HAL
open science

Deep Time Series Forecasting with Shape and Temporal Criteria

Vincent Le Guen, Nicolas Thome

To cite this version:

Vincent Le Guen, Nicolas Thome. Deep Time Series Forecasting with Shape and Temporal Criteria. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 45 (1), pp.342-355. 10.1109/TPAMI.2022.3152862 . hal-03588390

HAL Id: hal-03588390

https://hal.science/hal-03588390v1

Submitted on 24 Feb 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Deep Time Series Forecasting with Shape and Temporal Criteria

Vincent Le Guen, Nicolas Thome

Abstract—This paper addresses the problem of multi-step time series forecasting for non-stationary signals that can present sudden changes. Current state-of-the-art deep learning forecasting methods, often trained with variants of the MSE, lack the ability to provide sharp predictions in deterministic and probabilistic contexts. To handle these challenges, we propose to incorporate shape and temporal criteria in the training objective of deep models. We define shape and temporal similarities and dissimilarities, based on a smooth relaxation of Dynamic Time Warping (DTW) and Temporal Distortion Index (TDI), that enable to build differentiable loss functions and positive semi-definite (PSD) kernels. With these tools, we introduce DILATE (DIstortion Loss including shApe and TimE), a new objective for deterministic forecasting, that explicitly incorporates two terms supporting precise shape and temporal change detection. For probabilistic forecasting, we introduce STRIPE++ (Shape and Time diverRsIty in Probabilistic forEcasting), a framework for providing a set of sharp and diverse forecasts, where the structured shape and time diversity is enforced with a determinantal point process (DPP) diversity loss. Extensive experiments and ablations studies on synthetic and real-world datasets confirm the benefits of leveraging shape and time features in time series forecasting.

Index Terms—Time series forecasting, deep neural networks, differentiable programming, loss functions, structured prediction, shape and temporal criteria, dynamic time warping, time distortion, structured diversity, determinantal point processes

1 Introduction

Time series forecasting consists in analyzing historical signal correlations to anticipate future behaviour. Traditional approaches include linear autoregressive methods [3] or state space models [4], which are simple yet mathematically grounded and benefit from interpretability. They often exploit prior knowledge based on stationarity, e.g. by leveraging trend or seasonality to constrain forecasting.

These grounding assumptions are often violated in many real-world time series that are non-stationary and can present sharp variations such as sudden drops or changes of regime. Long-term multi-step forecasting in this context is particularly challenging and arises in a wide range of important application fields, e.g. analyzing traffic flows [5], [6], medical records [7], predicting sharp variations in financial markets [8] or in renewable energy production [9], [10], [11], *etc*.

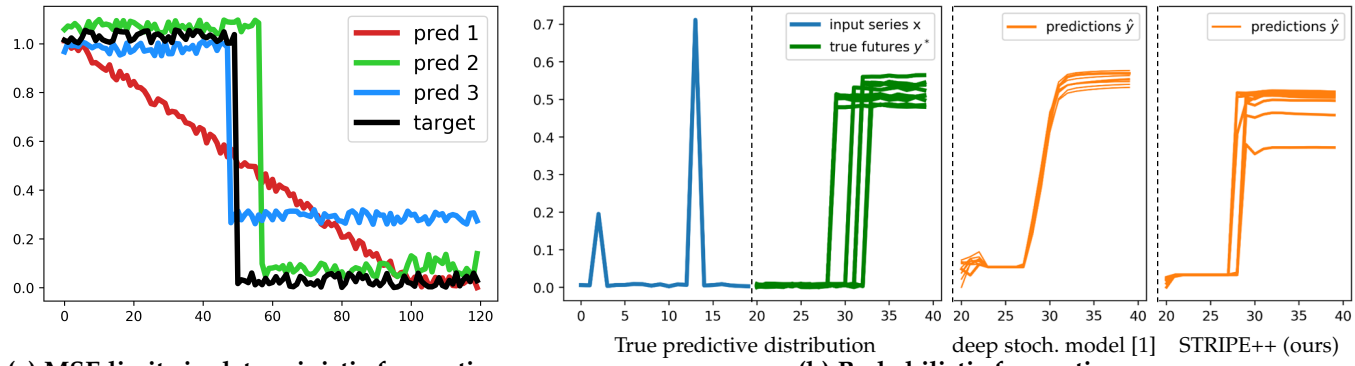
This work focuses on forecasting multi-step future trajectories with potentially sharp variations in the deterministic and probabilistic cases. Deep neural networks are an appealing solution for this problem [12], [13], [14], [15], [16], [17], due to their automatic feature extraction and complex nonlinear time dependencies modeling. However, the verification criteria typically used in applications are not used at training time because they are mostly not differentiable. We may cite for instance the ramp score [9] for assessing the detection of sharp ramping events, or the Time Distortion Index (TDI) [18] for assessing the time delay of a particular predicted event.

Instead, the huge majority of methods optimize at training time the Mean Squared Error (MSE) or its variants (MAE, quantile loss, *etc*) as a proxy loss function. However, the MSE has important drawbacks for assessing predictions, especially in non-stationary contexts with sharp variations [9], [18], [19], [20], [21], [22], [23]. We illustrate the limitations of the MSE in Figure 1. Figure 1 (a) shows three deterministic predictions, which have the same MSE loss compared to the target step function (in black). Thus, the MSE does not support predictions (2) and (3) over prediction (1), although they clearly are more adequate for regulation purposes because they do anticipate the drop to come, although with a slight delay (2) or with a slightly inaccurate amplitude (3). For probabilistic forecasting (Figure 1 (b)), current state-of-the art probabilistic methods trained with variants of the MSE tend to produce blurry predictions that do not match the sharp steps of the true futures (in green).

In this work, we intend to bridge this train/test criterion gap by introducing shape and temporal similarities and dissimilarities for training deep forecasting models. We introduce differentiable loss functions, which makes them amenable to gradient-based optimization. We decouple the definition of these criteria from the introduction of general deterministic and probabilistic forecasting methods leveraging such criteria.

For deterministic forecasting, we introduce the DILATE training loss function (DIstortion Loss with shApe and TimE) as an alternative to the MSE. DILATE efficiently combines a shape matching term based on a smooth approximation of Dynamic Time Warping (DTW) [19] with a temporal term, which ensures both sharp predictions with accurate temporal localization. In Figure 1 (a), DILATE favors predictions (2) and (3) over prediction (1), contrarily to the MSE.

For probabilistic forecasting, our goal is to cover the true future distribution density with a small number (e.g. $N = 10$) of admissible trajectories. To this end, we introduce the STRIPE++ framework (Shape and Time diverRsIty in Probabilistic forEcasting) for producing both diverse and accurate forecasts. We leverage the proposed shape and temporal positive semi-definite (PSD) kernels to design a structured diversity measure, which is embedded into



(a) MSE limits in deterministic forecasting **(b) Probabilistic forecasting**

Fig. 1: (a) We illustrate the limitations of the MSE in deterministic forecasting: the three predictions (1,2,3) have the same MSE with respect to the target, contrary to our proposed DILATE loss that favours predictions 2 (correct shape, slight delay) and 3 (correct timing, inaccurate amplitude) over prediction 1. (b) In probabilistic forecasting, state-of-the-art methods trained with variants of the MSE (e.g. [1], [2]) loose the ability to produce sharp forecasts (middle). In contrast, our proposed STRIPE++ model ensures both sharp and diverse forecasts (right) that better match the true future distribution (left).

a determinantal point processes (DPP). As illustrated in Figure 1 (b), STRIPE++ produces diverse predictions, which are i) representative of the ground truth future density and ii) consistent to the sharp nature of the expected forecasts. We validate the relevance of DILATE and STRIPE++ on extensive synthetic and real-world datasets highlighting the benefits brought up by leveraging shape and temporal features for time series forecasting.

This paper extends two previous conference papers that first introduced the DILATE loss [24] and the STRIPE model [25]. This submission includes the following improvements:

- We provide a unified view of shape and temporal differentiable criteria between time series, expressed in terms of dissimilarities (loss functions) and similarities (positive semi-definite kernels). This general view enables to put into perspective our contributions for deterministic and probabilistic forecasting with respect to the literature. In that regard, the related work has substantially been extended compared to [24], [25].
- STRIPE++ improves STRIPE [25] at two important methodological levels for guaranteeing a good quality/diversity cooperation: we enrich the prediction model with a posterior network inspired by [26] for better disentangling deterministic and probabilistic spaces, and we add a quality constraint into the DPP kernels to explicitly support predictions, which are both diverse and accurate.
- The experiments have been substantially extended, including new evaluations of DILATE on top of recent state-of-the-art forecasting models [16], [17] , experiments on the additional Electricity and ETTh1 datasets, a more in-depth analysis of the DTW smoothing parameter γ and a comparison to a recent smooth DTW variant [27], and a validation of the relevance of STRIPE++ compared to STRIPE [25] with respect to the F1 score.

