TILDE-Q: A TRANSFORMATION INVARIANT LOSS FUNCTION FOR TIME-SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Time-series forecasting has caught increasing attention in the AI research field due to its importance in solving real world problems across different domains, such as energy, weather, traffic, and economy. As shown in various types of data, it has been a must-see issue to deal with drastic changes, temporal patterns, and shapes in sequential data that previous models are weak in prediction. This is because most cases in time-series forecasting aims to minimize L_p norm distances as loss functions, such as mean absolute error (MAE) or mean square error (MSE). These loss functions are vulnerable to not only consider temporal dynamics modeling, but also capture the shape of signals. In addition, these functions often make models misbehave and return uncorrelated results to the original time-series. To become an effective loss function, it has to be invariant to the set of distortions between two time-series data instead of just comparing exact values. In this paper, we propose a novel loss function, called TILDE-Q (Transformation Invariant Loss function with Distance EQuilibrium), that not only considers the distortions in amplitude and phase but also allows models to capture the shape of time-series sequences. In addition, TILDE-Q supports modeling periodic and non-periodic temporal dynamics at the same time. We evaluate the effectiveness of TILDE-Q by conducting extensive experiments with respect to periodic and non-periodic conditions of data, from naive models to state-of-the-art models. The experiment results indicate that the models trained with TILDE-Q outperforms those trained with other training metrics (e.g., MSE, dynamic time warping (DTW), temporal distortion index (TDI), and longest common subsequence (LCSS)).

1 INTRODUCTION

Time-series forecasting has been a core problem across various domains, including traffic domain (Li et al., 2018; Lee et al., 2020), economy (Zhu & Shasha, 2002), and disease propagation analysis (Matsubara et al., 2014). The crucial part of the time-series forecasting is modeling of the complex temporal dynamics (e.g., non-stationary signal, periodicity). Temporal dynamics, intuitively, shape, has always been one of the most attention-getting keywords in time-series domains, such as rush hour of traffic data or abnormal usage of the electricity (Keogh et al., 2003; Bakshi & Stephanopoulos, 1994; Weigend & Gershenfeld, 1994; Wu et al., 2021; Zhou et al., 2022). Deep learning methods are one of the appealing solutions to model complex non-linear temporal dependencies and non-stationary signals, but recent work reveals that even deep learning is often insufficient to model temporal dynamics. To properly model the temporal dynamics, Wu et al. (2021); Zhou et al. (2022) have proposed a novel deep learning approaches with input sequence decomposition. Le Guen & Thome (2019) try to model sudden changes timely and accurately with dynamic time warping (DTW). Bica et al. (2020) adopts domain adversarial training to learn balanced representations, which is a treatment invariant representations over time. Wu et al. (2021); Zhou et al. (2022) have less attention to the essence of the problem: a shape, in other words, temporal dynamics. Le Guen & Thome (2019); Bica et al. (2020) try to capture the shape but still have some limitations like Fig. 1 (c).

A **shape** is a part of patterns in time-series data with a given time interval that could give valuable information, such as rise, drop, trough, peak, and plateau. We call the prediction is *informative* when it could properly consider the shape. In real-world applications like economics, such informative prediction is crucial to make decisions. To gain informative forecasting, the model should consider the shape rather than only aim to forecast accurate value for each time step. However, existing models do

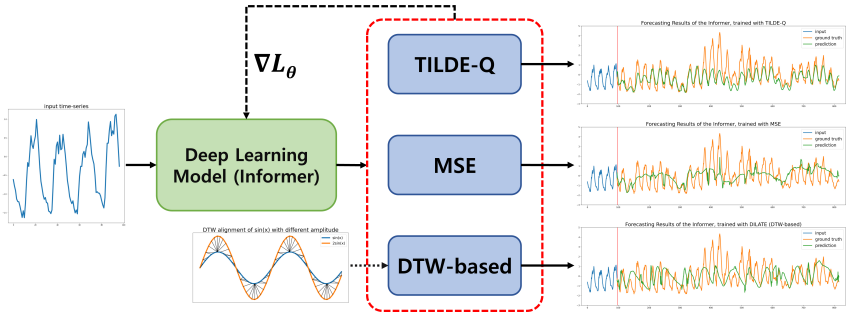

Figure 1: Ground-truth and forecasting results of Informer model with three training metrics (top) TILDE-Q, (middle) MSE, and (bottom) DTW-based loss function. (middle) MSE tends to generate non-informative forecasting results, similar to an average value of data and (bottom) DTW often produces misaligned results. Red dotted box contains three training metrics.

not consider learning shape (Wu et al., 2021; Zhou et al., 2022; Bica et al., 2020; Le Guen & Thome, 2019), so the forecasting results are often inaccurate and uninformative, because deep learning model tends to learn in *easy way* (Karras et al., 2019). Fig. 1 shows three real forecasting results with same model, different training metrics. When we utilize mean squared error (MSE) as an objectives, the model only aims to reduce gap between prediction and ground truth for each time-step. As a results, the model generates relatively *easy* prediction regardless of temporal dynamics (Fig. 1 (b)). It rarely gives information about original time-series. In contrast, if we consider both gap and shape of prediction and ground truth, the model could achieve both accuracy and temporal dynamics, as shown in Fig. 1 (a).

In this work, we aim to design a novel objective function that guides models to improve forecasting performance by learning the shapes in time-series data. To design such shape-aware loss function, we review existing literature (Esling & Agon, 2012; Bakshi & Stephanopoulos, 1994; Keogh, 2003) and investigate the notions of *shapes* and *distortions* that interrupt measurement for recognizing similarity of two time-series data in terms of shapes (Sec. 3.1, Sec. 3.2, and Sec. 3.3). Based on the investigation, we newly propose required conditions for constructing an objective function for shape-aware time-series forecasting (Sec. 3.4). We then present a novel loss function, TILDE-Q (Transformation Invariant Loss function with Distance EQualibrium), that enables shape-aware representation learning with three different loss terms, which are invariant to the distortions (Sec. 4). For evaluation, we conduct extensive experiments with state-of-the-art deep learning models for time-series forecasting with TILDE-Q. The results indicate that TILDE-Q is model-agnostic and could improve accuracy of existing models, compared to MSE and DILATE.

Contributions We make the following contributions: (1) To understand shape-awareness and distortion invariances in time-series forecasting, we investigate existing distortions in amplitude and phase; (2) we implement TILDE-Q that has invariances to many existing distortions and achieves shape-awareness and informative forecasting in a timely manner; and (3) we show that the proposed TILDE-Q allows models to have higher accuracy compared to those with existing metrics such as DTW, TDI, and LCSS on average.

2 Related Work

2.1 Time-Series Forecasting

There are many methods for time-series forecasting from traditional ones, such as ARIMA model (Box et al., 2015) and hidden markov model (Pesaran et al., 2004) to recent deep learning models. In this section, we briefly describe the recent deep learning models for time-series forecasting. Starting with the huge success of the recurrent neural networks (RNNs) (Clevert et al., 2016; Li et al., 2018; Yu et al., 2017), researchers have developed novel deep learning architectures, improving forecasting performance. To effectively capture long-term dependency, which is a weakness of RNNs, Stoller et al. (2020) have proposed convolutional neural networks (CNNs). However, it is required

to stack lots of the same CNNs to capture long-term dependency (Zhou et al., 2021). Attention-based approaches have been another popular research direction in time-series forecasting, including Transformer (Vaswani et al., 2017) and Informer (Zhou et al., 2021). Although the attention-based models effectively capture temporal dependencies, they require high computational cost and often struggle to find proper temporal information (Wu et al., 2021). To cope with the problem, Wu et al. (2021); Zhou et al. (2022) utilize the input decomposition method that helps models better encode appropriate information. The other state-of-the-art models adopt neural memory networks (Kaiser et al., 2017; Sukhbaatar et al., 2015; Madotto et al., 2018; Lee et al., 2022), which refer to historical data stored in memory to generate meaningful representation.

2.2 Training Metrics

Conventionally, mean squared error (MSE), L_p norm and its variants are the mainstream to optimize forecasting models. However, they are not the best metric to train forecasting models (Esling & Agon, 2012) because time-series is temporally continuous. Additionally, L_p norm gives less information about temporal correlation among time-series data. To better model temporal dynamics in time-series data, researchers have used differentiable, approximated dynamic time warping (DTW), as an alternative metric of MSE (Cuturi & Blondel, 2017; Abid & Zou, 2018; Mensch & Blondel, 2018). However, using DTW as a loss function results in ignoring temporal localization of changes. Recently, Le Guen & Thome (2019) suggests DILATE, a training metric to timely catch sudden changes of non-stationary signals with smooth approximation of DTW and penalized temporal distortion index (TDI). To guarantee to work in a timely manner, Le Guen & Thome (2019) introduce a loss function that gives a harsh penalty when predictions show high temporal distortion. However, TDI relies on the DTW path, and DTW often shows misalignment because of its noise- and scale-sensitive. Thus, DILATE often loses its advantage with complex data, showing disadvantages at the beginning of the training. In this work, we discuss distortions and transformation invariances and design a new loss function that allows models to learn shapes in the data and produce noise-robust forecasting results.

3 Preliminary

In this section, we aim to investigate common distortions without losing the goal of time-series forecasting (i.e., modeling temporal dynamics and accurate forecasting). To help understand the concepts, we first define notations and terms (Sec. 3.1). We then discuss common distortions in time-series in transformation perspectives that need to be considered for building a shape-aware loss function (Sec. 3.2) and describe how other loss functions (e.g., DTW and TDI) handle shapes during learning (Sec. 3.3). Last, we explain the conditions for effective time-series forecasting (Sec. 3.4).

3.1 Notations and Definitions

Let X_t denote a data point at a time step t. Then, we can define a time-series forecasting problem as:

Definition 1. *Given T-length historical time-series $\mathbf{X} = [X_{t-T+1}, \ldots, X_t], X_i \in \mathbb{R}^F$ at time t and corresponding T'-length future time-series $\mathbf{Y} = [Y_{t+1}, \ldots, Y_{t+T'}], Y_i \in \mathbb{R}^C$, time-series forecasting aims to learn mapping function $f : \mathbb{R}^{T \times F} \rightarrow \mathbb{R}^{T' \times C}$.*

To distinguish the label (i.e., ground-truth) and prediction time-series data, we note the label data as $\mathbf{Y}$ and prediction data as $\hat{\mathbf{Y}}$. Next, we set up two goals for time-series forecasting, which require not only precise, but also informative forecasting Wu et al. (2021); Zhou et al. (2022); Le Guen & Thome (2019) as follow:

- Mapping function f should be learnt to point-wisely reduce distance between $\hat{\mathbf{Y}}$ and $\mathbf{Y}$; and
- The output $\hat{\mathbf{Y}}$ should have similar temporal dynamics with $\mathbf{Y}$.

Temporal dynamics are informative patterns in time-series, such as rise, drop, trough, peak, and plateau. The optimization for the point-wise distance is the conventional methods utilized in deep learning domain, which could be obtained by using MAE or MSE. But in the real-world problem, for example, the traffic speed prediction or the economics, accurate forecasting of such "temporal dynamics." Esling & Agon (2012) also emphasized the importance of measuring temporal dynamics,

Figure 2: Example of the six distortions on the amplitude axis (top) and temporal axis (bottom).

as "...allowing the recognition of perceptually similar objects even though they are not mathematically identical." In this paper, we define the temporal dynamics as follows:

Definition 2. *Temporal dynamics (or shapes) are the informative periodic and non-periodic patterns in time-series data.*

In this work, we aim to design a shape-aware loss function that satisfies both goals. To this end, we first discuss distortions that two time-series with similar shapes can have.

