J. W. Rae, O. Vinyals, and L. Sifre, "Training compute-optimal large language models," 2022.

[14] X. Zhai, A. Kolesnikov, N. Houlsby, and L. Beyer, "Scaling vision transformers," 2022.

[15] Y. Bahri, E. Dyer, J. Kaplan, J. Lee, and U. Sharma, "Explaining neural scaling laws," 2024.

[16] U. Sharma and J. Kaplan, "A neural scaling law from the dimension of the data manifold," 2020.

[17] W. Toner and L. Darlow, "An analysis of linear time series forecasting models," 2024.

[18] Z. Li, S. Qi, Y. Li, and Z. Xu, "Revisiting long-term time series forecasting: An investigation on linear mapping," 2023.

[19] T. Kim, J. Kim, Y. Tae, C. Park, J.-H. Choi, and J. Choo, "Reversible instance normalization for accurate time-series forecasting against distribution shift," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=cGDAkQo1C0p

[20] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018.

[21] R. Sen, H.-F. Yu, and I. Dhillon, "Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting," 2019.

[22] E. Parzen, "An Approach to Time Series Analysis," *The Annals of Mathematical Statistics*, vol. 32, no. 4, pp. 951 – 989, 1961. [Online]. Available: https://doi.org/10.1214/aoms/1177704840

[23] J. G. De Gooijer and R. J. Hyndman, "25 years of time series forecasting," *International journal of forecasting*, vol. 22, no. 3, pp. 443–473, 2006.

[24] A. Garza, C. Challu, and M. Mergenthaler-Canseco, "Timegpt-1," 2024.

[25] Y. Liu, H. Zhang, C. Li, X. Huang, J. Wang, and M. Long, "Timer: Transformers for time series analysis at scale," 2024.

[26] A. Das, W. Kong, R. Sen, and Y. Zhou, "A decoder-only foundation model for time-series forecasting," 2024.

[27] T. D. P. Edwards, J. Alvey, J. Alsing, N. H. Nguyen, and B. D. Wandelt, "Scaling-laws for large time-series models," 2024. [Online]. Available: https://arxiv.org/abs/2405.13867

[28] N. Gruver, M. Finzi, S. Qiu, and A. G. Wilson, "Large language models are zero-shot time series forecasters," 2023.

[29] C. Chang, W.-Y. Wang, W.-C. Peng, and T.-F. Chen, "Llm4ts: Aligning pre-trained llms as data-efficient time-series forecasters," 2024.

[30] T. Zhou, P. Niu, X. Wang, L. Sun, and R. Jin, "One fits all:power general time series analysis by pretrained lm," 2023.

[31] M. Jin, S. Wang, L. Ma, Z. Chu, J. Y. Zhang, X. Shi, P.-Y. Chen, Y. Liang, Y.-F. Li, S. Pan, and Q. Wen, "Time-llm: Time series forecasting by reprogramming large language models," 2024.

[32] J. Liu, H. Mao, Z. Chen, T. Zhao, N. Shah, and J. Tang, "Neural scaling laws on graphs," 2024.

[33] P. Zador, "Asymptotic quantization error of continuous signals and the quantization dimension," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 139–149, 1982.

[34] N. Levi and Y. Oz, "The underlying scaling laws and universal statistical structure of complex datasets," 2024.

[35] S. Petrini, A. Casas-i Muñoz, J. Cluet-i Martinell, M. Wang, C. Bentz, and R. Ferrer-i Cancho, "Direct and indirect evidence of compression of word lengths. zipf's law of abbreviation revisited," *Glottometrics*, vol. 54, p. 58–87, 2023. [Online]. Available: http://dx.doi.org/10.53482/2023_54_407

[36] T. M. Buzug, J. von Stamm, and G. Pfister, "Characterising experimental time series using local intrinsic dimension," *Physics Letters A*, vol. 202, no. 2-3, pp. 183–190, 1995.

[37] T. N. Chan, Z. Li, L. H. U, and R. Cheng, "Plame: Piecewise-linear approximate measure for additive kernel svm," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 10, pp. 9985–9997, 2023.

[38] K. D. Ba, P. Indyk, E. Price, and D. P. Woodruff, "Lower bounds for sparse recovery," in *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*. SIAM, 2010, pp. 1190–1197.

[39] X. Qiu, J. Hu, L. Zhou, X. Wu, J. Du, B. Zhang, C. Guo, A. Zhou, C. S. Jensen, Z. Sheng, and B. Yang, "Tfb: Towards comprehensive and fair benchmarking of time series forecasting methods," 2024.

[40] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," 2019.

[41] H. Liu, Z. Dai, D. R. So, and Q. V. Le, "Pay attention to mlps," 2021.

A   More about the intrinsic space

A.1   Relationship between intrinsic dimension and horizon

We claimed in section 3.2.1 that $d_I(H)$ should be asymptotically linear to H. Here we give a rough explanation for this. Since the reconstruction error is bounded by a constant value irrelevant to L, the relative error should be $O(\frac{1}{L})$. That is, the error reconstructing $y \in \mathcal{O}(L)$ from $x \in \mathcal{M}(L)$ should be $O(\frac{1}{L})$. On the hand, this error can be viewed as an error caused by compressing a $d_O(L)$-dimensional vector into a $d_I(L)$-dimensional space. From a sparse recovery aspect, if the covariance matrix in $\mathcal{O}(L)$ has rank r, then the $d_O(L)$-dimensional vector would be equivalent with a r-sparse vector. [38] shows that, in this case, the minimal dimension that could recover it with relative error bounded by C is $\tilde{\Omega}(r)$. Since in our case $r \propto L$ due to intrinsic uncertainty, hence $d_I(L) = \Omega(L)$. And since $d_I(L) \le d_O(L) = O(L)$, we proved that $d_I(L) = O(L)$.

A.2   A Construction Method

We informally provide a specified method to construct $\mathcal{M}(L)$ theoretically in a recursion method. If $\mathcal{M}(0), \mathcal{M}(1), \dots, \mathcal{M}(L-1)$ has been defined, then we define $\mathcal{M}(L)$ as follows: given a time series $x_{0,1,\dots,L-1}$ of length L, we represent it by concatenating the representation of $x_{0,1,\dots,L-2}$ in $\mathcal{M}(L-1)$ and x_{L-1} in a space with dimension $dim(\mathcal{M}(L-1)+1)$. Then we find a manifold that these points lie in in this space to represent $\mathcal{M}(L)$.