2 Related work

We review here the literature on deterministic and probabilistic time series forecasting, focusing on the multi-step non stationary context. We insist on the previous works leveraging shape and temporal features and the methods for enforcing structured diversity.

2.1 Time series forecasting

Deterministic forecasting: Traditional methods for time series forecasting are based on linear state space models [4], including autoregressive models (e.g. ARIMA [3]) and Exponential Smoothing [28]. These methods often exploit strong structural assumptions on data such as stationarity and seasonality that are not satisfied for many real-world time series that can present abrupt changes of distribution. Since, Recurrent Neural Networks (RNNs), in particular Long Short Term Memory Networks (LSTMs) [29], have become popular due to their automatic feature extraction capabilities and long-term dependencies modeling. For multi-step forecasting, the most common approach is to apply recursively a one-step ahead trained model [14]. A thorough comparison of the different multi-step strategies [30] recommend the Direct Multi-Horizon Strategy. Of particular interest in this category are Sequence To Sequence (Seq2Seq) models [13], [31], [32], [33]. Recently, much effort has been devoted to design new architectures that address error accumulation in multi-step forecasting [34], [35]. In particular, Oreshkin *et al.* introduce the N-Beats model [16] with deep stacks of fully-connected layers with forward and backward residual connections. Zhou *et al.* propose the Informer model [17] that leverages the self-attention Transformer architecture [36] to extend predictions to much farther temporal horizons.

Probabilistic forecasting: Many critical applications require forecasts associated with uncertainty to make relevant decisions. Some methods propose to estimate the variance with Monte Carlo dropout [37], [38] or ensembling [39]. Other methods predict the quantiles of the conditional distribution of future values [40], [41], [42] by minimizing the pinball loss [43] or the continuous ranked probability score (CRPS) [44]. State space models, which have an inherent probabilistic nature, were recently revisited with deep learning [45], [46]. Salinas *et al.* introduce deepAR [15] by explicitly approximat-

ing the predictive distribution of the next time step with a Gaussian distribution; subsequent works investigate other parametric distributions [47]. Generative models were also used to implicitly approximate the predictive distribution, e.g. with conditional variational autoencoders (cVAEs) [1], conditional generative adversarial networks (cGANs) [48], or normalizing flows [2], [49]. However, these methods lack the ability to produce sharp forecasts by minimizing variants of the MSE (pinball loss, Gaussian maximum likelihood), at the exception of cGANs - but which suffer from mode collapse that limits predictive diversity. Moreover, these generative models are generally represented by unstructured distributions in the latent space (e.g. Gaussian), which do not allow to have an explicit control on the targeted diversity.

2.2 Evaluation and training metrics

Current research mainly focuses on forecasting architecture design and the question of the training loss is often overlooked. The MSE, MAE and its variants (SMAPE, *etc*) are predominantly used as proxies. Many works have highlighted the limitations of the MSE to assess the ability to anticipate sharp variations [9], [18], [19], [20], [21], [22], [23]. Metrics reflecting shape and temporal localization exist but their non-differentiability makes them unsuitable for training deep models. For characterizing shape, Dynamic Time Warping (DTW) [50], [51], [52] performs time series alignment, and the ramp score [9], [20] assesses the detection of ramping events in wind and solar energy forecasting. Timing errors can be characterized among other ways by the Temporal Distortion Index (TDI) [9], [18], or by computing detection scores (precision, recall, Hausdorff distance) after change point detection [53].

Recently, some attempts have been made to train deep neural networks based on alternatives to MSE, especially based on smooth approximations of DTW [19], [27], [54], [55], [56]. The soft-DTW of Cuturi and Blondel [19] was further normalized to ensure a non-negative divergence [27]. However, since DTW is by design invariant to elastic distortions, it completely ignores the temporal localization of the changes (see illustrations of soft-DTW in Figure 4). A differentiable timing error loss function based on DTW on the event (binary) space was proposed in [57] ; however it is only applicable for predicting binary time series. Some works explored the use of adversarial losses for time series [58], [59], which can be seen as an implicit way of enforcing semantic criteria learned from data. However, it gives a weaker and non-interpretable control on shape and time criteria and brings additional adversarial training difficulties. In this work, we focus on efficiently combining explicit shape and temporal differentiable criteria at training time.

2.3 Structured diversity for prediction

For diversifying forecasts, several repulsive schemes were studied such as the variety loss [60], [61] that consists in optimizing the best sample, or entropy regularization [62], [63] that encourages a uniform distribution. Besides, generative models, such as variational autoencoders (VAE) [64], are widely used for producing multiple predictions through sampling from a latent space. However latent states are typically sampled at test time from a standard Gaussian prior distribution, resulting in an unstructured diversity. To improve this unstructured mechanism, prior works [1], [65] introduced proposal neural networks for generating the latent variables that are trained with a diversity objective.

Determinantal Point Processes (DPP): DPPs are appealing probabilistic models for describing the diversity of a set of items $\mathcal{Y} = \{\mathbf{y}_1, ..., \mathbf{y}_N\}$. A DPP is a probability distribution over all subsets of $\mathcal{Y}$ that assigns the following probability to a random subset $\mathbf{Y}$:

$$\mathcal{P}_{\mathbf{K}}(\mathbf{Y} = Y) = \frac{\det(\mathbf{K}_Y)}{\sum_{Y' \subseteq \mathcal{Y}} \det(\mathbf{K}'_Y)} = \frac{\det(\mathbf{K}_Y)}{\det(\mathbf{K} + \mathbf{I})} \quad (1)$$

where $\mathbf{K}$ is a positive semi-definite (PSD) kernel and $\mathbf{K}_A$ is its restriction to the elements indexed by A.

DPPs offer efficient algorithms for sequentially sampling diverse items or maximizing the diversity of a set with a given sampling budget. Importantly, the choice of the kernel enables to incorporate prior structural knowledge on the targeted diversity. As such, DPPs have been successfully applied in various contexts, e.g. document summarization [66], recommendation systems [67], image generation [68] and diverse trajectory forecasting [1]. GDPP [68] proposed by Elfeki *et al.* is based on matching generated and true sample diversity by aligning the corresponding DPP kernels, and thus limits their use in datasets where the full distribution of possible outcomes is accessible. In contrast, our probabilistic forecasting approach is applicable in realistic scenarii where only a single future trajectory is available for each training sample. Although we share with the work of Yuan and Kitani [1] the goal to use DPP as diversification mechanism for future trajectories, the main limitation in [1] is to use the MSE loss for training the predictor and the MSE kernel for diversification, leading to blurred prediction, as illustrated in Figure 1 (b). In contrast, we design specific shape and time DPP kernels and we show the necessity to decouple the criteria used for quality and diversity.

3 Differentiable shape and temporal (dis)similarities for time series

In this section, we introduce an unified view of shape and temporal criteria applicable to deep time series forecasting. We insist on their differentiability and efficient computation. We propose both a dissimilarity version useful for defining loss functions and a similarity version useful for defining positive semi-definite (PSD) kernels.

3.1 Shape (dis)similarity

3.1.1 Background: Dynamic Time Warping

To assess the shape similarity between two time series, the popular Dynamic Time Warping (DTW) method [50] seeks a minimal cost alignment for handling time distortions. Given two d-dimensional time series $\mathbf{y} \in \mathbb{R}^{d \times n}$ and $\mathbf{z} \in \mathbb{R}^{d \times m}$ of lengths n and m, DTW looks for an optimal warping path represented by a binary matrix $\mathbf{A} \subset \{0,1\}^{n \times m}$ where $\mathbf{A}_{ij} = 1$ if $\mathbf{y}_i$ is associated to $\mathbf{z}_j$ and 0 otherwise. The set of admissible warping paths $\mathcal{A}_{n,m}$ is composed of paths connecting the endpoints $(1,1)$ to (n,m) with the following authorized moves $\rightarrow, \downarrow, \searrow$. The cost of warping path $\mathbf{A}$ is the

sum of the costs along the alignment ; this cost can be written as the scalar product $\langle \mathbf{A}, \mathbf{\Delta}(\mathbf{y}, \mathbf{z}) \rangle$, where $\Delta(\mathbf{y}, \mathbf{z})$ is a $n \times m$ pairwise dissimilarity matrix whose general term is typically chosen an the euclidean distance $\mathbf{\Delta}(\mathbf{y}, \mathbf{z})_{ij} = \|\mathbf{y}_i - \mathbf{z}_j\|_2^2$. DTW computes the minimal cost warping path:

$$\mathrm{DTW}^{\mathbf{\Delta}}(\mathbf{y}, \mathbf{z}) := \min_{\mathbf{A} \in \mathcal{A}_{n,m}} \langle \mathbf{A}, \mathbf{\Delta}(\mathbf{y}, \mathbf{z}) \rangle \tag{2}$$

Although the cardinality of $\mathcal{A}_{n,m}$ increases exponentially in $\min(n, m)$ [1], DTW and the optimal path $\mathbf{A}^*$ can be computed efficiently in $\mathcal{O}(nm)$ by dynamic programming. However, a major limitation of DTW is its non-diffentiability, which prevents its integration in neural network pipelines trained with gradient-based optimization.