Definition 3. *Given two time-series* $\mathbf{F}$ *and* $\mathbf{G}$ *having a similar shape but not being mathematically identical,* $\mathbf{F}$ *could be formulated by transformation* $\mathcal{H}(\mathbf{G})$*. Then, we can call time-series* $\mathbf{F}$ *and* $\mathbf{G}$ *have a distortion, which could be represented by transformation* $\mathcal{H}$*.*

Distortion generally occurs in different aspects. Distortions are defined as temporal distortion (i.e., *warping*) and amplitude distortion (i.e.,*scaling*) with respect to its relevance of dimension, time and amplitude. Existing distortion in data leads to misbehavior of the model, as measurements are interrupted by the distortion. For example, if we have two time-series $\mathbf{F}$ and $\mathbf{G} = \mathbf{F} + k$, which have a similar shape but different means, $\mathbf{G}$ could represent many temporal dynamics of $\mathbf{F}$. However, measurements often evaluate $\mathbf{F}$ and $\mathbf{G}$ are different (e.g., measuring with MSE) and causes misguidance of the model in training. As such, it is important to have measurements that consider similar shape invariant to distortion. We define a measurement for a distortion as follow:

Definition 4. *Let transformation* $\mathcal{H}$ *represents a distortion* H*. Then, we call measurement* $\mathcal{D}$ *invariant to* $\mathcal{H}$*, if* $\exists\delta > 0 : \mathcal{D}(\mathbf{T}, \mathcal{H}(\mathbf{T})) < \delta$ *for any time-series* $\mathbf{T}$*.*

3.2 Time-Series Distortions in Transformation Perspectives

Distortion, a gap between two similar time-series, affects on capturing shapes in time-series data. As such, it is important to investigate different distortions and their impact on representation learning aspects. There are six common time-series distortions that models encounter during learning (Esling & Agon, 2012; Batista et al., 2014; Berkhin, 2006; Warren Liao, 2005; Kerr et al., 2008)–Amplitude Shifting, Phase Shifting, Uniform Amplification, Uniform Time Scaling, Dynamic Amplification, and Dynamic Time Scaling. Next, we explain each common time-series distortion in terms of transformation with n-length time-series $\mathbf{F}(t) = [f(t_1), f(t_2), \ldots, f(t_n)]$, where t = $[t_1, t_2, ..., t_n]$. Fig. 2 presents example distortions, categorized by amplitude and time dimensions.

- *Amplitude Shifting* describes how much a time-series shifts against another time-series. This can be described with two time-series and the degree of shifting (k): $\mathbf{G}(t) = \mathbf{F}(t) + k = [f(t_1) + k, \ldots, f(t_n) + k]$, where $k \in \mathbb{R}$ is constant.
- *Phase Shifting* is the same type of transformation (i.e., translation) as amplitude shifting, but it occurs along with the temporal dimension. This distortion can be represented with two time-series functions with the degree of shift (k): $\mathbf{G}(t) = \mathbf{F}(t + k) = [f(t_1 + k), \ldots, f(t_n + k)]$, where $k \in \mathbb{R}$ is constant. Cross-correlation (Paparrizos & Gravano, 2015; Vlachos et al., 2005) is the most popular measure method that is invariant to this distortion.

- *Uniform Amplification* is a transformation that changes the amplitude by multiplication of $k \in \mathbb{R}$. This distortion can be described with two functions and a multiplication factor (k): $\mathbf{G}(t) = k \cdot \mathbf{F}(t) = [k \cdot f(t_1), \ldots, k \cdot f(t_n)]$.
- *Uniform Time Scaling* means a uniformly shortened or lengthened $\mathbf{F}(t)$ on the temporal axis. This distortion can be represented as $\mathbf{G}(t) = [g(t_1), \ldots, g(t_m)]$, where $g(t_i) = f(t_{\lceil k \cdot i \rceil})$ and $k \in \mathbb{R}^+$. Although Keogh et al. (2004) propose uniform time warping methods to handle this distortion, it still remains one of the difficult distortion types to measure, due to the difficulty in finding the scaling factor k without testing all possible cases (Keogh, 2003).
- *Dynamic Amplification* can be interpreted as any distortion occurred by non-zero multiplication on the amplitude dimension. This distortion can be described as follows: $\mathbf{G}(t) = \mathbf{H}(t) \cdot \mathbf{F}(t) = [h(t_1) \cdot f(t_1), \ldots, h(t_n) \cdot f(t_n)]$ with function $h(t)$ such that $\forall_{t \in \mathbb{T}}, h(t) \neq 0$. Local amplification is a representative distortion of this type of distortions, which still remains challenging to solve.
- *Dynamic Time Scaling* means any transformation that dynamically lengthens or shortens signals on the temporal dimension including local time scaling (Batista et al., 2014) and occlusion (Batista et al., 2014; Vlachos et al., 2003). It can be represented as follows: $\mathbf{G}(t) = \mathbf{F}(h(t)) = [f(h(t_1)), \ldots, f(h(t_n))]$, where $h(t)$ is a positive, strictly increasing function. Dynamic time warping (DTW) (Bellman & Kalaba, 1959; Berndt & Clifford, 1994; Keogh & Ratanamahatana, 2005) is the most popular technique on this distortion. Das et al. (1997) also introduce the longest common subsequence (LCSS) algorithm to tackle occlusion, noise, and outliers in this distortion.

There are several studies on shape-aware clustering (Bellman & Kalaba, 1959; Batista et al., 2014; Paparrizos & Gravano, 2015; Berkhin, 2006; Warren Liao, 2005; Kerr et al., 2008) and classification (Xi et al., 2006; Batista et al., 2014; Srisai & Ratanamahatana, 2009) tasks with the consideration of shapes. On the other hand, only a few studies exist for time-series forecasting tasks, including Le Guen & Thome (2019) that utilizes dynamic time warping (DTW) and temporal distortion index (TDI) for modeling temporal dynamics. Next we describe mean square error (MSE) and DILATE, proposed by Le Guen & Thome (2019), and discuss their invariance to the distortions.

3.3 Distortion Handling in Current Time-Series Forecasting Objectives

Many measurement metrics have been used in the time-series forecasting domain, and those based on the L_p distance, including Euclidean distance, are widely used to handle time-series data. However, such metrics do not have invariance to the aforementioned distortions (Ding et al., 2008; Le Guen & Thome, 2019) due to its point-wise mapping. Specifically, since L_p distance compares the values per time step, it cannot handle temporal distortions appropriately and vulnerable to scaling of the data. Le Guen & Thome (2019) propose a loss function, called DILATE, to overcome the inadequate characteristic in the L_p distance metrics by recognizing temporal dynamics with DTW and TDI. In terms of transformation, DILATE handles dynamic time scaling, especially, local time scaling with DTW, and phase shifting with penalized TDI, defined as follows:

$$\mathcal{L}_{DILATE}(\hat{y}_i, y_i) := -\gamma \log \Big(\sum_{\mathbf{A} \in \mathcal{A}_{k,k}} \exp \Big(- \frac{\langle \mathbf{A}, \alpha \Delta(\hat{y}_i, y_i) + (1-\alpha)\mathbf{\Omega} \rangle}{\gamma} \Big) \Big),$$

where $\mathbf{A}$, $\delta(\hat{y}_i, y_i)$, $\mathbf{\Omega}$ are the warping path, cost matrix, and squared penalization matrix, respectively.

While DILATE shows better performance than existing methods, there is a missing point in invariance point of view. DILATE highly depends on DTW, which allows dynamic alignment of the time-series for the predefined window. In such windows, DTW could align the signal regardless of their information, for example, periodicity. As a result, the model makes misbehavior that could cheat DTW within the window, as shown in Fig. 1 bottom. DTW's scale and noise sensitivity are also problematic. Basically, DTW computes the Euclidean distance of two time-series after its temporal alignment in dynamic programming and the alignment relies on the distance function. Consequently, the dynamic alignment of the DTW can be properly achieved only when two time-series have the same range (Esling & Agon, 2012; Bellman & Kalaba, 1959). That means, it hardly achieves invariance on amplitude distortion without appropriate pre-processing. Gong & Chen (2017) also show that DTW poorly matches the prediction and target (i.e., ground-truth) time-series with amplitude shifting. Even

when the target time-series is aligned with normalization, we cannot guarantee that the predicted and target time-series are properly aligned due to DTW's high sensitivity to noise. As a result, DILATE can generate poor alignment results that can cause wrong optimization of TDI, which produces instability during optimization steps and incorrect results. To design an effective shape-aware loss function, we have to understand measures and when the measures have transformation invariances. In the next section, we discuss how we interpret transformations in time-series forecasting point of view and which types of transformations should be considered in objective function design.

3.4 TRANSFORMATION INVARIANCES IN TIME-SERIES FORECASTING

In the time-series domain, data often have various distortions so measurements are needed to satisfy a number of transformation invariances for meaningfully modeling temporal dynamics. As discussed in Sec. 3.1, we set the goal of time-series forecasting as (1) point-wisely reducing the gap between prediction and target time-series and (2) preserving temporal dynamics of the target time-series. To satisfy both of them, we have to consider (1) the method that should not have a negative impact on the traditional goal of accurate time-series forecasting and (2) the distortions that play a crucial role in capturing the temporal dynamics of the target time-series. In this section, we review all six distortions whether it is a feasible loss function or not, discuss their benefits and trade-offs, and find appropriate distortions to be considered in time-series forecasting.

Amplitude Shifting In a wide range of situations, it is beneficial to capture the the trends of time-series sequence in spite of shifts in terms of amplitude. Thus, being invariant to amplitude shifting in a loss function takes many advantages in time-series forecasting: (1) shape-awareness invariant to amplitude shifting, (2) accurate deviation of values in modeling, and (3) effective on-time prediction of the peak or sudden changes. To guarantee the amplitude shifting invariant in the optimization stage, the loss function should induce an equal gap k between prediction and ground truth data in each step. Formally speaking, the loss function with consideration of the amplitude shifting should satisfy:

$$\mathcal{L}(\mathbf{Y}, \hat{\mathbf{Y}}) = 0 \Leftrightarrow \forall_{i \in [1,\dots,n]}, d(y_i, \hat{y}_i) = k, \tag{1}$$

where $k \in \mathbb{R}$ is an arbitrary and equal gap, and $d(y_i, \hat{y}_i)$ is a signed distance with a boundary $y_i > \hat{y}_i$. By allowing tolerance between prediction and target time-series, models can follow trends in time-series instead of tending to predict exact values in point-wise. In short, unlike existing loss functions that handle only point-wise distance (e.g., DTW), we should deal with both the point-wise distance and its relational distance values to guarantee amplitude shifting.

Phase Shifting There are forecasting tasks, whose main objectives concern accurate forecasting of peaks and periodicity in time-series (e.g., heart beat data and stock price data). For such tasks, phase shifting invariance is one of the best solutions for (1) modeling periodicity, regardless of translation on temporal axis and (2) having precise statistics with shapes, such as peak and plateau values. If a loss function is to be invariant to phase shifting, the function should satisfy:

$$\mathcal{L}(\mathbf{Y}, \hat{\mathbf{Y}}) = 0 \Leftrightarrow \mathbf{Y} \text{ and } \hat{\mathbf{Y}} \text{ have the same dominant frequency.} \tag{2}$$

Note Eq. 2 allows a similar shape as target time-series in forecasting, not exactly same shape (e.g., $\sin(x)$ and $2\sin(x + x_0)$ with the same dominant frequency).

Uniform Amplification This proposition will be useful in case of sparse data that contains a significant number of zeros. By adopting the uniform amplification invariance, models are able to focus non-zero sequences, whereas this proposition allows models to receive less penalty in zero sequences. Since it guarantees shape-awareness with a multiplication factor in a timely manner as Fig. 2, invariance for uniform amplification fits well. To have a model trained with the uniform amplification invariance, the loss function should satisfy:

$$\mathcal{L}(\mathbf{Y}, \hat{\mathbf{Y}}) = 0 \Leftrightarrow \forall_{i \in [1,\dots,n]}, \frac{y_i}{\hat{y}_i} = k(\hat{y}_i \neq 0). \tag{3}$$

Uniform Time Scaling, Dynamic Amplification, and Dynamic Time Scaling After careful consideration, we conclude that uniform time scaling, dynamic amplification, and dynamic time scaling are incompatible for optimization. We describe the reason below.