A.3   Non-linear truncation

We assumed that the truncating function is close to linear projection in section 3.2.1. But in fact the two functions might have some subtle differences. This is because for a certain feature, estimating the feature with a sequence of finite length L might result in error for the feature even if each element in the sequence is accurate enough. (For example, if the feature comes directly from DFT (Discrete Fourier Transform), there might be 'leakage error' if the base frequency is not an exact multiple of the feature frequency.) This is a systematic error caused by the 'precession', where the feature's characteristic frequency mismatches the measuring frequency.

For any sequence S and let $x_{t_1:t_2} \in \mathcal{M}(t_2 - t_1)$ denote the vector for the sequence $S[t1 : t2]$, we can define a 'truncating function' $t_p[H_1, H_2] : \mathcal{M}(H_1) \to \mathcal{M}(H_2)$ for any $H_1 \ge H_2 > 0$ with $t_p[H_1, H_2](x_{0:H_1}) = x_{0:H_2}$. Let $t_p^{-1}[H_1, H_2]$ denote the inverse image function, which maps $x \in \mathcal{M}(H_2)$ to $\{y \mid y \in \mathcal{M}(H_1), t_p[H_1, H_2](y) = x\}$. Notice that this truncating function is unrelated to the specific sequence. Now, assume that $t_p[H_1, H_2]$ is smooth, then from implicit function theorem we know that for any vector $x \in \mathcal{M}(H_1)$ if $t_p[H_1, H_2]$ is locally differentiable in a neighborhood of x then $t_p^{-1}[H_1, H_2](t_p[H_1, H_2](x))$ is a submanifold of $\mathcal{M}(H_1)$ with dimension $d_I(H_1) - d_I(H_2)$, and from Sard's theorem we know that the targets of these locally undifferentiable points have measure zero. That is:

Let $S = \{x \mid x \in \mathcal{M}(H_1), t_p^{-1}[H_1, H_2](x) \notin M^{d_I(H_1)-d_I(H_2)}\}$ where M^d denotes the set of all d-dimensional manifolds. Let μ denote a measuring function defined in $\mathcal{M}(H_1)$, then $\mu(S) = 0$.

Hence the impact of such 'critical points' to the total loss is negligible. For simplicity, we can assume $S = \emptyset$. Then t_p could be written as:

$$t_p[H_1, H_2](x) = P[H_1, H_2]g[H_1, H_2](x),$$

where $P[H_1, H_2]$ is the projection mapping from $\mathcal{M}(H_1)$ to $\mathcal{M}(H_2)$, and $g[H_1, H_2]$ is an invertible mapping in $\mathcal{M}(H_1)$ that maps $t_p^{-1}(y)$ to a $(d_I(H_1) - d_I(H_2))$-dimensional subspace of $\mathcal{M}(H_1)$ for each $y \in \mathcal{M}(H_2)$.

We can naturally assume that the 'precession' effect on each subsequence with a certain length l is only related to l and intrinsic properties of the task itself, then consider a length-$2l$ sequence, it could be viewed as two independent measuring of length-l subsequences, hence this error is always bounded by $O(\frac{1}{\sqrt{L}})$. We can formalize this condition into:

$$\frac{\|g[H_1, H_2](x) - x\|_2}{\|g[H_1, H_2](x)\|_2} \le \frac{\kappa}{\sqrt{H_1}}, \forall x \in \mathcal{M}(H_1),$$

13

where κ is a constant, and $\|\cdot\|_2$ represents L2-norm. Also we should have $\mathbb{E}[g(x) - x] = 0$. This results in an extra term in the Bayesian loss. We deduce it again formally from $\mathbb{E}[var(t_p^{-1}[H](x))]$.

$$\mathbb{E}[var(t_p^{-1}[H](x))] = \mathbb{E}[var(g^{-1}[H]P^{-1}[H](x))]$$

$$\leq \mathbb{E}[\frac{\kappa^2}{H}\|y\|_2^2] + \mathbb{E}[var(P^{-1}(x)]$$

$$= \frac{\kappa^2}{H}\sum_{0 \leq i < d_I}\lambda_i + \mathbb{E}[var(P^{-1}(x)]$$

Same to what we have done in section 3.2.3:

$$\mathbb{E}[var(P^{-1}(x))] = \sum_{d_I(H)<=i<d_I}\lambda_i$$

Hence:

$$\mathbb{E}[var(t_p^{-1}[H](x))] \leq \frac{\kappa^2}{H}\sum_{0 \leq i < d_I}\lambda_i + \sum_{d_I(H)<=i<d_I}\lambda_i$$

$$\leq \frac{\lambda_0}{\alpha_Z - 1}(\frac{\kappa^2\alpha_Z}{H} + \frac{1}{d_I(H)^{2\alpha_Z - 1}})$$

which indicates that:

$$\text{loss} \leq K^2(1 - \eta)\frac{\lambda_0}{\alpha_Z - 1}(\frac{\kappa^2\alpha_Z}{H} + \frac{1}{d_I(H)^{\alpha_Z - 1}}) + \eta\sigma_M^2 S^2 d_I(S)$$

Since κ should be small, and typically $\alpha < 2$ (it holds for all our experiment results), the extra term could be neglected. Hence, it is reasonable that we assume $t_p[H_1, H_2] \approx P[H_1, H_2]$.

A.4 Reduction of Loss into Intrinsic Space

We prove that under assumptions made in Section 3.2.1, the loss in the orginal space can be linearly bounded by the loss in the intrinsic space.