3.1.2 Smooth DTW shape dissimilarity

For handling the non-differentiability of DTW, Cuturi and Blondel [19] introduced the soft-DTW by replacing the hard-minimum operator by a smooth minimum with the log-sum-exp trick $\min_\gamma(a_1, ..., a_n) = -\gamma \log(\sum_i^n \exp(-a_i/\gamma))$:

$$\mathrm{DTW}_\gamma^{\mathbf{\Delta}}(\mathbf{y}, \mathbf{z}) \quad := \quad -\gamma \log \left(\sum_{\mathbf{A} \in \mathcal{A}_{n,m}} e^{-\langle \mathbf{A}, \mathbf{\Delta}(\mathbf{y}, \mathbf{z}) \rangle / \gamma} \right) \tag{3}$$

where $\gamma > 0$ is a smoothing parameter (when $\gamma \to 0$, this converges to the true DTW).

$\mathrm{DTW}_\gamma^{\mathbf{\Delta}}$ as defined in Eq. 3 is differentiable with respect to $\mathbf{\Delta}$ (and with respect to both series $\mathbf{y}$ and $\mathbf{z}$ by chain's rule, provided a differentiable cost function $\mathbf{\Delta}$).

We can interpret this relaxed DTW version by considering, instead of the unique optimal path $\mathbf{A}^*$, a Gibbs distribution over possible paths:

$$p_\gamma(\mathbf{A}; \mathbf{\Delta}) = \frac{1}{Z} e^{-\langle \mathbf{A}, \mathbf{\Delta}(\mathbf{y}, \mathbf{z}) \rangle / \gamma} \tag{4}$$

The soft-DTW is then the negative log-partition of this distribution: $\mathrm{DTW}_\gamma^{\mathbf{\Delta}}(\mathbf{y}, \mathbf{z}) := -\gamma \log Z$.

Since $\mathrm{DTW}_\gamma^{\mathbf{\Delta}}(\mathbf{y}, \mathbf{z})$ can take negative values and is not minimized for $\mathbf{y} = \mathbf{z}$, Mensch and Blondel [54] normalized the soft-DTW to make it a true divergence. We found experimentally that this does not improve performances and is heavier computationally (see Appendix D.4).

3.1.3 Shape similarity kernel

Based on the soft-DTW shape dissimilarity defined in Eq. 3, we define a shape similarity kernel as follows:

$$\mathcal{K}_{shape}(\mathbf{y}, \mathbf{z}) = e^{-\mathrm{DTW}_\gamma^{\mathbf{\Delta}}(\mathbf{y}, \mathbf{z})/\gamma}. \tag{5}$$

We experiment with the following choices of kernels $\Delta_{ij} = \Delta(\mathbf{y}, \mathbf{z})_{ij}$:

- Half-Gaussian: $\mathbf{\Delta}_{ij} = \|\mathbf{y}_i - \mathbf{z}_j\|_2^2 + \log(2 - e^{-\|\mathbf{y}_i - \mathbf{z}_j\|_2^2})$
- L1: $\mathbf{\Delta}_{ij} = \|\mathbf{y}_i - \mathbf{z}_j\|_1 \quad$ (for $d = 1$)
- Euclidean: $\mathbf{\Delta}_{ij} = \|\mathbf{y}_i - \mathbf{z}_j\|_2^2$

$\mathcal{K}_{shape}$ was proven to be positive semi-definite (PSD) for the half-Gaussian[2] and the L1 kernels [27], [69] and is conjectured to be PSD for the Euclidean kernel [27]. Experimentally we observed that these three cost matrices lead to similar behaviour.

1. $|\mathcal{A}_{n,m}|$ is equal to the Delannoy number $Delannoy(n, m)$ which grows exponentially in $\min(n, m)$

2. We denote this kernel "half-Gaussian" since the corresponding k kernel defined in the proof (Appendix A) equals $k(\mathbf{y}_i, \mathbf{z}_j) = e^{-\Delta(\mathbf{y}_i, \mathbf{z}_j)} = \left(\frac{1}{2} e^{-\|\mathbf{y}_i - \mathbf{z}_j\|^2}\right) \times \left(1 - \frac{1}{2} e^{-\|\mathbf{y}_i - \mathbf{z}_j\|^2}\right)^{-1}$

3.2 Temporal (dis)similarity

Quantifying the temporal similarity between two time series consists in analyzing the time delays between matched patterns detected in both series. As discussed in introduction, it of great importance for many applications to anticipate sharp variations.

3.2.1 Smooth temporal distortion index

A common temporal similarity is the Temporal Distortion Index (TDI) [9], [18]. The TDI computes the approximate area included between the optimal path $\mathbf{A}^*$ and the first diagonal, characterizing the presence of temporal distortion. A generalized version of the TDI, that we proposed in [24], can be written:

$$\mathrm{TDI}^{\mathbf{\Delta}, \mathbf{\Omega}_{\mathbf{dissim}}}(\mathbf{y}, \mathbf{z}) := \langle \mathbf{A}^*, \mathbf{\Omega}_{\mathbf{dissim}} \rangle \tag{6}$$

where $\mathbf{A}^* = \arg\min_{\mathbf{A} \in \mathcal{A}_{n,m}} \langle \mathbf{A}, \mathbf{\Delta}(\mathbf{y}, \mathbf{z}) \rangle$ is the DTW optimal path and $\mathbf{\Omega}_{\mathbf{dissim}} \in \mathbb{R}^{n \times m}$ is a matrix penalizing the association between $\mathbf{y}_i$ and $\mathbf{z}_j$ for $i \neq j$. We typically choose a quadratic penalization $\mathbf{\Omega}_{\mathbf{dissim}}(i, j) \propto (i - j)^2$, but other variants can encode prior knowledge and penalize more heavily late than early predictions, and *vice-versa*.

The TDI dissimilarity defined in Eq. 6 is however non-differentiable, since the optimal path $\mathbf{A}^*$ is not differentiable with respect to $\mathbf{\Delta}$. We handle this problem by defining a relaxed optimal path $\mathbf{A}_\gamma^*$ as the gradient of $\mathrm{DTW}_\gamma^{\mathbf{\Delta}}$:

$$\mathbf{A}_\gamma^* := \nabla_{\mathbf{\Delta}} \mathrm{DTW}_\gamma^{\mathbf{\Delta}}(\mathbf{y}, \mathbf{z}) = \frac{1}{Z} \sum_{\mathbf{A} \in \mathcal{A}_{n,m}} \mathbf{A}\, e^{-\langle \mathbf{A}, \mathbf{\Delta}(\mathbf{y}, \mathbf{z}) \rangle / \gamma} \tag{7}$$

The expression in Eq. 7 results from a direct computation from Eq. 3. Notice that this soft optimal path corresponds to the expected path $\mathbf{A}_\gamma^* = \mathbb{E}_{p_\gamma(\cdot; \mathbf{\Delta})}[\mathbf{A}]$ under the Gibbs distribution in Eq. 4. Note also that $\mathbf{A}_\gamma^*$ becomes a soft assignment, i.e. $\mathbf{A}_\gamma^*(i, j)$ represents the probability for a path to contain the cell (i, j). An illustration of soft optimal paths with the influence of γ is given in Figure 6.

We can now define a differentiable version of the TDI:

$$\mathrm{TDI}_\gamma^{\mathbf{\Delta}, \mathbf{\Omega}_{\mathbf{dissim}}}(\mathbf{y}, \mathbf{z}) := \left\langle \mathbf{A}_\gamma^*, \mathbf{\Omega}_{\mathbf{dissim}} \right\rangle = \frac{1}{Z} \sum_{\mathbf{A} \in \mathcal{A}_{n,m}} \langle \mathbf{A}, \mathbf{\Omega}_{\mathbf{dissim}} \rangle\, e^{-\frac{\langle \mathbf{A}, \mathbf{\Delta}(\mathbf{y}, \mathbf{z}) \rangle}{\gamma}} \tag{8}$$

which corresponds to the expected value of the TDI under the Gibbs distribution.