To achieve invariance for the uniform time scaling, the loss function should satisfy:

$$\mathcal{L}(\mathbf{Y}, \hat{\mathbf{Y}}) = 0 \Leftrightarrow \exists c \in \mathbb{Z}^+ : \{c|y_i = \hat{y}_{ci}\} \cup \{c|y_{ci} = \hat{y}_i\} \forall i \in [0, 1, \ldots, T'].$$

This proposition will influence negatively original temporal dynamics, considering that it gives the tolerance of mispredicting periodicity (e.g., daily periodic signals) and even cannot catch events (e.g., abrupt changing values) in timely manner. In summary, it hinders models from capturing shape and corrupts periodic information.

For both dynamic amplification and dynamic time scaling, loss functions always are zero for all pairs when we do not set the limit of tolerance. For example, if we do not limit tolerance, the proposition for dynamic amplification invariance is as follow:

$$\mathcal{L}(\mathbf{Y}, \hat{\mathbf{Y}}) = 0 \Leftrightarrow \forall c_i \in \mathbb{R} : y_i = c_i \hat{y}_i,$$

If a loss function satisfies the proposition, it is always zero because there always exists $c_i = y_i / \hat{y}_i$ except $\hat{y}_i = 0$. Therefore, it is not able to give any information because all random values could be an optimal solution. The same situation happens with the dynamic time scaling if we do not limit the window. Consequently, all of the uniform time scaling, dynamic amplification, and dynamic time scaling are unsuitable to be objectives in time-series forecasting.

4 METHODS

In this section, we describe a novel loss function TILDE-Q (a Transformation Invariant Loss function with Distance EQuilibrium), which allows models to perform shape-aware time-series forecasting based on the three distortion invariances. To build a transformation invariant loss function, we have to design a loss function that satisfies the proposition for amplitude shifting invariance (Eq. 1), phase shifting invariance (Eq. 2), and uniform amplification shifting invariance (Eq. 3), as discussed in Sec. 3.4. We select them for our loss function because they help models capture the shape and do not harm the goal of the traditional time-series forecasting (i.e., minimize gap between prediction and target time-series). Not only the loss function should satisfy these propositions, but also it should consider correlations between the whole sequence of outputs and ground truths rather than point-wisely optimizing the model. It is not achieved by other loss functions, such as MSE or DILATE. To handle all three distortions and the whole sequence of correlations, we build three objective functions (*a.shift*, *phase*, and *amp* losses) that achieve one or more invariance by utilizing softmax, Fourier coefficient, and auto-correlation to design a loss function.

Amplitude Shifting Invariance with Softmax (Amplitude Shifting) To strengthen amplitude shifting invariance, we design a loss function that satisfies Eq. 1. This means, $d(y_i, \hat{y}_i)$ needs to be the same value for all i. To satisfy the condition, we utilize the softmax function:

$$\mathcal{L}_{a.shift}(\mathbf{Y}, \hat{\mathbf{Y}}) = T' \sum_{i=1}^{T'} |\frac{1}{T'} - \text{Softmax}(d(y_i, \hat{y}_i)))|, \text{Softmax}(d(y_i, \hat{y}_i)) = \frac{e^{d(y_i, \hat{y}_i)}}{\sum_{j=1}^{T'} e^{d(y_j, \hat{y}_j)}} \quad (4)$$

where T', Softmax, and $d(\cdot, \cdot)$ are the length of sequence, softmax function, and signed distance function, respectively. Because the Softmax produces the proportion of each value, it only reaches to the optimal solution when it satisfies Eq. 1. Also, if we utilize Softmax, there is no need to know arbitrary equal gap k.

Invariances with Fourier Coefficients (Phase Shifting) As we discussed in Sec. 3.4, one candidate method to obtain phase shifting invariance is to use Fourier coefficients. As described in prior studies (NG & GOLDBERGER, 2007), we can reconstruct original time-series only with dominant frequencies. In this way, we utilize the norm of dominant Fourier coefficient of ground truth and prediction sequences as our additional objective function, achieving phase shifting invariance. When it comes to the other frequencies, we denote the norm of prediction sequence to reduce the value of Fourier coefficient. Consequently, with the help of our loss function, this loss function allows model to be noise robustness because the Fourier coefficients of white noises in original time-series are relatively small. Simply, we optimize the distance between Fourier coefficients of two time-series as:

$$\mathcal{L}_{phase}(\mathbf{Y}, \hat{\mathbf{Y}}) = \begin{cases} ||\mathcal{F}(\mathbf{Y}) - \mathcal{F}(\hat{\mathbf{Y}})||_p, & \text{if dominant frequency} \\ ||\mathcal{F}(\hat{\mathbf{Y}})||_p, & \text{otherwise} \end{cases} \quad (5)$$

where $||\cdot||_p$ is the L_p norm. To obtain the dominant frequency terms, we calculate the norm of the Fourier coefficient for each frequency and filter them with the squared root of sequence length, $\sqrt{T'}$. We also guarantee the minimum number of dominant frequencies as $\sqrt{T'}$. This loss function obtains uniform amplification invariance by utilizing a normalization technique to Fourier coefficients. For example, $\sin x$ and $c \cdot \sin x$ have the same Fourier coefficients if properly normalized. In summary, by Eq. 5, we could obtain (1) invariance for phase shifting, (2) invariance for uniform amplification, and (3) robustness to noise.

Invariances with auto-correlation (Uniform Amplification) Although Fourier coefficients can be considered as a reasonable solution to catch the periodicity of the target time-series, it is not fully invariant to phase shifting for three reasons–(1) the statistics (e.g., mean and variance) in data keep changing, (2) such changing statistics also cause the changes of Fourier coefficients even in the same frequency, and (3) objectives only with a norm of them cannot fully represent the original time-series. Thus, we introduce an objective based on normalized cross-correlation, which satisfies Eq. 2 for a periodic signal:

$$\mathcal{L}_{amp}(\mathbf{Y}, \hat{\mathbf{Y}}) = ||R(\mathbf{Y}, \mathbf{Y}) - R(\mathbf{Y}, \hat{\mathbf{Y}})||_p, \tag{6}$$

where $R(\cdot, \cdot)$ is a normalized cross correlation function. This loss function helps predicted sequences to mimic label sequences by calculating difference between the auto-correlation of the label sequences and cross-correlation between label and predicted sequences. Therefore, the label and prediction have similar temporal dynamics regardless of phase shifting and uniform amplification.

In summary, we introduce TILDE-Q (Transformation Invariant Loss Function with Distance Equilibrium), combining Eq. 4, Eq. 5, and Eq. 6 as follows:

$$\mathcal{L}_{TILDEq}(\mathbf{Y}, \hat{\mathbf{Y}}) = \alpha\mathcal{L}_{a.shift}(\mathbf{Y}, \hat{\mathbf{Y}}) + (1-\alpha)\mathcal{L}_{phase}(\mathbf{Y}, \hat{\mathbf{Y}}) + \gamma\mathcal{L}_{amp}(\mathbf{Y}, \hat{\mathbf{Y}}), \tag{7}$$

where $\alpha \in [0, 1]$ and γ is hyperparameter.

5 Experiments

In this section, we present the results of our comprehensive experiments, demonstrating the effectiveness of TILDE-Q and importance of transformation invariance.

Experimental Setup We conduct the experiments with four state-of-the-art models, including Informer (Zhou et al., 2021), N-Beats (Oreshkin et al., 2020), Autoformer (Wu et al., 2021), and FEDformer (Zhou et al., 2022) and one simple sequence-to-sequence gated recurrent unit (GRU) model. We use seven real-world datasets–ECG5000, Traffic, ETTh2, ETTm2, ECL, Exchange, and Weather and one synthetic dataset–Synthetic for model training. We repeat each experiment with a model and dataset 10 times in combinations with three different objective functions. Appendix A provides detailed explanations on the datasets, hyperparameter setting, and model architectures.

Table 1: Experimental results of short-term time-series forecasting on the three datasets with sequence-to-sequence GRU model.

Methods	GRU + MSE				GRU + DILATE				GRU + TILDE-Q			
Eval	MSE	DTW	TDI	LCSS	MSE	DTW	TDI	LCSS	MSE	DTW	TDI	LCSS
Synthetic	**0.0107**	3.5080	**1.0392**	0.3523	0.0130	3.4005	1.1242	**0.3825**	0.0119	**3.2873**	1.1564	0.3811
ECG5000	0.2152	1.9718	0.8442	0.7743	0.8270	3.9579	2.0281	0.4356	**0.2141**	**1.9575**	**0.7714**	**0.7773**
Traffic	**0.0070**	1.4628	0.2343	0.7209	0.0095	1.6929	0.2814	0.6806	0.0072	**1.4600**	**0.2276**	**0.7220**

5.1 Experiment Results

Evaluation Metrics In the experiment, we evaluate TILDE-Q with four evaluation metrics: mean squared error (MSE), dynamic time warping (DTW), its corresponding temporal distortion index (TDI), all of which are used in Le Guen & Thome (2019). As DTW is sensitive to noise and generates incorrect paths when one of the time-series data is noisy (as discussed in Sec. 3.3), we additionally use the longest common subsequence (LCSS) for comparison, which is more robust to outliers and

Table 2: Experimental results on six real-world datasets (four cases) with four SOTA models and three training metrics. For all experiment, we set input sequence length $T = 96$.