Consider we are predicting $x[0 : S]$ from $x[-H : 0]$, let $y_i \in \mathcal{M}(H)$ be the intrinsic vector of $x[-H : 0]$ and $y_o \in \mathcal{M}(S)$ be the intrinsic vector of $x[0 : S]$ (the true intrinsic vector). If we have a model m so that:

$$\mathbb{E}_{x[-H:S]\in\mathcal{O}(H+S)}[|m(y_i) - y_o|_2] \leq L_{ins}$$

where L_{ins} represents the expected error (or loss) in the intrinsic space. Then, from assumption 2 we have:

$$\mathbb{E}_{x[-H:S]\in\mathcal{O}(H+S)}[|\phi^{-1}(m(y_i)) - \phi^{-1}(y_o)|_2] \leq K_I L_{ins}$$

and from assumption 1 we know that $\mathbb{E}_{x[0:S]\in\mathcal{O}(S)}[|\phi^{-1}(y_o) - x[0 : S]|_2] \leq e(S)$. Therefore:

$$L_{ori} = \mathbb{E}_{x[-H:S]\in\mathcal{O}(H+S)}[|\phi^{-1}(m(y_i)) - x[0 : S]|_2]$$

$$\leq 2\mathbb{E}_{x[-H:S]\in\mathcal{O}(H+S)}[|\phi^{-1}(m(y_i)) - \phi^{-1}(y_o)|_2] + 2\mathbb{E}_{x[0:S]\in\mathcal{O}(S)}[|\phi^{-1}(y_o) - x[0 : S]|_2]$$

$$\leq 2K_I L_{ins} + 2e(S).$$

Therefore, the loss in the original space is linearly bounded by the loss in the intrinsic space. W.l.o.g we may focus on studying the loss in the intrinsic space.

B Details about the loss deduction

B.1 Boundary

Recall that in section 3.2.4 we mentioned that there are two scenarios in which the model should work in totally different ways. The boundary between the two scenarios depends not only on the

dataset size D but also on the model size and prediction horizon. A larger model subdivides the intrinsic space into smaller subregions, each containing fewer training samples, with the number of samples per subregion inversely related to model size, expressed as N^{-1}.

In time series forecasting, a unique characteristic is that a single sequence, containing multiple length-H subsequences, counts as multiple samples. However, these subsequences are not fully independent due to their proximity in the intrinsic space. To quantify independence, we define a separation function $f(H)$, assumed to be proportional to H, meaning that a sequence of total length L represents $\frac{L}{f(H)}$ independent samples. Thus, the effective number of independent samples in the dataset scales with $\frac{D}{H}$.

Combining the above analysis, we can define a hyperparameter as $\xi = \frac{D}{NH}$. If ξ is large enough, then we can assume data is sufficient, and the model tends to learn a linear approximation for each subregion; if ξ is small, we can assume data is scarce and the model tends to find the nearest neighbour for each test sample.

B.2 Sufficient Data

We formally derive the total loss in the scenario where we have a large amount of data. First recall that the Bayesian loss calculated in section 3.2.3 is:

$$L_{Bayesian} \approx K_1^2(1 - \eta)\frac{\lambda_0}{(\alpha_Z - 1)d_I(H)^{\alpha_Z - 1}} + \eta\sigma_M^2 S^2 d_I(S)$$

As we assume S is fixed, $\eta\sigma_M^2 S^2 d_I(S)$ is constant and hence could be ignored. Now let's consider the approximation loss, it should come from two sources: the limited model size would result in uncertainty in the subregions, and the limited dataset size would bring noise to the data. Let's consider these two causes separately.

B.2.1 Model Size Limitation

We assume that a model is separating the intrinsic space into N regions, and does a linear prediction for each region. It is easy to verify that the optimal prediction from $\mathcal{M}(H)$ to $\mathcal{M}(S)$ should be first-order K_1-Lipschitz and second-order K_2-Lipschitz, too.

For any region with volume V, the loss could be estimated as:

$$L = \int_V \mathrm{d}^{d_I(H)}x|f(x) - l(x)|^2$$

where l is the predicted linear function for the region. Expanding with Taylor's series, we have $f(x) \approx l(x) + \frac{1}{2}(f'(x) - f'(\mu_V))(x - \mu_V)$. From the second-order Lipschitz condition, we can conclude that:

$$|f(x) - l(x)|^2 \leq K_2^2(x - \mu_V)^4$$

The total loss could be bounded as follows:

$$\text{loss} \leq K_2^2 \int f(x)(x - Q(x))^4 \mathrm{d}x$$

where $Q(x)$ represents the center of the region that x is in and $f(x)$ represents the probability density. We can find that the integral in the formula is exactly the 4th power distortion of the quantization. For an arbitrary quantization with code density $g(x)$, [33] gives an exact value of distortion equal to:

$$\int f(x)(x - Q^*(x))^4 \mathrm{d}x = \pi^{-2} N^{-\frac{4}{d_I(H)}}\Gamma(\frac{d_I(H) + 4}{d_I(H)})(\Gamma(\frac{d_I(H) + 2}{2}))^{\frac{4}{d_I(H)}}\int \frac{f(x)}{g(x)^{\frac{4}{d_I(H)}}}\mathrm{d}x$$

Assume $d_I(H)$ is sufficiently large, then $\Gamma(\frac{d_I(H)+4}{d_I(H)}) \approx 1$. So the expression can be transformed into:

$$\int f(x)(x - Q^*(x))^4 \mathrm{d}x \approx \frac{d_I(H)^2 N^{-\frac{4}{d_I(H)}}}{4\pi^2}\int \frac{f(x)}{g(x)^{\frac{4}{d_I(H)}}}\mathrm{d}x$$

For uniform distribution, $g(x) = 1$ (as we assume $\mathcal{M}(H) = [0,1]^{d_I(H)}$). Hence:

$$\int \frac{f(x)}{g(x)^{\frac{4}{d_I(H)}}} \, dx = 1$$

And the total loss could be bounded as:

$$\text{loss} \leq K_2^2 \frac{d_I(H)^2 N^{-\frac{4}{d_I(H)}}}{4\pi^2}$$

From this bound, we can see that a larger horizon implies a larger burden for the model, which might lead to worse performance for the model.

B.2.2 Dataset Limitation

Following the previous chapter, as we assume the mapping f represented by the model is a piecewise linear function, then regionally it can be written as the usual form of the gaussian-markov linear model, assuming there are homoscedastic uncorrelated noise ε for every single sample in the dataset. Now we show that this noise will cause another term of loss even compared to the optimal performance of the model. We first analyze a single region containing D_i samples. Consider X, Y from the training set:

$$Y = X\beta + \varepsilon.$$

Where X is of shape $D_i \times d_I(H)$ and Y, ε is of shape $D \times d_I(S)$. β is the variable to learn of shape $d_I(H) \times d_I(S)$. The BLUE estimator $\hat{\beta}$ should be $\hat{\beta} = (X^T X)^{-1} X^T Y$.