3.2.2 Temporal similarity kernel

Based on the temporal dissimilarity in Eq. 8 and the shape similarity kernel in Eq. 5, we can define a time similarity as follows:

$$\mathcal{K}_{time}(\mathbf{y}, \mathbf{z}) := e^{-\mathrm{DTW}_\gamma^{\mathbf{\Delta}}(\mathbf{y}, \mathbf{z})/\gamma} \times \mathrm{TDI}_\gamma^{\mathbf{\Delta}, \mathbf{\Omega}_{\mathbf{sim}}}(\mathbf{y}, \mathbf{z}) \tag{9}$$

where in this case, we use a similarity matrix $\mathbf{\Omega}_{\mathbf{sim}}$ favoring pairs of time series with low temporal distortion, i.e. with an optimal path near the main diagonal. We typically choose a pointwise inverse of $\mathbf{\Omega}_{\mathbf{dissim}}$: $\mathbf{\Omega}_{\mathbf{sim}}(i, j) = \frac{1}{(i-j)^2 + 1}$. We

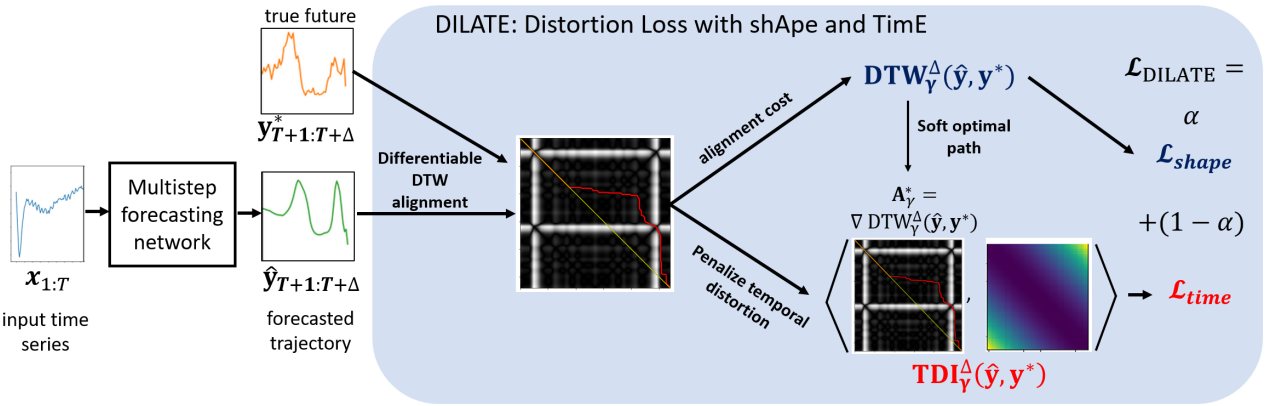


Fig. 2: The DILATE loss $\mathcal{L}_{\text{DILATE}}$ for training deterministic deep time series forecasting models is composed of two terms: $\mathcal{L}_{shape}$ based on the soft DTW and $\mathcal{L}_{time}$ that penalizes the temporal distortions visible on the soft optimal path. The overall loss $\mathcal{L}_{\text{DILATE}}$ is differentiable, and we provide an efficient implementation of its forward and backward passes.

prove that $\mathcal{K}_{time}$ defines a valid PSD temporal kernel (proof in Appendix A).

The table below provides an overview of the shape and temporal criteria introduced in this work:

criterion	differentiable loss	PSD sim. kernel
shape	$\mathrm{DTW}^{\mathbf{\Delta}}_{\gamma}(\mathbf{y}, \mathbf{z})$	$e^{-\mathrm{DTW}^{\mathbf{\Delta}}_{\gamma}(\mathbf{y},\mathbf{z})/\gamma}$
time	$\mathrm{TDI}^{\mathbf{\Delta},\mathbf{\Omega}_{\mathbf{dissim}}}_{\gamma}(\mathbf{y}, \mathbf{z})$	$e^{-\mathrm{DTW}^{\mathbf{\Delta}}_{\gamma}(\mathbf{y},\mathbf{z})/\gamma}$ $\times \mathrm{TDI}^{\mathbf{\Delta},\mathbf{\Omega}_{\mathbf{sim}}}_{\gamma}(\mathbf{y}, \mathbf{z})$

Efficient forward and backward computation: The direct computation of the shape and temporal dissimilarities in Eq. 3 and Eq. 8 is intractable, due to the exponential growth of cardinal of $\mathcal{A}_{n,m}$. We provide a careful implementation of the forward and backward passes in order to make learning efficient, limiting the forward and backward complexity to $\mathcal{O}(nm)$ (see details in Appendix B).

3.2.3 Discussion on differentiability

One may mask why not directly optimizing the true DTW and TDI objectives with subgradient methods instead of deriving smooth surrogates? The DTW is indeed continuous with respect to both input time series and when the optimal warping path $\mathbf{A}^*$ is unique (which is almost always the case for continuous data), DTW is differentiable almost everywhere with gradient $\mathbf{A}^*$. However, a first difficulty of directly optimizing DTW is that the gradient is discontinuous at the points where a change in the input time series causes a change in the optimal path $\mathbf{A}^*$, which can hamper the performances of gradient-based methods. A second difficulty is that the DTW objective is non-convex, making the subgradient method prone to collapsing to local minima [19]. Furthermore, for defining the TDI in the DILATE loss, we need second-order differentiability, which exacerbates the aforementioned difficulties.

Considering a smooth relaxation of DTW and TDI alleviates both difficulties. First, these relaxations are continuously differentiable by design. Second, the smoothing convexifies the DTW and TDI objectives and provide a better optimization landscape. This point was noted by several works [19], [54] which showed that the soft-DTW yielded better performances than the DTW optimized with subgradient, even when evaluated with the true DTW. We have also confirmed this experimentally for our smooth shape and time surrogate criteria in section 5.2.3.

4 Application to Time Series Forecasting

Given an input sequence $\mathbf{x}_{1:T} = (\mathbf{x}_1, \ldots, \mathbf{x}_T) \in \mathbb{R}^{p \times T}$, we consider as discussed in introduction the multi-step time series forecasting problem in two important contexts: **(1) deterministic forecasting** consisting in predicting a τ-steps future trajectory $\hat{\mathbf{y}} = (\hat{\mathbf{y}}_{T+1}, \ldots, \hat{\mathbf{y}}_{T+\tau}) \in \mathbb{R}^{d \times \tau}$ and **(2) probabilistic forecasting** where we seek to predict a set of N future trajectories $\{\hat{\mathbf{y}}^{(i)}\}_{i=1..N} \in \mathbb{R}^{d \times \tau}$ (corresponding to diverse scenarii sampled from the true future distribution $\hat{\mathbf{y}}^{(i)} \sim p(\cdot|\mathbf{x}_{1:T})$).

In this section, we introduce two general methods to leverage differentiable shape and temporal criteria - such as those derived in section 3 - for deterministic and probabilistic forecasting. We introduce in section 4.1 a differentiable loss function, termed DILATE, for tackling the deterministic forecasting problem (1) in non-stationary contexts, that ensures sharp predictions with accurate temporal localization. In section 4.2, we extend these ideas to the probabilistic problem (2) with the STRIPE++ model that enforces structured shape and temporal diversity with determinantal point processes (DPP).

4.1 DILATE loss for deterministic forecasting

Our proposed framework for deterministic forecasting is depicted in Figure 2. We introduce the DIstortion Loss with shApe and TimE (DILATE) for training any deterministic deep multi-step forecasting model. Crucially, the DILATE loss needs to be differentiable in order to train models with gradient-based optimization.

The DILATE objective function, which compares the prediction $\hat{\mathbf{y}} = (\hat{\mathbf{y}}_{T+1}, \ldots, \hat{\mathbf{y}}_{T+\tau})$ with the actual ground truth future trajectory $\mathbf{y}^* = (\mathbf{y}^*_{T+1}, \ldots, \mathbf{y}^*_{T+\tau})$, is composed of two terms balanced by the hyperparameter $\alpha \in [0, 1]$:

$$\begin{aligned}\mathcal{L}_{\text{DILATE}}(\hat{\mathbf{y}}, \mathbf{y}^*) &= \alpha \, \mathcal{L}_{shape}(\hat{\mathbf{y}}, \mathbf{y}^*) + (1-\alpha) \, \mathcal{L}_{time}(\hat{\mathbf{y}}, \mathbf{y}^*) \\ &= \alpha \, \mathrm{DTW}^{\mathbf{\Delta}}_{\gamma}(\hat{\mathbf{y}}, \mathbf{y}^*) + (1-\alpha) \, \mathrm{TDI}^{\mathbf{\Delta},\mathbf{\Omega}_{\mathbf{dissim}}}_{\gamma}(\hat{\mathbf{y}}, \mathbf{y}^*) \quad (10)\end{aligned}$$

Discussion: A variant of our approach to combine shape and temporal penalization would be to incorporate a temporal term inside our smooth $\mathcal{L}_{shape}$ function in Eq. 3, i.e. :

$$\mathcal{L}_{\text{DILATE}^t}(\hat{\mathbf{y}}_i, \mathbf{y}_i^*) := -\gamma \log \left(\sum_{\mathbf{A} \in \mathcal{A}_{n,m}} \exp\left(-\frac{\langle \mathbf{A}, \alpha \mathbf{\Delta}(\hat{\mathbf{y}}_i, \mathbf{y}_i^*) + (1-\alpha)\mathbf{\Omega} \rangle}{\gamma} \right) \right) \quad (11)$$

We can notice that Eq. 11 reduces to minimizing $\langle \mathbf{A}, \alpha \mathbf{\Delta}(\hat{\mathbf{y}}_i, \mathbf{y}_i^*) + (1-\alpha)\mathbf{\Omega} \rangle$ when $\gamma \rightarrow 0^+$. In this case, $\mathcal{L}_{\text{DILATE}^t}$ can recover DTW variants studied in the literature to bias the computation based on penalizing sequence misalignment, by designing specific $\mathbf{\Omega}$ matrices:

Sakoe-Chiba DTW band constraint [50]	$\Omega(i,j) = \begin{cases} +\infty \text{ if } \lvert i-j \rvert > T \\ 0 \quad \text{otherwise} \end{cases}$
Weighted DTW [51]	$\Omega(i,j) = f(\lvert i-j \rvert) \quad f \nearrow$

$\mathcal{L}_{\text{DILATE}^t}$ in Eq. 11 enables to train deep neural networks with a smooth loss combining shape and temporal criteria. However, $\mathcal{L}_{\text{DILATE}^t}$ presents limited capacities for disentangling the shape and temporal errors, since the optimal path is computed from both shape and temporal terms. In contrast, our $\mathcal{L}_{\text{DILATE}}$ loss in Eq. 10 separates the loss into two shape and temporal components, the temporal penalization being applied to the optimal unconstrained DTW path.