Model		N-Beats						Informer						Autoformer						FEDformer					
Methods		MSE		DILATE		TILDE-Q		MSE		DILATE		TILDE-Q		MSE		DILATE		TILDE-Q		MSE		DILATE		TILDE-Q	
Metric		MSE	LCSS	MSE	LCSS	MSE	LCSS	MSE	LCSS	MSE	LCSS	MSE	LCSS	MSE	LCSS	MSE	LCSS	MSE	LCSS	MSE	LCSS	MSE	LCSS	MSE	LCSS
ETTh2	96	0.187	0.468	0.310	0.487	**0.155**	**0.586**	0.246	0.463	0.328	0.503	**0.176**	**0.537**	0.153	0.618	0.221	0.531	**0.149**	**0.631**	**0.130**	**0.669**	0.191	0.526	0.138	0.662
	192	0.239	0.450	0.618	0.463	**0.173**	**0.581**	0.281	0.425	0.408	**0.489**	**0.243**	0.431	**0.197**	**0.601**	0.282	0.533	0.207	0.598	**0.182**	**0.623**	0.269	0.526	0.199	0.612
	336	0.289	0.454	1.140	0.458	**0.213**	**0.537**	0.308	0.443	0.416	**0.506**	**0.295**	0.416	0.239	0.595	0.375	0.525	**0.236**	**0.597**	**0.230**	**0.605**	0.351	0.509	0.238	0.604
	720	0.388	0.438	1.671	0.457	**0.304**	**0.528**	**0.287**	0.442	0.422	**0.481**	0.315	0.426	0.285	0.577	0.429	0.492	**0.237**	**0.579**	**0.278**	**0.591**	0.433	0.509	0.287	0.581
ETTm2	96	**0.079**	0.672	0.152	0.437	0.095	**0.690**	0.088	0.738	0.126	0.512	**0.087**	**0.781**	0.099	0.675	0.113	0.593	**0.094**	**0.707**	0.068	0.787	0.115	0.632	**0.067**	**0.792**
	192	**0.122**	0.576	0.205	0.510	0.128	**0.616**	**0.115**	0.670	0.234	0.526	0.131	**0.698**	0.134	0.651	0.185	0.550	**0.125**	**0.681**	0.098	0.734	0.185	0.539	**0.097**	**0.738**
	336	0.182	0.458	0.250	0.481	**0.170**	**0.619**	0.186	0.636	0.280	0.502	**0.176**	**0.655**	0.158	0.603	0.200	0.537	**0.154**	**0.616**	0.133	0.667	0.249	0.505	**0.127**	**0.682**
	720	0.237	0.492	0.417	0.583	**0.233**	**0.707**	0.216	0.576	0.374	0.474	**0.206**	**0.586**	0.199	0.606	0.266	0.500	**0.188**	**0.627**	0.196	0.626	0.291	0.481	**0.182**	**0.636**
ECL	96	0.366	0.658	1.115	0.507	**0.318**	**0.722**	**0.270**	0.703	0.985	0.632	0.280	**0.727**	0.420	0.648	0.681	0.625	**0.351**	**0.691**	**0.253**	**0.732**	0.479	0.694	0.264	0.727
	192	0.430	0.621	1.185	0.497	**0.338**	**0.718**	**0.279**	0.706	1.120	0.605	0.307	**0.733**	0.420	0.657	0.731	0.611	**0.403**	**0.668**	0.295	0.731	0.549	0.681	**0.282**	**0.734**
	336	0.519	0.596	1.246	0.509	**0.383**	**0.711**	**0.320**	**0.722**	1.233	0.569	0.327	0.714	**0.462**	**0.653**	0.789	0.609	0.463	0.642	**0.331**	0.721	0.697	0.689	0.339	**0.730**
	720	0.624	0.571	1.306	0.533	**0.454**	**0.696**	0.641	0.456	1.370	0.550	**0.467**	**0.629**	**0.500**	0.618	0.863	0.607	0.504	**0.642**	0.396	0.696	0.774	0.640	**0.394**	**0.701**
Exchange	96	0.450	0.442	0.394	0.432	**0.275**	**0.447**	0.353	**0.469**	**0.326**	0.468	0.526	0.455	0.247	0.458	0.192	**0.465**	**0.173**	0.458	0.144	0.435	0.388	0.444	**0.122**	**0.470**
	192	**1.216**	0.416	1.568	0.406	1.662	**0.435**	**0.968**	0.465	0.974	0.458	1.285	**0.496**	0.325	0.432	0.473	0.412	**0.295**	**0.443**	**0.269**	0.420	0.591	0.419	0.296	**0.447**
	336	**1.453**	0.413	3.678	0.387	1.843	**0.460**	**1.371**	0.468	1.673	0.443	1.691	**0.493**	0.548	0.328	0.803	0.311	**0.533**	**0.332**	**0.492**	0.414	0.752	0.397	0.590	**0.434**
	720	**1.856**	0.407	3.901	0.340	2.849	**0.462**	**1.764**	0.468	1.829	**0.529**	1.913	0.510	1.362	**0.236**	1.494	0.230	**1.199**	0.223	1.212	0.384	1.511	0.376	**1.170**	**0.393**
Traffic	96	0.234	0.830	2.332	0.525	**0.229**	**0.837**	0.261	0.833	2.961	0.731	**0.228**	**0.849**	0.256	0.876	0.483	0.852	**0.227**	**0.888**	0.207	0.882	0.353	0.861	**0.187**	**0.898**
	192	**0.301**	0.792	2.563	0.552	0.335	**0.803**	0.292	0.816	2.998	0.739	**0.275**	**0.825**	0.260	0.878	0.565	0.819	**0.250**	**0.882**	0.205	0.895	1.468	0.859	**0.196**	**0.898**
	336	**0.345**	0.792	2.460	0.521	0.399	**0.821**	0.311	0.811	2.970	0.712	**0.299**	**0.817**	0.247	**0.880**	0.816	0.805	**0.242**	0.876	0.214	**0.902**	2.974	0.852	**0.206**	0.895
	720	**0.430**	0.796	2.352	0.518	0.448	**0.809**	**0.347**	**0.815**	2.685	0.587	0.386	0.775	**0.272**	**0.871**	1.073	0.818	0.284	0.868	**0.229**	**0.892**	3.083	0.858	0.231	0.873
Weather	96	0.004	0.407	0.002	0.426	**0.001**	**0.517**	0.004	0.456	0.007	0.516	**0.002**	**0.560**	0.017	0.482	0.002	0.531	**0.001**	**0.546**	0.007	0.526	0.002	0.554	**0.001**	**0.579**
	192	0.006	0.421	0.003	0.431	**0.002**	**0.508**	0.003	0.452	0.004	0.470	**0.003**	**0.552**	0.007	0.494	0.003	**0.542**	**0.002**	0.535	0.006	0.542	0.003	**0.600**	**0.002**	0.586
	336	0.006	0.424	0.009	0.358	**0.003**	**0.507**	0.005	0.445	0.005	0.488	**0.004**	**0.567**	0.005	0.490	0.003	0.485	**0.002**	**0.525**	0.005	0.526	0.005	0.480	**0.002**	**0.578**
	720	0.007	0.432	0.153	0.398	**0.003**	**0.508**	0.006	0.448	0.074	0.514	**0.005**	**0.569**	0.008	0.474	0.011	0.472	**0.002**	**0.510**	0.006	0.491	0.003	0.489	**0.002**	**0.574**
Count		8	0	0	0	**16**	**24**	9	3	1	4	**14**	**17**	4	4	0	1	**20**	**17**	9	7	0	1	**15**	**16**

noise (Esling & Agon, 2012). The longer the length of matched subsequences is achieved, the better performance LCSS shows in modeling the shapes. For the state-of-the-art models, we reports the MSE and LCSS. For the detailed results including DTW and TDI, please refer to Appendix B.

Results and Analysis Table 1 shows the results of short-term forecasting performance of gated recurrent unit (GRU) optimized with MSE, DILATE, and TILDE-Q metrics. Synthetic, ECG5000, and Traffic datasets are used for the experiment. With the Synthetic dataset, every used metric shows its own benefits. This result indicates that similarity of the shape and MSE measures have a clear advantage when a model is trained and evaluated with themselves. Also, since the model is evaluated with real-world datasets, it is revealed TILDE-Q outperforms other objective functions in most evaluation metrics. These results indicate our approach for learning shapes in time-series data works better than existing methods for forecasting. DILATE does not show impressive performance with ECG5000 due to its high sensitiveness to noise, as discussed in Sec. 3.3.

Table 2 summarize the experiment results with four state-of-the-art models, N-Beats, Informer, Autoformer, and FEDformer. The models make predictions for both short-term (L=96) and long-term (L up to 720), so that we can investigate their performances with different forecasting difficulties. In most of datasets, the models with TILDE-Q outperform those with other training metrics. Especially for long-term forecasting, we observe that for N-Beats and Informer with TILDE-Q significantly improve the performance with the other metrics. We provide some visual examples in Appendix B and more detailed analysis, qualitative experiments with example visualizations, ablation study results in Appendix B. This result implies that TILDE-Q improves performances of the models in learning temporal dynamics, including LCSS of N-Beats (improved over 10%).

6 Conclusion and Future Work

We propose TILDE-Q, a transformation invariant loss function with distance equilibrium, which allows shape-aware time-series forecasting in a timely manner. To design TILDE-Q, we review existing transformations in time-series data and discuss the conditions that ensure transformation invariances during optimization tasks. The designed TILDE-Q ensures a model to be invariant to the amplitude shifting, phase shifting, and uniform amplification so that a model better captures the shape in time-series data. To prove the effectiveness of TILDE-Q, we conduct comprehensive experiments with state-of-the-art models and real-world datasets. The results indicate that the model trained with TILDE-Q generates more timely, robust, accurate, and shape-aware forecasting in both short-term to long-term forecasting tasks. We conjecture that this work can facilitate future research on transformation invariances and shape-aware forecasting.

REFERENCES

Abubakar Abid and James Y Zou. Learning a warping distance from unlabeled time series using sequence autoencoders. In *Advances in Neural Information Processing Systems*, volume 31, pp. 10568–10578, 2018.

B.R. Bakshi and G. Stephanopoulos. Representation of process trends—iv. induction of real-time patterns from operating data for diagnosis and supervisory control. *Computers & Chemical Engineering*, 18(4):303–332, 1994.

Gustavo E. A. P. A. Batista, Eamonn J. Keogh, Oben Moses Tataw, and Vinícius M. A. de Souza. CID: an efficient complexity-invariant distance for time series. *Data Mining and Knowledge Discovery*, 28(3):634–669, 2014. doi: 10.1007/s10618-013-0312-3.

R. Bellman and R. Kalaba. On adaptive control processes. *IRE Transactions on Automatic Control*, 4 (2):1–9, 1959.

Pavel Berkhin. A survey of clustering data mining techniques. In *Grouping Multidimensional Data - Recent Advances in Clustering*, pp. 25–71. Springer, 2006.

Donald J. Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, AAAIWS'94, pp. 359–370. AAAI Press, 1994.

Ioana Bica, Ahmed M Alaa, James Jordon, and Mihaela van der Schaar. Estimating counterfactual treatment outcomes over time through adversarially balanced representations. In *International Conference on Learning Representations*, 2020.

George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, and Greta M. Ljung. *Time series analysis: forecasting and control*. John Wiley, 2015.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). In *Proceedings of the International Conference on Learning Representations*, 2016.

Marco Cuturi and Mathieu Blondel. Soft-dtw: A differentiable loss function for time-series. In *Proceedings of the 34th International Conference on Machine Learning*, ICML'17, pp. 894–903, 2017.

Gautam Das, Dimitrios Gunopulos, and Heikki Mannila. Finding similar time series. In *Principles of Data Mining and Knowledge Discovery*, pp. 88–100, 1997.

Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh. The ucr time series archive. *IEEE/CAA Journal of Automatica Sinica*, 6(6):1293–1305, 2019.

Hui Ding, Goce Trajcevski, Peter Scheuermann, Xiaoyue Wang, and Eamonn Keogh. Querying and mining of time series data: Experimental comparison of representations and distance measures. *Proceedings of the VLDB Endowment*, 1(2):1542–1552, 2008.

Philippe Esling and Carlos Agon. Time-series data mining. *ACM Computing Surveys*, 45(1), 2012.

Zhichen Gong and Huanhuan Chen. Dynamic state warping. *CoRR*, abs/1703.01141, 2017.

Lukasz Kaiser, Ofir Nachum, Aurko Roy, and Samy Bengio. Learning to remember rare events. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Eamonn J. Keogh. Efficiently finding arbitrarily scaled patterns in massive time series databases. In *Knowledge Discovery in Databases: PKDD 2003*, volume 2838 of *Lecture Notes in Computer Science*, pp. 253–265, 2003.

Eamonn J. Keogh and Chotirat (Ann) Ratanamahatana. Exact indexing of dynamic time warping. *Knowledge and Information Systems*, 7(3):358–386, 2005.

Eamonn J. Keogh, Jessica Lin, and Wagner Truppel. Clustering of time series subsequences is meaningless: Implications for previous and future research. In *Proceedings of the IEEE International Conference on Data Mining*, pp. 115–122. IEEE Computer Society, 2003.

Eamonn J. Keogh, Themis Palpanas, Victor B. Zordan, Dimitrios Gunopulos, and Marc Cardle. Indexing large human-motion databases. In *Proceedings of the International Conference on Very Large Data Bases*, pp. 780–791, 2004.

G. Kerr, H.J. Ruskin, M. Crane, and P. Doolan. Techniques for clustering gene expression data. *Computers in Biology and Medicine*, 38(3):283–293, 2008.

Vincent Le Guen and Nicolas Thome. Shape and time distortion loss for training deep time series forecasting models. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Chunggi Lee, Yeonjun Kim, Seungmin Jin, Dongmin Kim, Ross Maciejewski, David Ebert, and Sungahn Ko. A visual analytics system for exploring, monitoring, and forecasting road traffic congestion. *IEEE Transactions on Visualization and Computer Graphics*, 26(11):3133–3146, 2020. doi: 10.1109/TVCG.2019.2922597.

Hyunwook Lee, Seungmin Jin, Hyeshin Chu, Hongkyu Lim, and Sungahn Ko. Learning to remember patterns: Pattern matching memory networks for traffic forecasting. In *International Conference on Learning Representations*, 2022.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *Proceedings of the International Conference on Learning Representations*. OpenReview.net, 2018.

Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. Mem2seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pp. 1468–1478, 2018.