Now consider X', Y' from the test set:

$$Y' = X'\beta + \varepsilon'$$

$$L_{recon} = E[(Y' - X'\beta)^2]$$

$$= E[(\varepsilon' - X'(X^T X)^{-1} X^T \varepsilon)^2]$$

$$= E[\varepsilon'^2] + E[(X'(X^T X)^{-1} X^T \varepsilon)^2]$$

The first term goes to 0 if the test set is sufficiently large, and it is a constant given a fixed test set, so we can simply ignore it in our analysis. Consider the second term, we can show that:

$$\mathbb{E}[(X'(X^T X)^{-1} X^T \varepsilon)^2] = D_i^{-1} d_I(H)\varepsilon_y^2$$

Recall the noise consisting of two sources:

$$Y = X\beta + F[S](X^* - P_c[H]^{-1}(X)) + \varepsilon_B$$

As we've done in section 3.2.3, we can derive that:

$$\text{var}(X^* - P_c[H]^{-1}(X)) \approx \sum_{d_I(H) \leq i < d_I} \lambda_i$$

$$\approx \frac{\lambda_0}{(\alpha_Z - 1)d_I^{\alpha_Z - 1}(H)}$$

As $F[S]$ is K_1-Lipschitz, this indicates that:

$$\varepsilon_h \leq \frac{K_1^2 \lambda_0}{(\alpha_Z - 1)d_I^{\alpha_Z - 1}(H)}$$

And as mentioned in section 3.2.3 $\text{var}(\varepsilon_B) = \sigma_M^2 S^2 d_I(S)$, hence we can bound the total loss related to D_i into:

$$\text{loss}_i \leq D_i^{-1} d_I(H)(\sigma_M^2 S^2 d_I(S) + \frac{K_1^2 \lambda_0}{(\alpha_Z - 1)d_I^{\alpha_Z - 1}(H)})$$

It is worth noticing that, although different regions have their own distributions and hence may have different eigenvalues, the loss expression is irrelevant with the eigenvalues, so we can ignore this issue. Considering the entire dataset with all regions, the total loss should be:

$$\text{loss} \approx \frac{\sum_{j=1}^{N} \text{loss}_i \cdot D_i}{D}$$

$$= \frac{N d_I(H)}{D}(\sigma_M^2 S^2 d_I(S) + \frac{K_1^2 \lambda_0}{(\alpha_Z - 1)d_I^{\alpha_Z - 1}(H)})$$

This indicates that, using a larger model would be more prone to being affected by noise in the dataset, so especially when the data is noisy and the dataset size is limited, our theory suggests that it would perhaps be better to use a smaller model.

B.2.3 Total loss

Combining the above approximation loss and the Bayesian loss, the total loss would be:

$$\text{loss} \approx K_2^2 \frac{d^2 N^{-\frac{4}{d}}}{4\pi^2} + K_1^2(1-\eta)\frac{\lambda_0}{\alpha_Z - 1}\frac{1}{d_I(H)^{\alpha_Z - 1}} + \frac{Nd_I(H)}{D}\sigma_M^2 S^2 d_I(S)$$

B.3 Scarce Data

As the model finds the nearest neighbor, the approximation loss in this scenario should consist of two sources: the distance from the test sample to its nearest neighbor and the noise in its nearest neighbor. If the distribution is smooth, then locally around any point x, the distribution of other points can be viewed as uniform when D is large.

Consider a sphere of radius r around a test sample, the probability that no point in the training set lies in the ball is approximately:

$$P(r) \approx e^{-DV(r)} = e^{-DC_d r^{d_I(H)}}$$

where $C_d = \frac{\pi^{\frac{d_I(H)}{2}}}{\Gamma(d_I(H)/2+1)}$ denotes the volume of a unit sphere.

From this, we can derive that:

$$\mathbb{E}_x[r^2] \approx \frac{(\Gamma(\frac{d}{2}+1))^{\frac{2}{d_I(H)}}\Gamma(\frac{2}{d_I(H)})}{\pi d_I(H)}D^{-\frac{2}{d_I(H)}}f(x)^{\frac{2}{d_I(H)}}$$

Considering $d_I(H)$ is large, we can get that:

$$\mathbb{E}[r^2] \approx \frac{d_I(H)}{4\pi}D^{-\frac{2}{d_I(H)}}$$

which means:

$$\text{loss} \approx \frac{K_1^2 d}{4\pi}fD^{-\frac{2}{d}}$$

Combining the Bayesian loss similar to the previous scenario, the total loss should be:

$$\text{loss} \approx K_1^2(1-\eta)\frac{\lambda_0}{\alpha_Z - 1}\frac{1}{d_I(H)^{\alpha_Z - 1}} + \frac{K_1^2}{4\pi}d_I(H)D^{-\frac{2}{d_I(H)}}$$

C   Optimal Horizon deduction

Continuing from section B, we study the optimal horizon in two scenarios.

C.1   Sufficient Data

Let's first study the case where the dataset is large, which is the case in section B.2.

C.1.1   Small Model

If the model size is too small compared to the dataset size, i.e. $N = o(D^{\frac{d_I(H)}{d_I(H)+4}})$ (however we still assume N is sufficiently large compared to other variables), then the effect of dataset size on picking the optimal horizon could be neglected. In this case, the optimal value of $d_I(H)$ is unrelated with the dataset size D, and should satisfy:

$$d_I^* = (\frac{K_1^2\pi^2(1-\eta)\lambda_0 N^{\frac{4}{d_I^*}}}{K_2^2 \ln N})^{\frac{1}{\alpha_Z}}$$

Solving the formula we get the following result:

$$d_I^* = \mathcal{W}\left(\frac{4}{\alpha_Z C_0^{\frac{1}{\alpha_Z}}} \ln^{1+\frac{1}{\alpha_Z}} N\right) \approx \frac{4}{\alpha_Z C_0^{\frac{1}{\alpha_Z}}} \ln^{1+\frac{1}{\alpha_Z}} N$$

where $\mathcal{W}(\cdot)$ is the Lambert W function and $C_0 = \frac{K_1^2 \pi^2 (1-\eta)\lambda_0}{K_2^2}$.