4.2 Probabilistic forecasting with structured diversity

We introduce the STRIPE++ framework (Shape and Time diverRsIty in Probabilistic forEcasting), that extends STRIPE [25]. Depicted in Figure 3, STRIPE++ builds upon a general multi-step forecasting pipeline: the input time series $\mathbf{x}_{1:T}$ is fed into an encoder that summarizes the input into a latent vector h. This context vector h is then transformed by a decoder into a future trajectory.

The key idea of STRIPE++ is to augment the deterministic latent state h with stochastic diversifying variables z_s (resp. z_t) meant to capture the shape (resp. temporal) variations of the future time series. We distinguish two phases for training the overall model: (i) we train the predictor with a quality loss and (ii) we train the diversifying STRIPE++ mechanism with a DPP diversity loss (with the weights of the predictor frozen). For both of these steps, we detail now how the diversifying variables are sampled.

4.2.1 Training the predictor with a quality loss

For training the predictor (upper part in Figure 3) with possibly multiple admissible futures as supervision, we take inspiration from the probabilistic U-Net [26] and introduce a posterior network from which to sample the diversifying variables z_s^* and z_t^* (which represent the shape and temporal variant attached to a particular future $\mathbf{y}^*$). The posterior net outputs the parameters μ_s^* and σ_s^* of a Gaussian distribution $\mathcal{N}(\mu_s^*, \sigma_s^*)$ for parameterizing the shape posterior distribution $q(z_s|\mathbf{x}, \mathbf{y}^*)$ (and similarly for the temporal posterior distribution).

To train this generative model (encoder, decoder and posterior networks), we resort to variational inference [64] and maximize the evidence lower bound (ELBO) of the log-likelihood, or equivalently, minimize the following prediction loss over all training examples:

$$\mathcal{L}_{prediction}(\hat{\mathbf{y}}, \mathbf{y}^*) = \mathcal{L}_{quality}(\hat{\mathbf{y}}, \mathbf{y}^*) + \text{KL}\left(q(z_s|\mathbf{x}, \mathbf{y}^*) \,||\, p(z_s)\right) + \text{KL}\left(q(z_t|\mathbf{x}, \mathbf{y}^*) \,||\, p(z_t)\right) \quad (12)$$

In our non-stationary context, we choose the DILATE loss for $\mathcal{L}_{quality}$, in order to guarantee sharp predictions with accurate temporal localization. The Kullback-Leibler (KL) losses enforce that the shape posterior distribution $q(z_s|\mathbf{x}, \mathbf{y}^*)$ matches a prior distribution $p(z_s)$ (we use a Gaussian prior $\mathcal{N}(0, \mathbf{I})$ - a common choice in variational inference).

4.2.2 Training the STRIPE++ diversification mechanism

For including structured shape and temporal diversity (lower part in Figure 3), we introduce two proposal neural networks $\text{STRIPE}^{++}_{\text{shape}}$ and $\text{STRIPE}^{++}_{\text{time}}$ that aim to produce a set of N_s shape latent codes $\{z_s^i\}_{i=1..N_s} \in \mathbb{R}^k$ (resp. N_t time codes $\{z_t^i\}_{i=1..N_t} \in \mathbb{R}^k$) dedicated to generate diverse trajectories in terms of shape (resp. time).

When training $\text{STRIPE}^{++}_{\text{shape}}$ (the description for $\text{STRIPE}^{++}_{\text{time}}$ is similar), we concatenate h with the posterior time latent code μ_t^* and the N_s shape latent codes z_s^i provided by $\text{STRIPE}^{++}_{\text{shape}}$, which leads to N_s future trajectories $\hat{\mathbf{y}}^i = \text{Decoder}\left((h, z_s^i, \mu_t^*)\right)$, $i = 1..N_s$[3]. The shape diversity of this set of N_s trajectories is then enforced by a shape diversity loss that we describe below.

DPP diversity loss: We resort to determinantal point processes (DPP) for their appealing properties for maximizing the diversity of a set of items $\mathcal{Y} = \{\mathbf{y}_1, ..., \mathbf{y}_N\}$ given a fixed sampling budget N and for structuring diversity via the choice of the DPP kernel. Following [1], we minimize the negative expected cardinality of a random subset Y from the DPP:

$$\mathcal{L}_{diversity}(\mathcal{Y}; \mathbf{K}) = -\mathbb{E}_{Y \sim \text{DPP}(\mathcal{K})}|Y| \quad (13)$$
$$= -Tr(\mathbf{I} - (\mathbf{K} + \mathbf{I})^{-1}) \quad (14)$$

Intuitively, a larger expected cardinality means a more diverse sampled set according to kernel $\mathcal{K}$. We provide more details on DPPs and the derivation of $\mathcal{L}_{diversity}$ in Appendix E.1. This loss is differentiable and can be computed in closed form.

Quality regularizer in the DPP: When training the shape and time proposal networks with the diversity loss, we do not have control over the quality of predictions, which can deteriorate to improve diversity. To address this, we introduce a quality regularization term in the DPP kernels. Crucially, we decouple the criteria used for quality (DILATE) and diversity (shape or time). $\mathcal{K}_{shape}$ maximizes the shape (DTW) diversity, while maintaining a globally low DILATE loss (thus playing on the temporal localization to ensure a good tradeoff). This contrasts with [1] which uses the same MSE criterion for both quality and diversity (see Appendix E.3 for a detailed explanation). In practice, we introduce a quality

3. If there exists multiple futures as supervision, we repeat this operation for each posterior latent code $\mu_t^{*,j}$ (it corresponds to consider each tuple $(\mathbf{x}_{1:T}, \mathbf{y}^{*,j})$ as a separate training example).

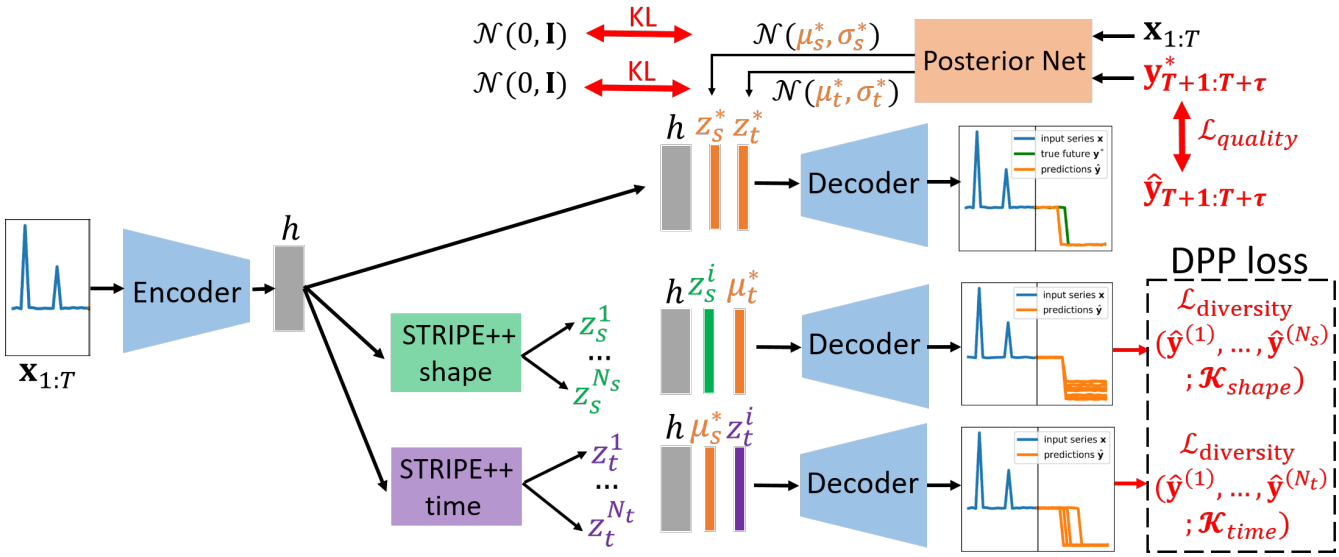


Fig. 3: Our STRIPE++ model builds upon a forecasting architecture trained with a quality loss $\mathcal{L}_{quality}$ enforcing sharp predictions. The latent state is disentangled into a deterministic part h from the encoder and two stochastic codes z_s and z_t that account for the shape and time variations. First step (figure upper part), we train the predictor with a quality loss, the stochastic codes are sampled from a posterior network. Second step (bottom part), we diversify the predictions with two STRIPE++ shape and time proposal networks trained with a DPP diversity loss (keeping the encoder and decoder frozen).