Yasuko Matsubara, Yasushi Sakurai, Willem G. van Panhuis, and Christos Faloutsos. FUNNEL: automatic mining of spatially coevolving epidemics. In *The ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 105–114. ACM, 2014.

Arthur Mensch and Mathieu Blondel. Differentiable dynamic programming for structured prediction and attention. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 3462–3471, 2018.

JASON NG and JEFFREY J GOLDBERGER. Understanding and interpreting dominant frequency analysis of af electrograms. *Journal of Cardiovascular Electrophysiology*, 18(6):680–685, 2007.

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

John Paparrizos and Luis Gravano. K-shape: Efficient and accurate clustering of time series. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, SIGMOD '15, pp. 1855–1870, 2015. doi: 10.1145/2723372.2737793.

M.H. Pesaran, D. Pettenuzzo, and A. Timmermann. Forecasting time series subject to multiple structural breaks. Cambridge Working Papers in Economics 0433, Faculty of Economics, University of Cambridge, 2004.

Dararat Srisai and Chotirat Ann Ratanamahatana. Efficient time series classification under template matching using time warping alignment. In *Proceedings of the International Conference on Computer Sciences and Convergence Information Technology*, pp. 685–690, 2009.

Daniel Stoller, Mi Tian, Sebastian Ewert, and Simon Dixon. Seq-u-net: A one-dimensional causal u-net for efficient sequence modelling. In Christian Bessiere (ed.), *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 2893–2900. ijcai.org, 2020.

Sainbayar Sukhbaatar, arthur szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.

Michail Vlachos, Marios Hadjieleftheriou, Dimitrios Gunopulos, and Eamonn Keogh. Indexing multi-dimensional time-series with support for multiple distance measures. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, pp. 216–225, 2003.

Michail Vlachos, Philip S. Yu, and Vittorio Castelli. On periodicity detection and structural periodic similarity. In *Proceedings of the SIAM International Conference on Data Mining*, pp. 449–460, 2005.

T. Warren Liao. Clustering of time series data—a survey. *Pattern Recognition*, 38(11):1857–1874, 2005.

Andreas S. Weigend and Neil A. Gershenfeld. *Time Series Prediction: Forecasting the Future and Understanding the Past*. Addison-Wesley, 1994. ISBN 0-201-62601-2.

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

Xiaopeng Xi, Eamonn Keogh, Christian Shelton, Li Wei, and Chotirat Ann Ratanamahatana. Fast time series classification using numerosity reduction. In *Proceedings of the International Conference on Machine Learning*, ICML '06, pp. 1033–1040. Association for Computing Machinery, 2006.

Fisher Yu, Vladlen Koltun, and Thomas A. Funkhouser. Dilated residual networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 636–644. IEEE Computer Society, 2017.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12):11106–11115, 2021.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 27268–27286, 2022.

Yunyue Zhu and Dennis Shasha. Statstream: Statistical monitoring of thousands of data streams in real time. In *Proceedings of the International Conference on Very Large Databases*, pp. 358–369. Morgan Kaufmann, 2002.

A DETAILED EXPERIMENT SETUP

Dataset In our experiment, we utilize six datasets – Synthetic, ECG5000, and Traffic dataset for the simple model (i.e., Sequence-to-Sequence Gated Recurrent Unit) and ETTh2, ETTm2, and Electricity for the state-of-the-art model (i.e., Informer and N-Beats). For each dataset, we describe some metadata of them and experimental setting, including the input length n and prediction window L.

Synthetic: As Le Guen & Thome (2019) describe, the Synthetic dataset is an artificial dataset for measuring model performance on sudden changes (step functions) with an input signal composed of two peaks. The amplitude and temporal position of the two peaks are randomly selected. Then the selected position and amplitude of the step are determined by a peak position and amplitude. We use 500 time-series for training, 500 for validation and 500 for testing. For the Synthetic dataset, we set input length as $n = 20$ and prediction window as $L = 40$. The generation code is provided in DILATE Github[1].

ECG5000: This dataset is originally a 20-hour long ECG (Electrocardiogram), downloaded from Physionet[2] and archived in UCR Time Series Classification Archive (Dau et al., 2019). The data is split by each heartbeat and processed to be in equal lengths (140). In the training, we use 500 for training, 500 for validation, and 4000 for testing. We take first $n = 84$ steps as input and predict last $L = 56$ steps.

Traffic: Traffic dataset is a collection of 48 months (2015-2016) hourly road occupancy rate (between 0 to 1) data from the California Department of Transportation[3]. As Le Guen & Thome (2019) do, we utilize univariate series of the first sensor, a total of 17544 data points. We set our problem as forecasting $L = 24$ future occupancy rates with $n = 168$ historical data (past week). We use 60% of the data for training, 20% for validation, and the rest for evaluation.

ETT: The ETT (Electricity Transformer Temperature) dataset, published by Zhou et al. (2021), is 2-year data collected from two separated counties in China, including ETTh2 and ETTm2 datasets. Each data point has a target value of "oil temperature" and other 6 power load features. ETTh2 and ETTm2 datasets have 1-hour and 15-minute intervals, respectively. As Zhou et al. (2021) do, we split them into 12/4/4 months for the training/validation/testing. Detailed settings, such as the input and output length and hyperparameter setting, are based on the information at Informer Github[4].

ECL: The ECL (Electricity Consuming Load) is a dataset recorded in kWh every 15-minutes from 2012 to 2014, for 321 clients. In our experiment, we split them into 15/3/4 months for the train/validation/test, as Zhou et al. (2021) do. Note that we use the same hyperparameter settings in the ETTh2 dataset.

Deep Learning Model Architectures We perform experiments with three different model architectures, including Sequence-to-Sequence GRU, Informer, and N-Beats. To induce models to predict future time-series in a timely manner, we set $\alpha = 0.5$ and $\gamma = 0.01$ for TILDE-Q. Other training metrics, including MSE and DILATE, are used as described in their original papers. All models are trained with Early Stopping and ADAM optimizer.

Sequence-to-Sequence GRU To evaluate TILDE-Q in simple model, we utilize one layer Sequence-to-Sequence GRU model. For the training of the GRU model, we set learning rate of $1e-3$, hidden size of 128, trained by maximum 1000 epochs with Early Stopping and ADAM optimizer.

Informer When we train Informer with ETTh2, ETTm2, and ECL dataset, we utilize the official code and hyperparameter setting. In the case of ECL dataset, as author answered in their official code[4], we utilize same hyperparameter and dataset splitting criteria as ETTh2 dataset.

N-Beats For N-Beats, we utilize two generic blocks with the hidden size of 128. Additionally, we set the learning rate as $1e-3$ for all three datasets.

[1] https://github.com/vincent-leguen/DILATE

[2] https://physionet.org/

[3] http://pems.dot.ca.gov

[4] https://github.com/zhouhaoyi/Informer2020

Autoformer For Autoformer[5], we use the official code and hyperparameter setting. For the ETTh2 dataset, we utilize hyperparameter settings described in the official code of FEDFormer[6].

FEDformer For FEDformer[6], we use the official code and hyperparameter setting.

B Additional Evaluations

B.1 Detailed Experiment Results and Analysis

Table 3: Detailed experimental results on six real-world datasets (four cases) with N-Beats.

Methods		N-Beats + MSE				N-Beats + DILATE				N-Beats + TILDE-Q			
Metric		MSE	DTW	TDI	LCSS	MSE	DTW	TDI	LCSS	MSE	DTW	TDI	LCSS
ETTh2	96	0.1869	7.2379	2.3787	0.4688	0.3105	6.5849	3.6490	0.4879	**0.1557**	**5.1011**	**1.3240**	**0.5862**
	192	0.2385	11.5667	4.9153	0.4505	0.6186	9.7254	7.0831	0.4637	**0.1738**	**7.6334**	**2.4122**	**0.5819**
	336	0.2889	16.5255	11.5207	0.4544	1.1406	13.7328	14.6986	0.4584	**0.2132**	**11.3351**	**5.3556**	**0.5373**
	720	0.3881	24.1570	18.8462	0.4381	1.6713	19.4392	23.7028	0.4575	**0.3044**	**17.6006**	**9.6636**	**0.5287**
ETTm2	96	**0.0790**	**3.9685**	**2.0436**	0.6721	0.1524	7.9302	5.5597	0.4379	0.0952	4.0110	2.1939	**0.6902**
	192	**0.1224**	6.8695	**3.2834**	0.5762	0.2055	10.0393	8.5602	0.5107	0.1286	**6.3556**	4.9798	**0.6160**
	336	0.1824	12.1438	8.5915	0.4587	0.2501	12.6342	16.1473	0.4819	**0.1705**	**8.9377**	**8.3539**	**0.6195**
	720	0.2370	22.8676	17.8458	0.4929	0.4170	17.7764	24.6877	0.5836	**0.2336**	**14.2715**	**19.0883**	**0.7070**
ECL	96	0.3666	3.5207	0.2989	0.6589	1.1156	5.1430	2.6613	0.5074	**0.3183**	**2.9707**	**0.4844**	**0.7229**
	192	0.4307	5.7578	0.4253	0.6212	1.1859	7.3406	2.8488	0.4973	**0.3383**	**4.1817**	**0.4229**	**0.7187**
	336	0.5199	8.5563	0.5384	0.5965	1.2460	9.5096	3.0517	0.5091	**0.3831**	**5.6643**	**0.3024**	**0.7112**
	720	0.6240	13.9436	0.6510	0.5717	1.3061	13.1928	3.7279	0.5337	**0.4540**	**8.9997**	**0.3251**	**0.6960**
Exchange	96	0.4496	8.6395	**4.3197**	0.4424	0.3945	8.9661	4.3286	0.4316	**0.2748**	**7.9744**	5.2964	**0.4467**
	192	**1.2161**	12.1857	10.5166	0.4157	1.5684	13.0560	9.2434	0.4061	1.6629	**11.5557**	**8.7896**	**0.4348**
	336	**1.4529**	14.7085	19.0407	0.4130	3.6784	17.5189	**17.2512**	0.3871	1.8432	**12.5648**	20.8871	**0.4603**
	720	**1.8563**	21.7347	**50.6751**	0.4073	3.9008	26.7020	74.0546	0.3400	2.8487	**19.1588**	53.8069	**0.4619**
Traffic	96	0.2349	2.1046	0.0216	0.8303	2.3325	3.9657	1.2052	0.5250	**0.2286**	**2.0699**	**0.0207**	**0.8371**
	192	**0.3014**	3.4040	0.0142	0.7916	2.5627	5.4169	1.1355	0.5515	0.3352	**3.2559**	**0.0119**	**0.8028**
	336	**0.3455**	4.6409	0.0088	0.7918	2.4599	8.2828	1.3377	0.5208	0.3990	**4.2622**	**0.0066**	**0.8206**
	720	**0.4298**	7.0561	0.0045	0.7958	2.3522	12.6258	0.9967	0.5177	0.4480	**6.7344**	**0.0034**	**0.8085**
Weather	96	0.0042	9.3228	5.9134	0.4072	0.0023	8.9289	**5.0617**	0.4256	**0.0010**	**6.5198**	6.0450	**0.5168**
	192	0.0056	10.9682	11.9549	0.4212	0.0030	12.8164	10.6858	0.4307	**0.0017**	**8.8391**	**9.0867**	**0.5076**
	336	0.0058	13.3578	14.6572	0.4243	0.0087	20.4895	23.7903	0.3579	**0.0026**	**11.8682**	**9.6758**	**0.5074**
	720	0.0068	18.5861	22.1432	0.4315	0.1534	28.6021	47.4488	0.3982	**0.0029**	**17.1895**	**19.1942**	**0.5078**

At first, we observe that the model optimized with TILDE-Q outperforms the same model optimized with other objective functions in both short- and long-term forecasting tasks. An interesting point in the results is the large increased errors of TDI and DTW with long-term forecasting. For example, TDI of Informer with DILATE shows dramatically increased error with ECL dataset, as the forecasting window increases, while LCSS does not produce such large increased error. We attribute this to the weakness of DTW-based loss functions, which have a weakness due to high sensitiveness on noise. In contrast, TILDE-Q does not show such large performance drop and even achieves better performance in the long-term forecasting (e.g., Table 4, ETTh2). Additionally, we can find that Informer with TILDE-Q on ECL data and N-Beats with TILDE-Q on all three datasets show significant improvements. It indicates that TILDE-Q success to model *shape*, but other metrics could not. We provide additional qualitative results below.