We can see that in this case, d_I^* is not only irrelevant with D, but also changes very little with N.

C.1.2 Large Model

On the other hand, if we assume the model is large enough, such that $N = \omega(D^{\frac{d_I(H)}{d_I(H)+4}})$, then the noise effect would be dominant in picking the optimal H. In this case, the optimal value of $d_I(H)$ should be:

$$d_I^* = \left(\frac{K^2(1-\eta)\lambda_0 D}{N\sigma_M^2 S^2 d_I(S)}\right)^{\frac{1}{\alpha_Z}}$$

We can see that in this case the optimal d_I changes rapidly to D and N. Moreover, the optimal horizon increases with a larger dataset size D and decreases with a larger model size N (this is quite counter-intuitive but it surprisingly matches our experiment result).

C.2 Scarce Data

This is the case in section B.3. We can directly solve $\frac{\partial \text{loss}}{\partial d_I(H)} = 0$, which is equivalent to:

$$\frac{4\pi(1-\eta)\lambda_0}{d_I^{*\alpha_Z}} = D^{-\frac{2}{d_I^*}}\left(1 + \frac{2\ln D}{d_I^*}\right)$$

We can estimate that $d_I^* = \beta C^{\frac{1}{\alpha_Z-1}} \frac{\ln D}{\ln \ln D}$

$$d_I^* \approx \frac{2}{\alpha_Z + 1} \frac{\ln D}{\ln \ln D}$$

where $C = 4\pi(1-\eta)\lambda_0$. Substituting d_I^* into the formula, we can get:

$$(\alpha_Z - 1)\beta^2 + \beta - \frac{2}{C^{\frac{1}{\alpha_Z-1}}} = 0$$

Hence we have:

$$\beta = \frac{\sqrt{1 + \frac{8(\alpha_Z-1)}{C^{\frac{1}{\alpha_Z-1}}}}}{2(\alpha_Z - 1)}$$

and d_I^* should be:

$$d_I^* = \frac{\sqrt{1 + \frac{8(\alpha_Z-1)}{C^{\frac{1}{\alpha_Z-1}}}}}{2(\alpha_Z - 1)} C^{\frac{1}{\alpha_Z-1}} \frac{\ln D}{\ln \ln D} \propto \frac{\ln D}{\ln \ln D}$$

Compared to the first scenario, the optimal d_I almost doesn't change in this scenario.

D Details about model's partitioning

As we mentioned in section 3.2.4, we assume the model partitions the intrinsic space uniformly, but there should be better partitions. As [33] states the optimal partition should satisfy $g(x) \propto f(x)^{\frac{d_I(H)}{d_I(H)+4}}$, and if the distribution is Gaussian then the loss (only for this term) should satisfy:

$$L_p \approx K_2^2 \frac{\lambda_0^2 N^{-\frac{4}{d_I(H)}}}{e^2 d_I(H)^{2\alpha-2}}$$

Replacing the corresponding term with this, we see that in the case in section C.1.2, since the noise makes the most impact, this term does not matter. However, in the case in section C.1.1, if we assume the dataset size is infinite, then the loss is asymptotically monotonically decreasing with $d_I(H)$. Thus, the larger the horizon we use, the better performance the model will potentially reach.

However, experiments show that the optimal horizon always exists in this case. This might suggest that the model structure limits itself from finding the best partitioning. Hence, we expect that when we design a good enough model structure (maybe only exists theoretically) and when we have enough data, we should be able to make full use of our horizon regardless of the model size or computational resources.

E Experiment settings

E.1 Dataset

Dataset	Dim	Pred Len	Dataset Size	Frequency	Information
ETTh1	7	192	$(8545, 2881, 2881)$	Hourly	Electricity
ETTh2	7	192	$(8545, 2881, 2881)$	Hourly	Electricity
ETTm1	7	192	$(34465, 11521, 11521)$	15 min	Electricity
ETTm2	7	192	$(34465, 11521, 11521)$	15 min	Electricity
Exchange	8	192	$(5120, 665, 1422)$	Daily	Economy
Weather	21	192	$(36792, 5271, 10540)$	10 min	Weather
ECL	321	192	$(18317, 2633, 5261)$	Hourly	Electricity
Traffic	862	192	$(12185, 1757, 3509)$	Hourly	Transportation

Table 1: Dataset used for experiments

We conduct experiments on 8 datasets listed here. ETTh1, ETTh2, ETTm1, ETTm2 [4] contains 7 factors of electricity transformer from July 2016 - July 2018 at a certain frequency. Exchange [5] includes panel data for daily exchange rates from 8 countries across 27 years. Weather [5] includes 21 meteorological factors from a Weather Station. ECL [5] records electricity consumption of 321 clients. Traffic [5] records hourly road occupancy rates for 2 years in San Francisco Bay area freeways. For all experiments, the pred-len is set to 192.

In our experiments, we do 3 iterations for ETTh1, ETTh2, ETTm1, ETTm2, Weather and Exchange, and draw graphs with error bar equaling to the standard error of these iterations. For Weather dataset, actually the error is minor. (see Appendix E.3), and for larger datasets including Traffic and Electricity, the error is even smaller. Therefore, we conduct 1 iters for Traffic and ECL since they are large and will introduce minor std error.

E.2 Drop-last issue

The drop-last issue is reported by several researchers [3, 39]. That is, in some previous work evaluating the model on test set with drop_last=True setting may cause additional errors related to test batch size. In our experiments, we use drop-last set to False to avoid this issue.

E.3 Time-related Domain Difference in Datasets: can be neglected to a certain extent

We use the first p percent of our training dataset as our dataset as available training data. This may introduce an error caused by time-index-related domain differences in these datasets. To avoid this we utilize the Instance Normalization [19]. Our further experiments on the Weather dataset show that this error is actually minor compared to the randomness introduced in the training procedure. We use an NLinear model to forecast 192 frames based on a look-back window of size 336. We use 10% of training data, from different slices in the training set. For each slice, we repeat the experiment for 3 times. The result is:

It can be seen that the variance introduced by different slicing points is very small.