vector $\mathbf{q} = (q_1, \dots, q_{N_s})$ between the prediction $\hat{\mathbf{y}}^i$ and the ground truth $\mathbf{y}^*$ [4]. We choose $q_i = \mu(1 - \text{DILATE}(\hat{\mathbf{y}}^i, \mathbf{y}^*))$, where $\mu > 0$ is a hyperparameter to tune the influence of the quality regularization. The modified shape kernel becomes (and similarly for the time kernel):

$$\tilde{\mathbf{K}}_{shape} = \text{Diag}(\mathbf{q})\ \mathbf{K}_{shape}\ \text{Diag}(\mathbf{q}) \tag{15}$$

This decomposition enables to sample sets of items of both high quality and diversity:

$$\mathcal{P}_{\tilde{\mathbf{K}}}(\mathbf{Y} = Y) \propto \left(\prod_{i \in Y} q_i^2\right) \det(\mathbf{K}_Y) \tag{16}$$

We then train STRIPE$^{++}_{\text{shape}}$ by applying the shape kernel $\tilde{\mathbf{K}}_{shape}$ (Eq. 15) to the set of N_s shape future trajectories $\mathcal{L}_{diversity}(\hat{\mathbf{y}}^1, \dots, \hat{\mathbf{y}}^{N_s}; \tilde{\mathbf{K}}_{shape})$ and STRIPE$^{++}_{\text{time}}$ by applying the time kernel $\tilde{\mathbf{K}}_{time}$ to the set of N_t time future trajectories $\mathcal{L}_{diversity}(\hat{\mathbf{y}}^1, \dots, \hat{\mathbf{y}}^{N_t}; \tilde{\mathbf{K}}_{time})$.

4.2.3 Diverse trajectory generation at test time

At test time, the posterior network is discarded and we only rely on the trained encoder, STRIPE$^{++}_{\text{shape}}$, STRIPE$^{++}_{\text{time}}$ proposal networks and decoder to generate future predictions. More precisely, we combine the shape and temporal proposals $\{z_s^i\}_{i=1..N_s}$ and $\left\{z_t^j\right\}_{j=1..N_t}$ to obtain $N_s \times N_t$ predictions $\hat{\mathbf{y}}^{i,j} = \text{Decoder}((h, z_s^i, z_t^j))$.

4.2.4 Discussion: differences to prior work STRIPE

Our proposed STRIPE++ model improves over STRIPE [25] by gaining a better control on quality at the diversification stage. This leads to forecasts of better quality while maintaining the diversity performances (see the experiments in Table 3). This is attained with two mechanisms described above:

the use of a posterior network which provides the diversification variables z_s^* and z_t^* attached to a particular true future trajectory $\mathbf{y}^*$. This ensures a better disentanglement between the deterministic latent variable h and the stochastic diversification variable z_s and z_t. In contrast in STRIPE [25], the z_t variables are random standard Gaussian samples when training STRIPE-shape.the quality constraint in the diversity kernels $\tilde{\mathbf{K}}_{shape}$ and $\tilde{\mathbf{K}}_{time}$ which enables to favour predictions of both good quality and diversity.

Note that an important additional benefit of using a posterior net at training time is the ability to accommodate multiple future supervision trajectories, whereas in [25] the diversifying variables are zeroed when training the predictor, meaning that the future ambiguity would be captured in h, and not in z_s and z_t.

Another minor difference with STRIPE [25] is that we discarded the conditioning of STRIPE-time on the shape codes generated by STRIPE-shape (which leads to no experimental benefits over a non-conditional proposal network).

5 Experiments

We evaluate our approach on the two tasks presented in introduction: deterministic and probabilistic time series forecasting, focusing particularly on non-stationary series that can present sudden changes.

5.1 Datasets

We carry out experiments on 6 synthetic and real-world datasets from various domains to illustrate the broad applicability of our methods. For each dataset, the task is to predict

4. If there are multiple futures as supervision, we again consider each tuple (input sequence, possible future) as a separate training example.

TABLE 1: **DILATE forecasting results on generic MLP and RNN architectures**, averaged over 10 runs (mean $\pm$ standard deviation). Metrics are scaled for readability. For each experiment, best method(s) (Student t-test) in bold.

		Fully connected network (MLP)			Recurrent neural network (Seq2Seq)		
Dataset	Eval \ Train	MSE	DTW$^{\Delta}_{\gamma}$ [19]	DILATE (ours)	MSE	DTW$^{\Delta}_{\gamma}$ [19]	DILATE (ours)
Synthetic	MSE (x1000)	**16.5 $\pm$ 1.4**	48.2 $\pm$ 4.0	**16.7$\pm$ 1.8**	**11.0 $\pm$ 1.7**	23.1 $\pm$ 4.5	**12.1 $\pm$ 1.3**
	DTW (x10)	38.6 $\pm$ 1.28	**27.3 $\pm$ 1.37**	32.1 $\pm$ 5.33	**24.6 $\pm$ 1.20**	**22.7 $\pm$ 3.55**	**23.1 $\pm$ 2.44**
	TDI (x10)	15.3 $\pm$ 1.39	26.9 $\pm$ 4.16	**13.8 $\pm$ 0.71**	17.2 $\pm$ 1.22	20.0 $\pm$ 3.72	**14.8 $\pm$ 1.29**
	Ramp (x10)	5.21 $\pm$ 0.10	**2.04 $\pm$ 0.23**	3.41 $\pm$ 0.29	5.80 $\pm$ 0.10	**4.27 $\pm$ 0.8**	4.99 $\pm$ 0.46
	Hausdorff (x1)	4.04 $\pm$ 0.28	4.71 $\pm$ 0.50	**3.71 $\pm$ 0.12**	2.87 $\pm$ 0.13	3.45 $\pm$ 0.32	**2.70 $\pm$ 0.17**
ECG	MSE (x100)	**31.5 $\pm$ 1.39**	70.9 $\pm$ 37.2	37.2 $\pm$ 3.59	**21.2 $\pm$ 2.24**	75.1 $\pm$ 6.30	30.3 $\pm$ 4.10
	DTW (x10)	19.5 $\pm$ 0.16	18.4 $\pm$ 0.75	**17.7 $\pm$ 0.43**	17.8 $\pm$ 1.62	17.1 $\pm$ 0.65	**16.1 $\pm$ 0.16**
	TDI (x10)	**7.58 $\pm$ 0.19**	17.9 $\pm$ 0.7	**7.21 $\pm$ 0.89**	8.27 $\pm$ 1.03	27.2 $\pm$ 11.1	**6.59 $\pm$ 0.79**
	Ramp (x1)	**4.9 $\pm$ 0.1**	5.1 $\pm$ 0.3	**5.0 $\pm$ 0.1**	**4.84 $\pm$ 0.24**	**4.79 $\pm$ 0.37**	**4.80 $\pm$ 0.25**
	Hausdorff (x1)	**4.1 $\pm$ 0.1**	6.3 $\pm$ 0.6	4.7 $\pm$ 0.3	**4.32 $\pm$ 0.51**	6.16 $\pm$ 0.85	**4.23 $\pm$ 0.41**
Traffic	MSE (x1000)	**6.58 $\pm$ 0.11**	25.2 $\pm$ 2.3	19.3 $\pm$ 0.80	**8.90 $\pm$ 1.1**	22.2 $\pm$ 2.6	**10.0 $\pm$ 2.6**
	DTW (x100)	25.2 $\pm$ 0.17	**23.4 $\pm$ 5.40**	**23.1 $\pm$ 0.41**	24.6 $\pm$ 1.85	**22.6 $\pm$ 1.34**	**23.0 $\pm$ 1.62**
	TDI (x100)	24.8 $\pm$ 1.1	27.4 $\pm$ 5.01	**16.7 $\pm$ 0.51**	**15.4 $\pm$ 2.25**	22.3 $\pm$ 3.66	**14.4$\pm$ 1.58**
	Ramp (x10)	6.18 $\pm$ 0.1	**5.59 $\pm$ 0.1**	**5.6 $\pm$ 0.1**	6.29 $\pm$ 0.32	**5.78 $\pm$ 0.41**	**5.93 $\pm$ 0.24**
	Hausdorff (x1)	**1.99 $\pm$ 0.2**	**1.91 $\pm$ 0.3**	**1.94 $\pm$ 0.2**	**2.16 $\pm$ 0.38**	**2.29 $\pm$ 0.33**	**2.13 $\pm$ 0.51**

TABLE 2: **DILATE forecasting results on state-of-the-art architectures N-Beats [16] and Informer [17]**. Evaluation metrics are scaled for readability. Results are averaged over 10 runs, best(s) method(s) in bold (Student t-test).