Next, we present qualitative analysis of the results. Fig. 3 shows how the model with different training metrics forecast with different datasets. From the figure, we have noticed that TILDE-Q allows the model to generate more robust, shape-aware forecasting, regardless of the amplitude shifting, phase shifting, and uniform amplification. For example, in the case of N-Beats (Fig. 3 (b) bottom), TILDE-Q generate forecasting results, which are more robust, shape-aware prediction compared to other metrics. We also see the strength in the Informer case (Fig. 3 (b), top). Even when the model has not enough ability to capture shape, TILDE-Q tries to retrieve the shape. We provide additional qualitative results with visualization below. When the model have enough ability to capture shape (i.e., except ETTh2, Informer of $T' \in [192, 336, 720]$), TILDE-Q shown its noise-robust,

[5] https://github.com/thuml/Autoformer

[6] https://github.com/MAZiqing/FEDformer

Table 4: Detailed experimental results on six real-world datasets (four cases) with Informer.

Methods		Informer + MSE				Informer + DILATE				Informer + TILDE-Q			
Metric		MSE	DTW	TDI	LCSS	MSE	DTW	TDI	LCSS	MSE	DTW	TDI	LCSS
ETTh2	96	0.2466	6.9254	3.6676	0.4633	0.3284	6.3109	3.5838	0.5037	**0.1768**	**5.8437**	**1.6734**	**0.5379**
	192	0.2818	10.2654	11.1580	0.4254	0.4086	**8.8262**	**7.1780**	**0.4893**	**0.2432**	10.2134	9.9865	0.4317
	336	0.3089	12.1822	18.7014	0.4434	0.4164	**10.3779**	**13.2580**	**0.5062**	**0.2958**	13.5586	20.2850	0.4165
	720	**0.2877**	17.6369	38.4617	0.4425	0.4229	**14.1196**	**23.9403**	**0.4815**	0.3157	18.4617	43.3238	0.4262
ETTm2	96	0.0889	3.4007	1.5719	0.7386	0.1263	6.0144	2.7757	0.5129	**0.0871**	**3.1354**	**1.3474**	**0.7817**
	192	**0.1157**	5.7964	**2.8128**	0.6705	0.2340	9.7004	7.8354	0.5266	0.1317	**5.7093**	2.9129	**0.6983**
	336	0.1860	**8.9971**	**6.7970**	0.6365	0.2805	11.7889	13.3861	0.5025	**0.1767**	9.0866	7.4023	**0.6555**
	720	0.2165	14.7685	24.6694	0.5768	0.3745	16.7734	29.2783	0.4747	**0.2063**	**15.3057**	**24.1959**	**0.5860**
ECL	96	**0.2709**	**2.8067**	**0.1720**	0.7032	0.9856	3.6394	1.4794	0.6324	0.2800	2.9466	0.2473	**0.7275**
	192	**0.2793**	**4.1193**	**0.1508**	0.7060	1.1209	5.2289	2.1749	0.6053	0.3077	4.2693	0.2978	**0.7336**
	336	0.3203	5.9533	0.1642	0.7222	1.2331	7.8470	3.0415	0.5694	**0.3271**	**5.8090**	**0.1984**	**0.7143**
	720	0.6414	15.8561	4.4284	0.4564	1.3706	12.5981	5.6720	0.5506	**0.4676**	**11.4027**	**0.7107**	**0.6298**
Exchange	96	0.3534	8.0965	**4.8843**	**0.4689**	**0.3260**	**7.7370**	5.6336	0.4678	0.5264	7.9866	6.5120	0.4553
	192	**0.9682**	11.0843	11.3110	0.4647	0.9737	10.8894	15.6770	0.4584	1.2845	**10.4358**	**10.7009**	**0.4959**
	336	**1.3710**	12.8076	18.5937	0.4676	1.6735	12.7034	29.2013	0.4428	1.6912	**12.2349**	**18.2197**	**0.4932**
	720	**1.7586**	22.6852	59.4243	0.4681	1.8292	**16.0093**	**56.8687**	**0.5293**	1.9130	24.0510	62.8152	0.5104
Traffic	96	0.2606	2.0994	0.0208	0.8329	2.9612	2.3355	0.9646	0.7312	**0.2284**	**2.0027**	**0.0194**	**0.8490**
	192	0.2920	3.2573	0.0126	0.8158	2.9978	3.5451	0.8429	0.7394	**0.2753**	**3.1721**	**0.0125**	**0.8248**
	336	0.3109	4.6581	0.0078	0.8115	2.9696	4.9879	1.2672	0.7117	**0.2993**	**4.4715**	**0.0077**	**0.8170**
	720	**0.3472**	**6.7989**	**0.0040**	**0.8146**	2.6845	10.7450	3.4514	0.5874	0.3859	7.5424	0.0051	0.7752
Weather	96	0.0043	8.2890	5.4604	0.4556	0.0069	6.5571	4.7505	0.5159	**0.0021**	**5.5412**	**4.5012**	**0.5602**
	192	0.0031	10.7993	9.2928	0.4523	0.0041	10.5645	9.4713	0.4704	**0.0028**	**8.2535**	**5.8289**	**0.5516**
	336	0.0051	13.8721	22.2699	0.4451	0.0055	12.0586	16.4933	0.4884	**0.0039**	**10.8802**	**10.5220**	**0.5668**
	720	0.0061	21.7720	41.5877	0.4476	0.0737	16.8378	29.8112	0.5142	**0.0047**	**13.9934**	**20.9991**	**0.5689**

Table 5: Detailed experimental results on six real-world datasets (four cases) with Autoformer.

Methods		Autoformer + MSE				Autoformer + DILATE				Autoformer + TILDE-Q			
Metric		MSE	DTW	TDI	LCSS	MSE	DTW	TDI	LCSS	MSE	DTW	TDI	LCSS
ETTh2	96	0.1538	5.2227	2.1865	0.6187	0.2211	6.0453	2.5345	0.5315	**0.1494**	**5.1060**	**1.9752**	**0.6317**
	192	**0.1974**	**7.8730**	**3.3382**	**0.6019**	0.2825	8.6696	5.6671	0.5335	0.2079	7.8917	3.7532	0.5984
	336	0.2393	10.8002	7.3141	0.5954	0.3759	11.0335	13.1347	0.5257	**0.2360**	**10.7212**	**7.0085**	**0.5971**
	720	0.2859	16.3502	15.9233	0.5772	0.4296	15.9819	22.2173	0.4924	**0.2378**	**16.0002**	**13.7906**	**0.5795**
ETTm2	96	0.0990	4.3498	2.5052	0.6756	0.1135	5.3097	**2.2211**	0.5936	**0.0940**	**3.9078**	2.2587	**0.7075**
	192	0.1340	6.3207	3.3676	0.6512	0.1854	8.5209	3.7894	0.5506	**0.1259**	**6.0979**	**2.9278**	**0.6810**
	336	0.1587	**9.4374**	6.9205	0.6036	0.2001	12.0265	8.8305	0.5370	**0.1548**	9.5223	**7.2875**	**0.6169**
	720	0.1999	14.8332	11.9655	0.6064	0.2665	17.8025	17.4114	0.5001	**0.1885**	**14.5844**	**9.9918**	**0.6277**
ECL	96	0.4209	3.5957	0.2461	0.6487	0.6813	3.6490	0.4780	0.6253	**0.3515**	**3.2173**	**0.2298**	**0.6912**
	192	0.4206	4.9924	0.3416	0.6574	0.7319	5.5324	**0.2775**	0.6118	**0.4032**	**4.8581**	0.3301	**0.6680**
	336	**0.4621**	6.6888	0.2795	**0.6535**	0.7895	7.5665	**0.2503**	0.6091	0.4637	**6.7335**	0.3923	0.6429
	720	**0.5005**	10.8571	0.2383	0.6183	0.8630	12.1416	**0.1877**	0.6074	0.5049	**9.8492**	0.2525	**0.6420**
Exchange	96	0.2472	8.2957	5.8340	0.4577	0.1921	8.4651	5.6328	**0.4646**	**0.1730**	**8.2046**	**5.1165**	0.4577
	192	0.3255	11.4212	17.0909	0.4319	0.4732	12.8599	19.0164	0.4124	**0.2955**	**11.3655**	**15.4372**	**0.4433**
	336	0.5483	**15.1853**	**44.4975**	0.3277	0.8035	18.0948	57.5819	0.3114	**0.5331**	16.7350	45.8166	**0.3321**
	720	1.3620	24.6397	145.3080	**0.2357**	1.4936	27.7069	151.6671	0.2302	**1.1993**	**19.5296**	**121.8509**	0.2233
Traffic	96	0.2562	1.9689	0.0178	0.8761	0.4835	1.9044	0.0392	0.8521	**0.2275**	**1.8778**	**0.0168**	**0.8879**
	192	0.2604	2.8922	**0.0091**	0.8780	0.5653	3.0466	0.0343	0.8187	**0.2497**	**2.8793**	0.0116	**0.8817**
	336	0.2474	4.0026	**0.0051**	**0.8797**	0.8155	4.2637	0.0327	0.8047	**0.2422**	**3.9469**	0.0059	0.8760
	720	**0.2720**	6.4371	**0.0030**	**0.8710**	1.0729	**6.0776**	0.0217	0.8176	0.2836	6.1751	0.0034	0.8674
Weather	96	0.0168	7.4658	6.0336	0.4818	0.0019	**5.9775**	4.9688	0.5306	**0.0015**	5.9829	**4.8957**	**0.5461**
	192	0.0069	10.6173	7.4506	0.4941	0.0026	8.3686	**5.4565**	**0.5423**	**0.0017**	**7.7799**	6.1032	0.5355
	336	0.0052	12.5224	13.2607	0.4898	0.0030	12.4524	12.1816	0.4854	**0.0020**	**10.3144**	**9.0025**	**0.5252**
	720	0.0078	18.5079	25.8063	0.4744	0.0115	20.1354	36.7754	0.4721	**0.0023**	**15.2563**	**18.3134**	**0.5102**

smooth forecasting with correctly modeled temporal dynamics. In the most of N-Beats results and some of Informer results, TILDE-Q reveals that these models have enough ability to capture the temporal dynamics with proper loss function. In summary, TILDE-Q proves that it is model-agnostic, noise-robust, and able to capture the shape.

Table 6: Detailed experimental results on six real-world datasets (four cases) with FEDformer.