MSE	starting point=20%	40%	60%	80%
Exp1	0.223	0.220	0.222	0.223
Exp2	0.221	0.221	0.223	0.224
Exp3	0.221	0.221	0.223	0.221
Avg	0.222 ± 0.001	0.221 ± 0.001	0.223 ± 0.001	0.223 ± 0.001

Table 2: MSE measured for weather dataset

E.4 Models and Experiment settings

For all models we utilize instance normalization that normalizes an input sequence based on its mean and std. All the deep learning networks are implemented in PyTorch[40] and conducted on a cluster with NVIDIA RTX 3080, RTX 3090, RTX 4090D and A100 40GB GPUs. Although we use multiple types of GPUs, it is possible to carry out all experiments on a single GPU with greater or equal to 24GB of memory. Experiments conducted in this work cost about $O(10^3)$ gpu hours for RTX 3090 in total.

E.4.1 Linear Models

For linear models, we use the linear layer after instance normalization, and perform de-normalization after the linear layer, equivalent to the NLinear model[1], for all datasets and all experiments.Batch size are chosen from $\{1024, 2048, 4096, 8192, 16384\}$, learning rate is chosen from $\{0.003, 0.001\}$ and weight decay is chosen from $\{0.0005, 0.005, 0.001, 0.0001\}$, and we decay the learning rate with a factor of $\{0.96, 0.97, 0.98\}$ for at most 100 epochs with patience at most 30. For larger datasets including Electricity, Traffic and weather the learning rate and weight decay barely make differences to the result. For relatively small datasets includgin ETTh1/ETTh2, ETTm1/ETTm2 and Exchange, larger weight decay may improve performance.

E.4.2 MLP

We use two types of MLPs. For the data scaling (including the HorizonXDataset) experiments, we use the gated MLP[41]: $\text{hidden}(x) = W_1(x) \odot \text{sigmoid}(W_2(x))$ where $\odot$ is the element-wise product to improve the capability of our model. For the width scaling (and the HorizonXWidth) experiments, we use plain MLP with leaky-relu as activation function.

Gated MLP is used on Traffic, Weather and Electricity dataset. On these datasets we use $3 - layer$ mlp with model dimension 768 and hidden dimension $2 * 768$. Learning rate is set to 0.001 and weight decay is set to $\{0.00001, 0.0005\}$.

For single layer mlp that we use to experiment the effect of model size scaling, we us the relu activation function and the model width denotes the hidden dimension of the single hidden layer. For different datasets, the learning rate is set to $\{0.003\}$ and the weight decay is set to $\{0.0005\}$.

Batchsize is set to 4096 or 8192 for these cases.

E.4.3 iTransformer

We follow the original codes provided by iTransformer[10].

For dataset size scaling and (Horizon X Dataset Size) case we use 4 iTransformer encoder blocks of dimension 512 for Traffic, Weather and Electricity. For other smaller datasets, we use 1 blocks of dimension 192.

For Model size scaling case we use 2 iTransformer encoder blocks for large datasets including Traffic, Weather and ECL, and 1 block for smaller datasets. The width indicates the model dimension as well as the feedforward dimension.

Learning rate is set to 0.003 for small datasets and 0.0003 for large datasets. We decay the learning rate with a factor of 0.97 or 0.98 every epoch, and run the training procedure for at most 100 epochs with patience 10 for small datasets, and 40 epochs with patience 10 for larger datasets including Traffic, Weather and Electricity. Batch size is chosen from $\{12, 20, 32\}$, mainly to fit into the gpu memory.

E.4.4 ModernTCN

We follow the original codes provided by ModernTCN[10]. We use the recommended settings given by the authors of ModernTCN except for the horizon, the channel dimension and the training procedure. Since ModernTCN has many hyperparameters that can be flexibly tuned, we do not include other same hyperparameter here, and please refer to the original paper and cods for ModernTCN for a more detailed description. Here we only include the different hyper-parameters.

For Dataset Size Scaling and (Horizon X Dataset Size) experiments, we only modify the look back horizon. For training procedure, we use learning rate in $\{0.00005, 0.0001\}$. We train for 100 epochs with patience 20 for small datasets and 40 epochs with patience 10 for larger datasets.

For Model Size Scaling and (Horizon X Model Size) experiments, we only modify the look back horizon as well as the chanenl dimension and the depth-wise dimensions in the code.Again the learning rate is chosen from $\{0.00005, 0.0001\}$, and we train for 100 epochs with patience 20 for small datasets and 40 epochs with patience 10 for larger datasets. Batchsize is chosen from $\{16, 64\}$ following the original hyperparameter settings given by the author of ModernTCN.

F Downsampling May Improve Performance: power of patch, low-pass-filter, and downsampling

Previous work has proven the success of patches [7] and low-pass-filter [3] for time series prediction. These can be explained with our theory by making the assumption that high frequency features are the most unimportant ones: thus by filtering up high frequency features we make the model visible to the most important dimensions of the intrinsic space.

From a more detailed theoretical perspective, downsampling can be viewed as a projection to a subspace in the intrinsic space and thus has a similar effect as decreasing the horizon. Experimental results (as shown in experiments in previous works like FITS[3], PatchTST[7] and in Figure 7) show that the projected subspace of higher frequency tends to fall on the large-eigenvalue directions, or the 'invisible' dimensions masked by the projection tends to be the more unimportant ones. Although the precise effect of down sampling is unknown or may need further assumptions and methods for precise consideration, it is acceptable that we may approximate the effect of down sampling to be similar to a projection to the first $d_{eff} < d_I(H)$ dimensions in the intrinsic space. After making this assumption, the overall loss could be expressed as:

$$loss_{new} = L(d_{eff}) \text{where } d_{eff} < d_I(H).$$

Hence, if the original intrinsic dimension is larger than (local) optimal d, indicating $\partial loss/\partial d > 0$, reducing d from $d_I(H)$ to d_{eff} would help reduce loss. Otherwise if $d_I(H)$ is already smaller than (local) optimal d^*, meaning $\partial loss/\partial d < 0$, we would expect no performance improvement from reducing d from $d_I(H)$ to d_{eff}.