Dataset	Model	MSE	DTW	Ramp	TDI	Hausdorff	DILATE
Synthetic	N-Beats [16] MSE	**13.6 $\pm$ 0.5**	24.9 $\pm$ 0.6	5.9 $\pm$ 0.1	**13.8 $\pm$ 1.1**	**2.8 $\pm$ 0.1**	**19.3 $\pm$ 0.5**
	N-Beats [16] DILATE	**13.3 $\pm$ 0.7**	**23.4 $\pm$ 0.8**	**4.8 $\pm$ 0.4**	**14.4 $\pm$ 1.3**	**2.7 $\pm$ 0.5**	**18.9 $\pm$ 0.8**
	Informer [17] MSE	**10.4 $\pm$ 0.3**	20.1 $\pm$ 1.1	4.3 $\pm$ 0.3	13.1 $\pm$ 0.9	**2.5 $\pm$ 0.1**	16.6 $\pm$ 0.8
	Informer [17] DILATE	11.8 $\pm$ 0.7	**18.5 $\pm$ 1.2**	**2.4 $\pm$ 0.3**	**11.6 $\pm$ 0.9**	**2.4 $\pm$ 0.9**	**15.1 $\pm$ 0.7**
Electricity	N-Beats [16] MSE	**24.8 $\pm$ 0.4**	**15.6 $\pm$ 0.2**	**13.3 $\pm$ 0.3**	4.6 $\pm$ 0.1	**2.6 $\pm$ 0.3**	**13.4 $\pm$ 0.2**
	N-Beats [16] DILATE	25.8 $\pm$ 0.9	**15.5 $\pm$ 0.2**	**13.3 $\pm$ 0.3**	**4.4 $\pm$ 0.2**	3.1 $\pm$ 0.5	**13.2 $\pm$ 0.2**
	Informer [17] MSE	**38.1 $\pm$ 2.1**	18.9 $\pm$ 0.6	13.2 $\pm$ 0.2	6.5 $\pm$ 0.3	2.1 $\pm$ 0.2	16.4 $\pm$ 0.5
	Informer [17] DILATE	**37.8 $\pm$ 0.8**	**18.5 $\pm$ 0.3**	**12.9 $\pm$ 0.2**	**5.7 $\pm$ 0.2**	**1.9 $\pm$ 0.1**	**15.9 $\pm$ 0.3**
ETTH1 (96)	N-Beats [16] MSE	32.5 $\pm$ 1.4	3.9 $\pm$ 0.2	13.3 $\pm$ 2.0	21.6 $\pm$ 4.3	**5.7 $\pm$ 0.7**	7.4 $\pm$ 1.0
	N-Beats [16] DILATE	**26.0 $\pm$ 2.8**	**2.9 $\pm$ 0.1**	**4.6 $\pm$ 0.6**	**11.4 $\pm$ 1.7**	**6.4 $\pm$ 1.0**	**4.6 $\pm$ 0.4**
	Informer [17] MSE	**28.2 $\pm$ 2.6**	4.3 $\pm$ 0.3	5.8 $\pm$ 0.1	21.6 $\pm$ 3.3	**6.6 $\pm$ 1.9**	7.8 $\pm$ 0.9
	Informer [17] DILATE	32.5 $\pm$ 3.8	**3.2 $\pm$ 0.3**	**4.5 $\pm$ 0.3**	**19.1 $\pm$ 1.9**	**6.4 $\pm$ 1.0**	**6.4 $\pm$ 0.6**

the τ-steps ahead future trajectory (or multiple trajectories in the probabilistic case) given a T-steps context window:

- **Synthetic-det** ($T = 20, \tau = 20$): deterministic dataset consisting in predicting sudden changes (step functions) based on an input signal composed of two peaks. This controlled setup was designed to measure precisely the shape and time errors of predictions (see Figure 4).
- **Synthetic-prob** ($T = 20, \tau = 20$): probabilistic version of the previous dataset where for each input series, there exists 10 admissible futures obtained by varying the amplitude and the timing of the step (see Figure 1 (b)).
- **ECG5000** ($T = 84, \tau = 56$): dataset composed of 5000 electrocardiograms (ECG) with sharp spikes whose time intervals are of great importance (see Figure 4).
- **Traffic** ($T = 168, \tau = 24$) dataset consisting in hourly occupancy rates in California (see Figure 4).
- **Electricity** ($T = 168, \tau = 24$): consisting in hourly electricity consumption measurements (kWh) from 370 customers.
- **ETTh1** [17] ($T = 96, \tau = 96$): dataset of hourly Electricity Transformer Temperature measurements, which is an important indicator for electricity grids. This dataset enables to assess the generalization of our approach on much longer term predictions.

We give more details for each dataset in Appendix C.

5.2 Deterministic forecasting with the DILATE loss

To evaluate the benefits of our proposed DILATE training loss, we compare it against the widely used Euclidean (MSE) loss, and the smooth DTW introduced in [19], [54]. We use the following multi-step prediction metrics: MSE, DTW (shape), TDI (temporal). To consolidate the evaluation, we also consider two additional (non differentiable) metrics for assessing shape and time. For shape, we compute the ramp score [9]. For time, we compute the Hausdorff distance between a set of detected change points in the target signal $\mathcal{T}^*$ and in the predicted signal $\hat{\mathcal{T}}$:

$$\text{Hausdorff}(\mathcal{T}^*, \hat{\mathcal{T}}) := \max(\max_{\hat{t} \in \hat{\mathcal{T}}} \min_{t^* \in \mathcal{T}^*} |\hat{t} - t^*|, \max_{t^* \in \mathcal{T}^*} \min_{\hat{t} \in \hat{\mathcal{T}}} |\hat{t} - t^*|) \tag{17}$$

which corresponds to the largest possible distance between a change point and its prediction. Additional details about these external metrics, the architectures and implementation of models used in this section are given in Appendix D.

5.2.1 *DILATE performances on generic architectures*

To demonstrate the broad applicability of our approach, we first perform multi-step forecasting with two generic neural network architectures: a fully connected network (1 layer of 128 neurons), which does not make any assumption on data structure, and a more specialized Seq2Seq model with

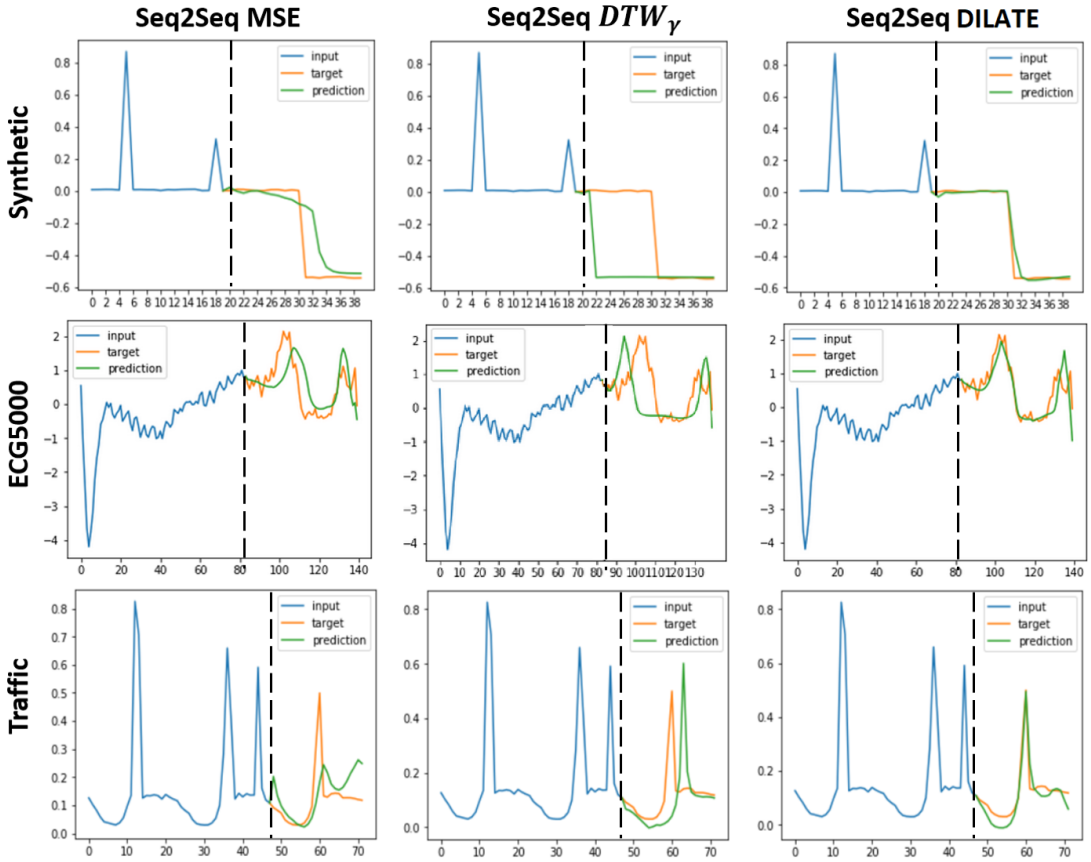


Fig. 4: **Qualitative prediction results with the DILATE loss.** For each dataset, the MSE training loss leads to non-sharp predictions, whereas the soft-DTW loss can predict sharp variations but has no control over their temporal localization. In contrast, the DILATE loss produces sharp predictions with accurate temporal localization.