Methods		FEDformer + MSE				FEDformer + DILATE				FEDformer + TILDE-Q			
Metric		MSE	DTW	TDI	LCSS	MSE	DTW	TDI	LCSS	MSE	DTW	TDI	LCSS
ETTh2	96	**0.1299**	**4.7265**	**1.2607**	**0.6690**	0.1906	6.2294	1.8228	0.5261	0.1381	4.7578	1.3560	0.6621
	192	**0.1819**	7.6178	**2.6979**	**0.6229**	0.2688	8.8422	4.8043	0.5261	0.1988	**7.6174**	2.7712	0.6124
	336	**0.2305**	10.5860	6.7027	**0.6050**	0.3506	11.4834	12.8408	0.5091	0.2382	**10.4108**	**6.6218**	0.6039
	720	**0.2776**	15.7013	**14.7466**	**0.5911**	0.4327	**14.0692**	20.6266	0.5091	0.2871	15.3120	16.4059	0.5808
ETTm2	96	0.0682	3.0962	1.3862	0.7868	0.1147	4.6648	2.2981	0.6325	**0.0669**	**3.0328**	**1.3556**	**0.7918**
	192	0.0976	5.2417	**2.0295**	0.7340	0.1848	8.0678	4.4893	0.5391	**0.0971**	**5.1508**	2.1782	**0.7384**
	336	0.1326	8.3151	5.4619	0.6667	0.2493	13.6349	11.7563	0.5049	**0.1279**	**8.3010**	**4.5488**	**0.6828**
	720	0.1957	14.2579	11.8328	0.6262	0.2913	17.4636	41.9434	0.4806	**0.1822**	**14.1131**	**10.4778**	**0.6361**
ECL	96	**0.2531**	**2.6402**	**0.1436**	**0.7322**	0.4794	2.8685	0.2482	0.6943	0.2638	2.6594	0.1614	0.7265
	192	0.2945	3.8647	0.1831	0.7306	0.5485	4.3313	0.1732	0.6813	**0.2821**	**3.7830**	**0.1277**	**0.7340**
	336	**0.3313**	5.2789	**0.1078**	0.7207	0.6967	5.7911	0.1985	0.6892	0.3385	**5.1763**	0.1229	**0.7290**
	720	0.3956	8.5881	**0.0632**	0.6961	0.7741	10.1163	0.8837	0.6403	**0.3939**	**8.5665**	0.0784	**0.7013**
Exchange	96	0.1437	8.5595	**4.4258**	0.4347	0.3884	8.9178	7.0385	0.4439	**0.1215**	**8.0591**	6.1979	**0.4704**
	192	**0.2694**	12.5168	12.3117	0.4202	0.5912	13.1929	15.4207	0.4187	0.2956	**11.5607**	**12.1302**	**0.4474**
	336	**0.4916**	16.4673	27.0756	0.4140	0.7520	18.1381	33.7878	0.3969	0.5896	**15.9267**	**24.8808**	**0.4342**
	720	1.2115	25.6243	108.0500	0.3838	1.5110	26.7031	91.6313	0.3760	**1.1700**	**24.6190**	**71.8783**	**0.3935**
Traffic	96	0.2074	1.9132	0.0165	0.8824	0.3533	1.8617	0.0291	0.8609	**0.1867**	**1.8386**	**0.0152**	**0.8983**
	192	0.2051	2.7761	0.0085	0.8951	1.4682	2.7545	0.1312	0.8591	**0.1961**	**2.7380**	**0.0083**	**0.8975**
	336	0.2140	**3.7583**	**0.0047**	**0.9023**	2.9741	3.7440	0.1779	0.8519	**0.2059**	3.7834	0.0050	0.8947
	720	**0.2291**	**5.9735**	0.0026	**0.8919**	3.0829	5.8173	0.0949	0.8580	0.2312	6.1080	**0.0022**	0.8735
Weather	96	0.0070	6.1550	4.7039	0.5264	0.0017	5.9507	4.5891	0.5535	**0.0014**	**5.5205**	**4.1085**	**0.5792**
	192	0.0063	7.6906	6.2940	0.5417	0.0020	8.6593	**4.6333**	**0.6000**	**0.0017**	**6.9706**	5.0003	0.5863
	336	0.0046	10.3501	9.4691	0.5261	0.0053	14.8249	9.6239	0.4801	**0.0018**	**9.3016**	**7.4610**	**0.5784**
	720	0.0060	15.2429	24.5379	0.4911	0.0029	16.3929	20.0782	0.4892	**0.0024**	**13.7936**	**15.7668**	**0.5737**

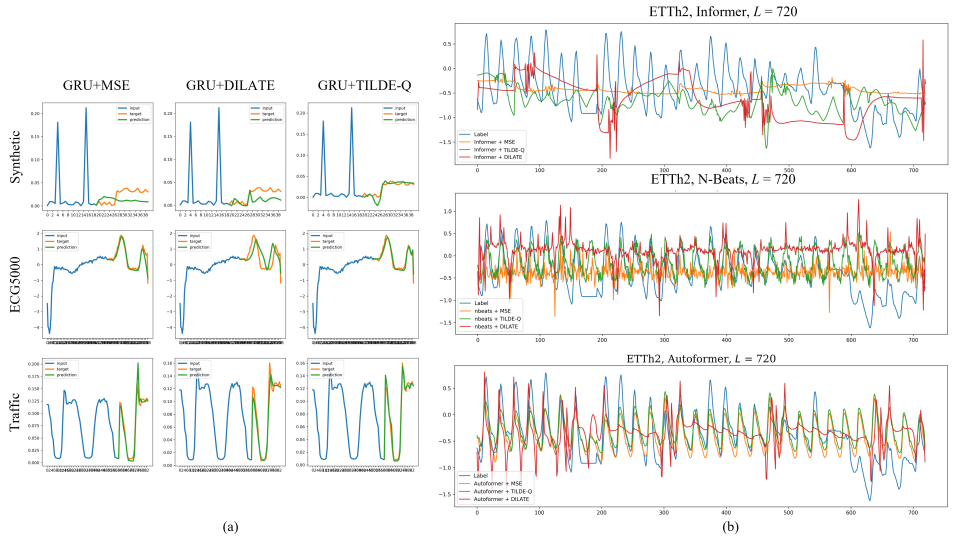

Figure 3: Qualitative results with simple sequence-to-sequence GRU model (a) and state-of-the-art model (b).