To validate this idea, We do a similar experiment here, that is to do down-sampling and to use the downsampled sequence as input sequence. We conduct experiments on Traffic dataset and Weather dataset with a lookback horizon set to be 336 and a prediction length set to be 192. We use a 4-layer 512-dim gated mlp for traffic dataset and 2-layer 192-dim gated mlp for weather dataset. We can see that doing downsampling would improve the performance of the weather dataset and the optimal downsample ratio also changes with different training dataset size. The explanation for this is similar to how we explain the optimal horizon growth with the growth of the training dataset. More data in the training dataset would lead to less noise on higher dimensions, thus seeing more of these dimensions (or filtering out fewer dimensions with downsampling with longer interpolate length) can improve performance.

Meanwhile, for the traffic dataset, downsampling may hurt performance. This is because Traffic has large amount of training data, and 336 is smaller than the optimal horizon. At this all dimensions in the intrinsic space are not dominated by noise, hence filtering them out would hurt performance, and we would expect the performance to be improved with the increment of interpolate length.
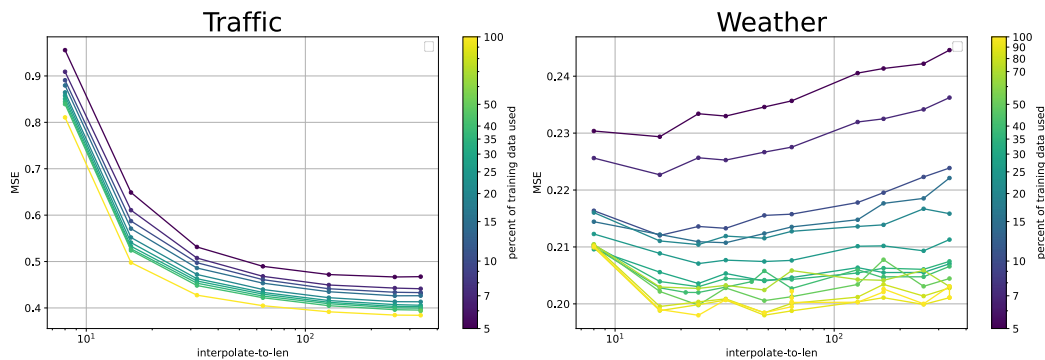
Figure 7: MSE loss v.s. interpolate-to-len (length after downsampling) for Traffic and Weather dataset for a certain amount of training data.

G   PCA settings

We use PCA to obtain eigen values that we use to approximate the importance of features in the intrinsic space. We conduct PCA at (all or part of the) training samples (after instance normalization), mainly for the Channel-Independent case. For the PCA used in Figure4 and Figure 8 the horizon is set to 2000 for both datasets and we perform PCA on all the training samples. We provide more PCA results in Figure 8: if we approximately view the PCA eig-val as the feature importance, then they do follow a Zip-f law approximately.
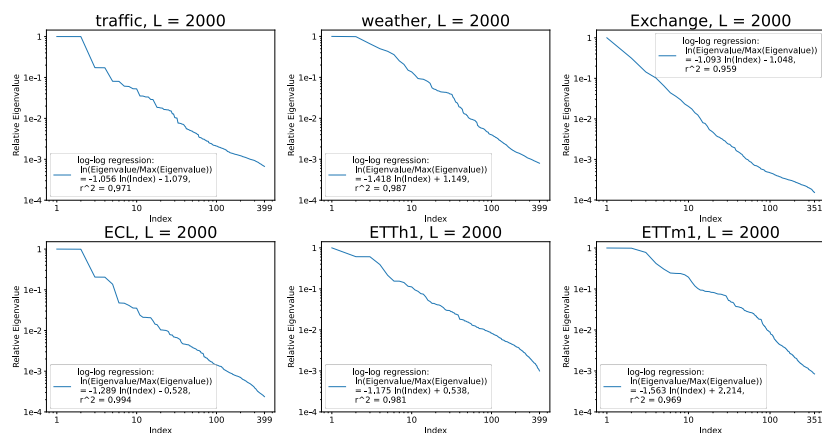

Figure 8: PCA results for $L = 2000$ under Channel-Independent setting on some datasets.

We further conduct PCA on intermediate vectors of the iTransformer[8] model and obtain results, further validating the Zip-f assumption in our paper for intrinsic space for Non-Linear Channel-Dependent Multivariable cases, as shown in Figure 9.

H   Foundation Models for Mixed Datasets

Our theory holds without the need for any modification as long as the dataset itself follows a Zip-f distribution in the intrinsic space. This assumption is sort of natural given that Zip-f law is a natural distribution, and previous datasets we examined (like Traffic, ETT, etc) to follow the Zip-f law are composed of smaller datasets. Moreover, we provide further analysis from Theoretical and Experimental perspective on why it holds for mixed datasets.
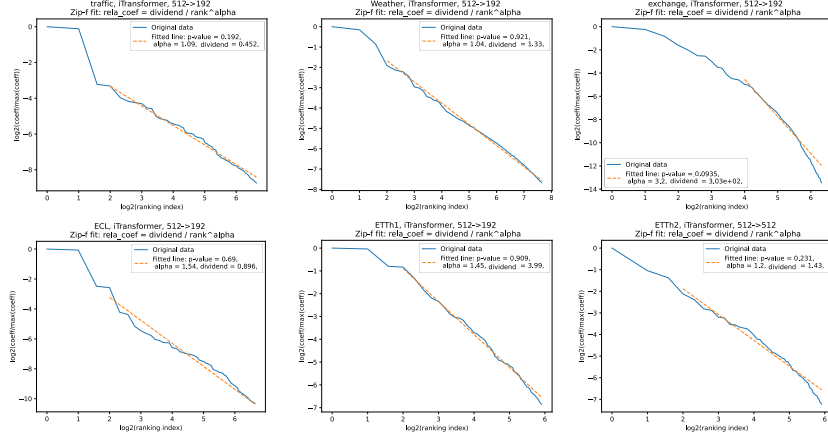
Figure 9: PCA obtained with iTransformer features. Zip-f law fitting receives $p > 0.05$ in all cases. Note that it is hard for deep-learning methods (like iTransformer) to learn feature with small covariance very well, it may be hard to learn features in exchange dataset well (in which feature degrades very fast), hence the seemingly not-fitting figure for exchange is more likely caused by under-training of deep learning models.