B.2 Additional Qualitative Examples

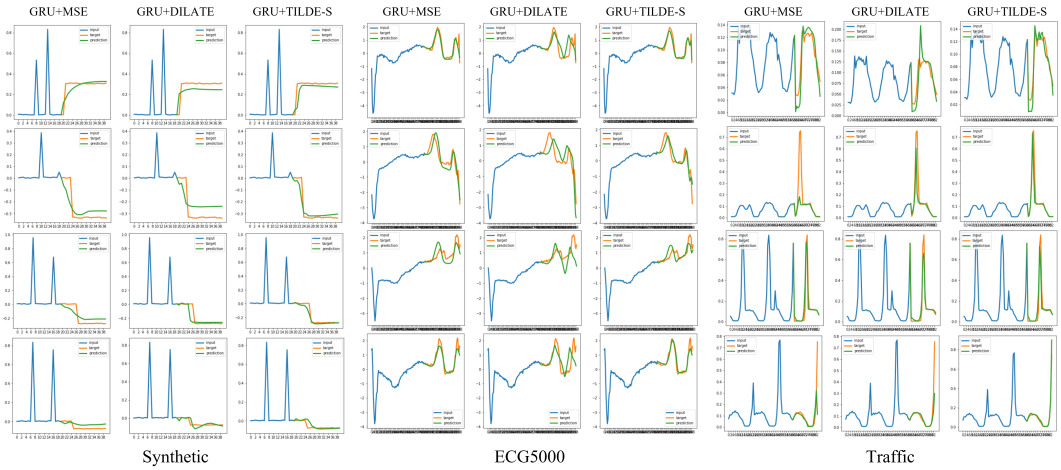

Figure 4: Qualitative results with simple sequence-to-sequence GRU model

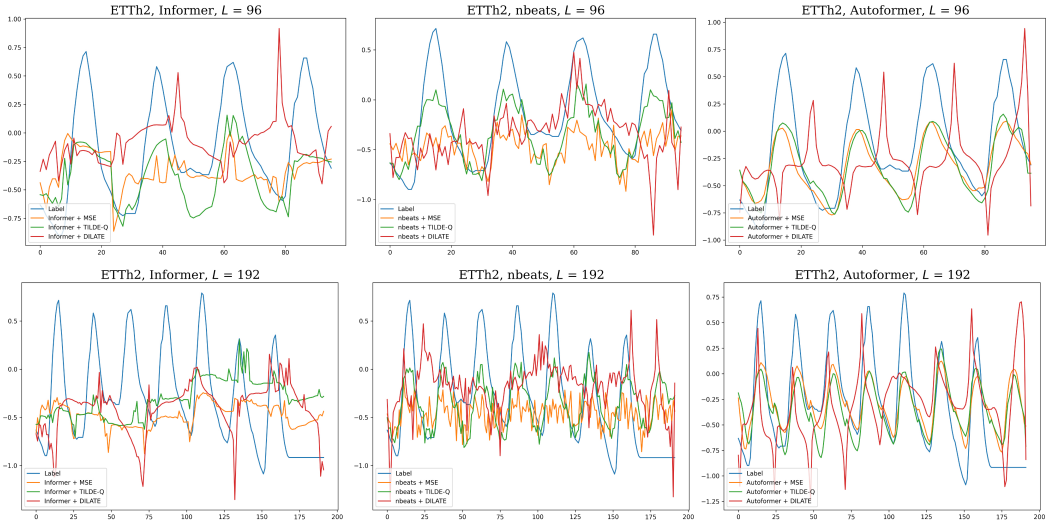

Figure 5: Qualitative results with ETTh2 in short-term forecasting

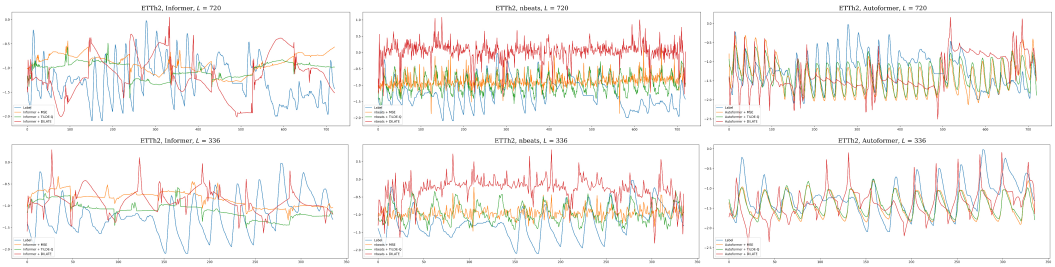

Figure 6: Qualitative results with ETTh2 in long-term forecasting

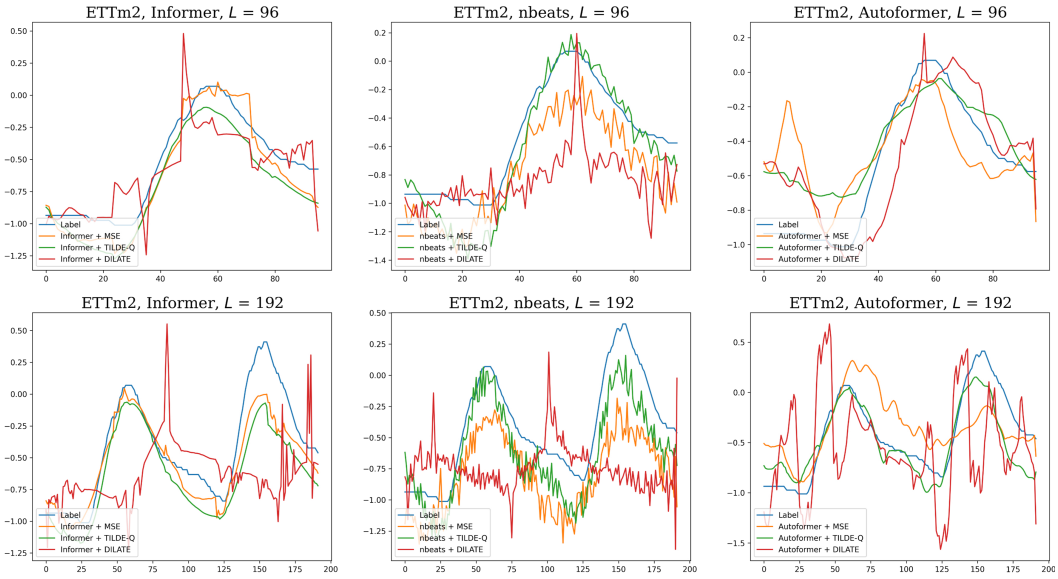

Figure 7: Qualitative results with ETTm2 in short-term forecasting

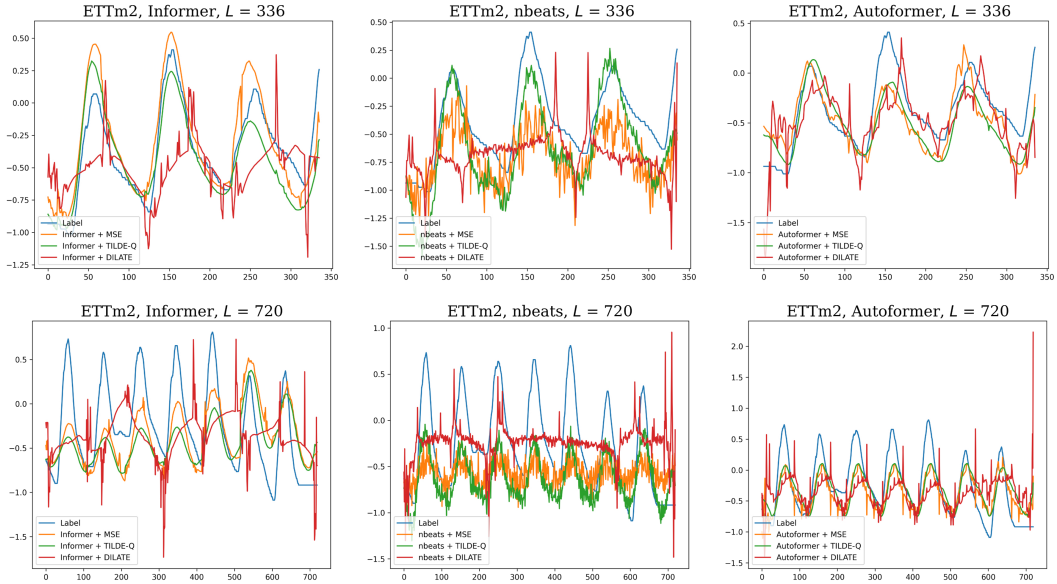

Figure 8: Qualitative results with ETTm2 in long-term forecasting

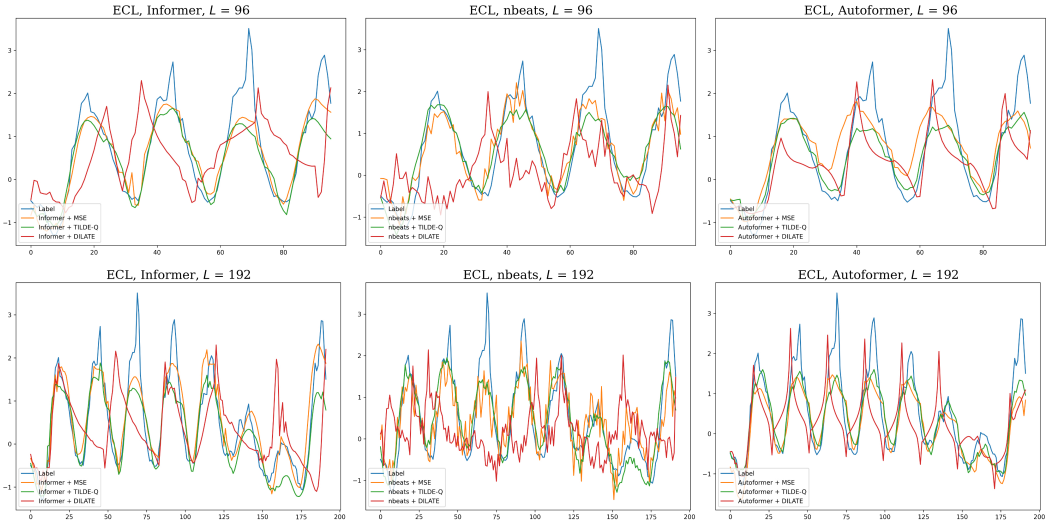

Figure 9: Qualitative results with ECL in short-term forecasting

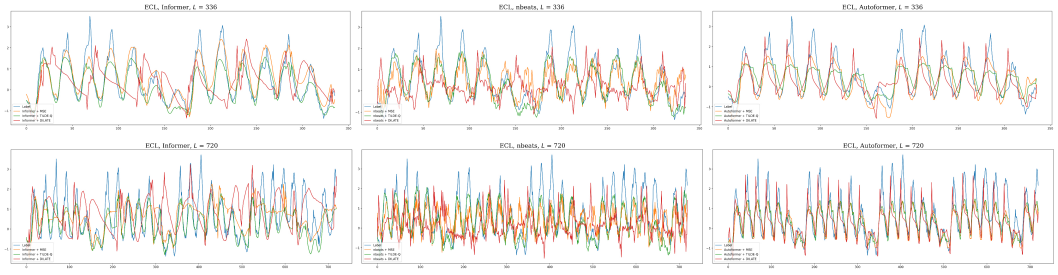

Figure 10: Qualitative results with ECL in long-term forecasting

B.3 ABLATION STUDY

To evaluate the effect of the α, γ, and measure the effect of each loss function, we conduct a set of experiment with ETTh2 dataset and N-Beats on the long-term forecasting problem. As we can see in the Fig. 11, the model tends to predict amplification-free forecasting when α increases. This results indicate our motivation, "$\mathcal{L}_{a.shift}$ will return the forecasting results with same standard deviation with timely manner but without consideration of proper average value."

Furthermore, in the top of Fig. 12, we can observe three things: (1) if we utilize $\mathcal{L}_{a.shift}$ only, as we intended, it have different average (-1.19 vs. 0.11) but relatively similar standard deviation (0.408 vs. 0.299); (2) In the case of $\mathcal{L}_{phase}$ only, they can capture dominant frequency and produce relatively less-noisy forecasting; (3) $\mathcal{L}_{amp}$ have relatively similar average value (-1.195 vs. -0.319), but it has far different standard deviation (0.408 vs. 8.592). In contrast, forecasting results of the model trained with MSE is very noisy and hard to interpret (Fig. 12, bottom). Note that we normalized the results in Fig. 12 because of the scale issue.

In Table 7, we provide how model performances vary with respect to hyperparameters of TILDE-Q. For the default setting, we utilized $\alpha = 0.5, \gamma = 0.01$. Because the design of TILDE-Q mainly focuses on shape modeling, we can see that DTW and LCSS are not critically changing for the hyperparameter. But their trade-offs are revealed in the MSE and TDI. For example, when we decrease α, we can observe TDI increases. It indicates the trade-offs of phase shifting invariance, which has tolerance for non-timely forecasting. Also, we can see that increasing α or γ affects the MSE. When we have $\alpha = 1$, we have no $\mathcal{L}_{phase}$ and less penalty for the statistical differences, and its absence causes the high MSE, as we can see in Fig. 11. γ also affects the MSE, but $\mathcal{L}_{phase}$ reduces $\mathcal{L}_{amp}$'s side effect.

Table 7: Ablation study on with ETTh2, $L = 720$, and N-Beats

Metric	Default	$\gamma = 0.1$	$\gamma = 0.5$	$\gamma = 1.0$	$\alpha = 0.0$	$\alpha = 0.1$	$\alpha = 0.8$	$\alpha = 1.0$	$\mathcal{L}_{a.shift}$ only	$\mathcal{L}_{phase}$ only	$\mathcal{L}_{amp}$ only
MSE	0.3005	0.2968	0.3083	0.3168	0.3075	0.3161	**0.2872**	1.1752	1.5123	0.3391	1.8453
DTW	**17.5154**	17.5265	17.5649	17.7302	17.7564	17.5931	17.6508	17.6886	17.7261	17.848	18.0261
TDI	9.2197	9.1303	9.2261	9.4366	10.3550	9.8957	**8.4725**	8.7118	10.2519	12.8568	10.5602
LCSS	0.5382	0.5366	0.5277	0.5137	0.5050	0.5137	**0.5584**	0.5445	0.5341	0.4920	0.5086

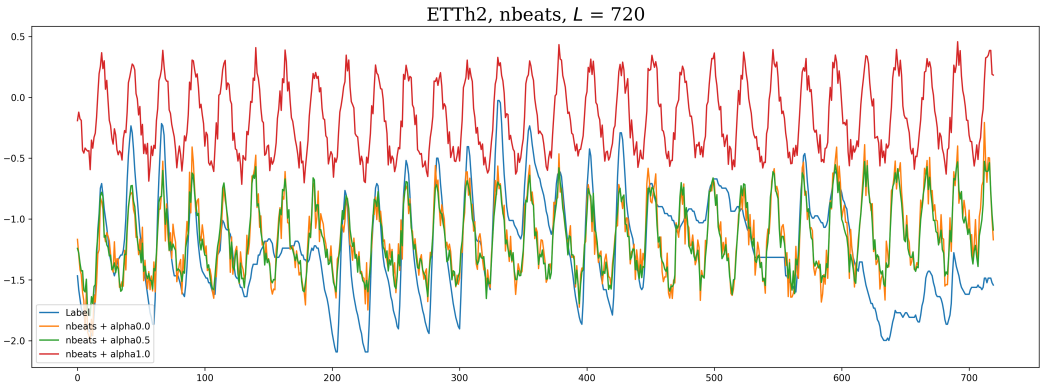

Figure 11: Ablation study result visualization with different α on ETTh2 dataset

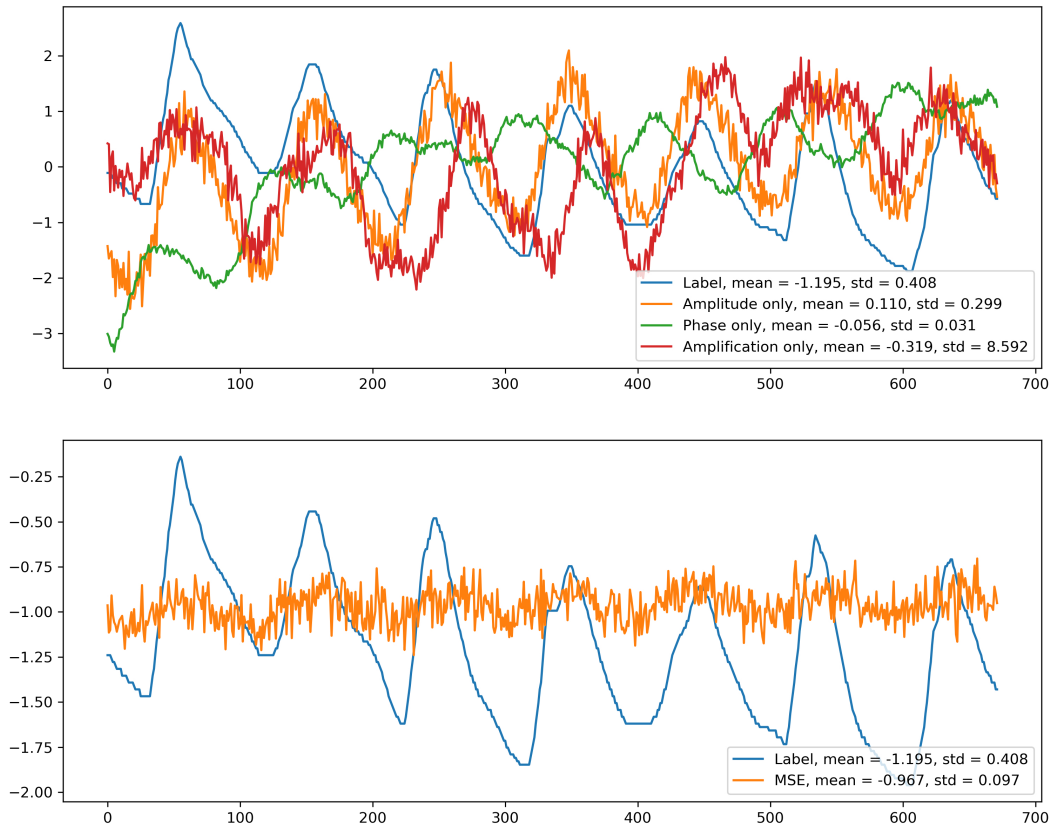

Figure 12: Ablation study result visualization of three proposed loss function on ETTm2 dataset