H.1 Zip-f distribution for Mixed Datasets: a toy model

Theoretically, if a large dataset is composed of s sub-datasets of similar size each following the Zip-f law with degradation coefficient $\alpha_1 < \alpha_2 < \ldots < \alpha_s$, each with size S_i and follows Zip-f law: $\lambda_{ij} = A_i/j^{\alpha_i}$ where λ_{ij} represents the j-th largest eigenvalue of the i-th dataset. Suppose the intrinsic dimensions are orthogonal with each other (hence PCA components are orthogonal). A simple assumption is that the new intrinsic space is a direct product of the old intrinsic spaces, hence eigenvalues are the union of old eigenvalues. For which, an eigenvalue of value S should be the idx_{total}-largest eigen value, in which:

$$idx_{total} = \sum_i idx_i = \sum_i (A_i/S)^{1/\alpha_i}.$$

When S is small (or correspondingly, when idx_{total} is large) this sum is dominated by small αs, and in limitation cases the sum is dominated by α_1 term: $idx \approx (A_1/S)^{1/\alpha_1} + C$, which is approximately a Zip-f distribution.

H.2 Experiment results for Zip-f distribution for Mixed Datasets

Experimentally, we use the Mixed Dataset of Traffic, Weather, ETTh1, ETTh2, ETTm1, ETTm2, exchange and ECL to train a Channel-Independent 2-layer 512-dimension MLP and use the Intermediate vector before the decoder layer as a feature vector to do PCA analysis. We found that the result follows a Zip-f distribution for higher-order components, as shown in Figure 10.

I   More comparison with other formulas

We compare AiC and BiC value of our proposed formula $f(x) = A + B/x^\alpha$ with other possible formulas, on ModernTCN, iTraansformer and MLP or Linear models:

- $g_1(x) = A/x^\alpha$
- $g_2(x) = A + B\log(x)$
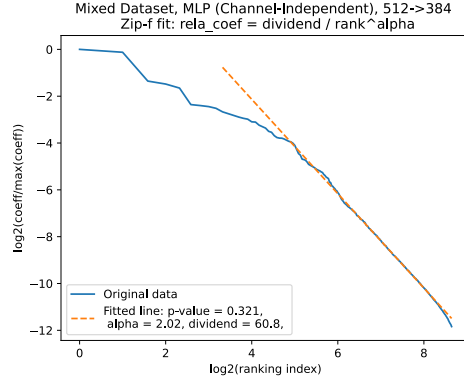- $g_3(x) = A + Bx + Cx^2$

23

Figure 10: PCA obtained with MLP features trained on Mixed Dataset. The 2-layer 512-dimension MLP is trained under Channel Independent setting on a mixed dataset of Traffic, Weather, Exchange, ETTh1, ETTh2, ETTm1, ETTm2 and ECL. The mixed feature shows a well-aligned Zip-f law for features of higher rankings.

AiC, BiC	Traffic	Weather	ETTh1	ETTh2
f	**-103.9,-103.4**	**-87.2,-87.0**	**-69.5,-69.6**	**-64.7,-65.3**
g1	-95.3,-94.9	-79.1,-79.0	-60.6,-60.7	-45.6,-46.0
g2	-94.6,-63.3	-71.1,-71.0	-59.7,-59.8	-45.4,-45.8
g3	-93.3,-94.9	-81.3,-83.1	-56.5,-56.7	-43.1,-43.8

Table 3: regression results for ModernTCN

AiC, BiC	Traffic	Weather	ETTh1	ETTh2
f	**-71.6,-70.7**	**-74.8,-74.6**	**-50.7,-50.9**	-56.9,-56.7
g1	-66.7,-66.1	-73.1,-72.9	-45.7,-45.8	**-57.9,-57.8**
g2	-63.3,-62.6	-71.1,-71.0	-41.7,-41.8	**-57.9,-57.7**
g3	-54.9,-56.3	-70.1,-72.0	-38.9,-39.7	**-56.4,-56.2**

Table 4: regression results for iTransformer

AiC, BiC	Traffic	Weather	ETTh1	ETTh2
f	**-91.9,-91.3**	**-91.5,-91.7**	**-89.1,-88.8**	**-62.8,-62.6**
g1	-67.1,-66.7	-83.1,-83.2	-67.5,-67.4	-61.1,-60.9
g2	-66.0,-65.6	-82.1,-82.3	-65.6,-65.5	-59.9,-59.8
g3	-60.1,-61.2	-81.3,-83.4	**-89.1,-88.8**	-60.1,-59.9

Table 5: regression results for MLP or Linear models

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We made claims in the abstract and introduction part that our main contribution includes: (1) proposes a theory (please refer to Section 3) for scaling law for time series forecasting, taking into account the impact of dataset size, model size and look back horizon and (2) conduct experiments (please refer to Section 4) to verify the validity of scaling law on dataset size and model complexity, as well as to validate our theoretical framework on time series forecasting models. This aligns with our claims and our paper's contribution.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: We discuss our limitation in Section 5.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We gave thorough assumptions and clear proofs. More specifically, we formulate the intrinsic space in section 3.2.1, and calculate the loss in the rest of section 3.2. Then we analyze the optimal horizon in section 3.3. The detailed proof is shown from Appendix A to C

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.

- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We give details for our experiments at Appendix E. We also provide code and logs in the supplementary materials.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: we describe the hyperparameters used in Appendix E, and we provide code in our supplementary material. We will make our code publicly available after refactoring.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We give different dataset configs and corresponding hyperparameters at Appendix E.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report and analyze the error in our experiments. Please refer to Appendix E for more details about errors.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide information about implementation framework and compute resources at Appendix E.4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We follow NeurIPS Code of Ethics and give clear details.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: Our work proposes a theory for scaling behaviors for time series forecasting tasks and experimentally validates the proposed theory. Though we hope our work may inspire work for time series forecasting, this work alone is a foundational research that sees no social impact on its own.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Our work does not release new models or datasets, hence the paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We use previously proposed models and datasets, and include license for existing assets.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: We use publicly available models and datasets and do not release new assets including models or datasets. We mainly conduct experiments based on code bases of previous work.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: Our work does not involve crowd sourcing or research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: Our work does not involve crowd sourcing or research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.