1 layer of 128 Gated Recurrent Units (GRU). We perform a Student t-test with significance level 0.05 to highlight the best(s) method(s) in each experiment (averaged over 10 runs). Overall results are presented in Table 1.

Comparison to MSE training loss: DILATE outperforms MSE when evaluated on shape (DTW) in all experiments, with significant differences on 5/6 experiments. When evaluated on time (TDI), DILATE also performs better in all experiments (significant differences on 3/6 tests). Finally, DILATE is equivalent to MSE when evaluated on MSE on 3/6 experiments.

Comparison to DTW_γ^Δ training loss: When evaluated on shape (DTW), DILATE performs similarly to DTW_γ^Δ (2 significant improvements, 1 significant drop and 3 equivalent performances). For time (TDI) and MSE evaluations, DILATE is significantly better than DTW_γ^Δ in all experiments, as expected.

We can notice that the ramp score (resp. the Haussdorff distance) provides the same trends than the shape metric DTW (resp. the time metric TDI). It reinforces our conclusions and shows that DILATE indeed improves shape and temporal accuracy beyond the metrics being optimized.

We display a few qualitative examples for Synthetic, ECG5000 and Traffic datasets in Figure 4 (other examples are provided in Appendix D.5). We see that MSE training leads to predictions that are non-sharp, making them inadequate in presence of drops or sharp spikes. DTW_γ^Δ leads to very sharp predictions in shape, but with a possibly large temporal misalignment. In contrast, our DILATE loss predicts series that have both a correct shape and precise temporal localization.

5.2.2 DILATE performances with state-of-the-art models

Beyond generic forecasting architectures, we show that DILATE can also improve the performances of state-of-the-art deep architectures. We experiment here with two recent and competitive models: N-Beats [16] and Informer [17]. Results in Table 2 are consistent with those in Table 1: models trained with DILATE improve over MSE in shape (in DTW and ramp score for 6/6 experiments) and time (in TDI for 5/6 and Hausdorff for 4/6 experiments) and are equivalent to MSE when evaluated in MSE (equivalent or better for 3/6 experiments). We provide qualitative predictions of N-Beats on Electricity in Appendix D.3.

5.2.3 DILATE loss analysis

Impact of α: We analyze in Figure 5 the influence of the tradeoff parameter α when training a Seq2Seq model on the synthetic-det dataset. When $\alpha = 1$, $\mathcal{L}_{\text{DILATE}}$ reduces

to $\text{DTW}_\gamma^{\mathbf{\Delta}}$, with an accurate shape but a large temporal error. When $\alpha \longrightarrow 0$, we only minimize $\mathcal{L}_{time}$ without any shape constraint. Both MSE and shape errors explode in this case, illustrating the fact that $\mathcal{L}_{time}$ is only meaningful in conjunction with $\mathcal{L}_{shape}$. Both the MSE and DILATE error curves present a U-shape ; in this case, $\alpha = 0.5$ seems an acceptable tradeoff for the Synthetic-det dataset.

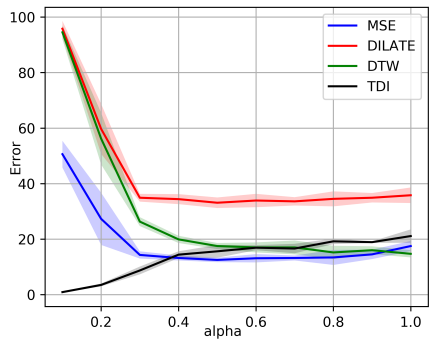


Fig. 5: **Influence of α in the DILATE loss.** The shaded areas represent $\pm$ std computed over 10 runs.

Influence of γ: We analyse the influence of the $\text{DTW}_\gamma^{\mathbf{\Delta}}$ smoothing parameter γ in Figure 6. We represent (bottom of the figure) the assignment probabilities of the $\text{DTW}_\gamma^{\mathbf{\Delta}}$ path between two given test time series, the true DTW path being depicted in red. When γ increases, the $\text{DTW}_\gamma^{\mathbf{\Delta}}$ path is more uncertain and becomes multimodal. When $\gamma \longrightarrow 0$, the soft DTW converges toward the true DTW. However, we see (top of the figure) that for small γ values, optimizing $\text{DTW}_\gamma^{\mathbf{\Delta}}$ becomes more difficult, resulting in higher test error and higher variance (on the synthetic-det dataset). We fixed $\gamma = 10^{-2}$ in all our experiments, which yields a good tradeoff between an accurate soft optimal path and a low test error.

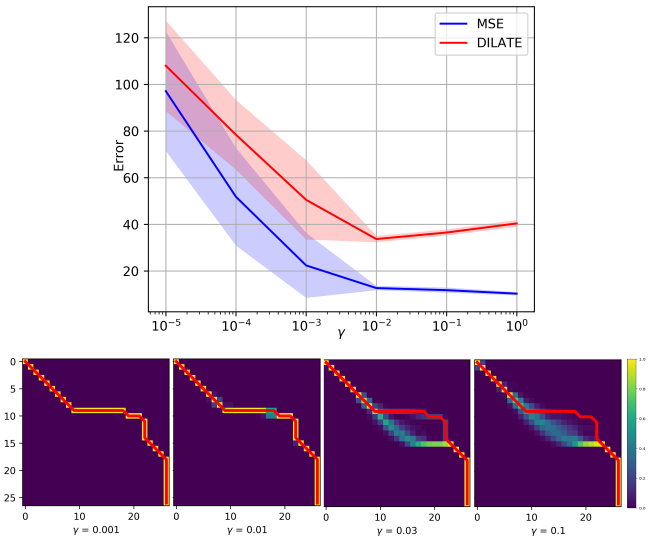


Fig. 6: **Influence of γ in the DILATE loss.** The shaded areas represent $\pm$ std computed over 10 runs.

5.3 Probabilistic forecasting with STRIPE++

We firstly assess the ability of STRIPE++ to capture the full predictive distribution of future trajectories. To do so, we need for evaluation the ground truth set of admissible futures for a given input ; we use here the synthetic-prob dataset (described in section 5.1) constructed for this purpose. Secondly, on a more realistic setting where we only know one future for each input, we evaluate STRIPE++ on the Traffic and Electricity datasets with the best (resp. the mean) sample metrics as a proxy for diversity (resp. quality). We describe the implementation details and neural network architectures (encoder, decoder, posterior net and STRIPE++ proposal network) in Appendix E.2.

5.3.1 Full predictive distribution evaluation on synthetic-prob

Metrics: To assess the discrepancy between the predicted and true distributions of futures trajectories, we define two measures $\text{H}_{quality}(\ell)$ and $\text{H}_{diversity}(\ell)$ (ℓ = DTW, TDI or DILATE in our experiments):

$$\text{H}_{quality}(\ell) := \mathbb{E}_{\mathbf{x}\in\mathcal{D}_{test}}\mathbb{E}_{\hat{\mathbf{y}}}\left[\inf_{\mathbf{y}\in F(\mathbf{x})} \ell(\hat{\mathbf{y}},\mathbf{y})\right] \tag{18}$$

$$\text{H}_{diversity}(\ell) := \mathbb{E}_{\mathbf{x}\in\mathcal{D}_{test}}\mathbb{E}_{\mathbf{y}\in F(\mathbf{x})}\left[\inf_{\hat{\mathbf{y}}} \ell(\hat{\mathbf{y}},\mathbf{y})\right] \tag{19}$$

$$F1 \text{ score} = \frac{2\,\text{H}_{quality}(\ell)\cdot\text{H}_{diversity}(\ell)}{\text{H}_{quality}(\ell)+\text{H}_{diversity}(\ell)} \tag{20}$$

$\text{H}_{quality}$ penalizes forecasts $\hat{\mathbf{y}}$ that are far away from a ground truth future of $\mathbf{x}$ denoted $\mathbf{y} \in F(\mathbf{x})$ (similarly to the *precision* concept in pattern recognition) whereas $\text{H}_{diversity}$ penalizes when a true future is not covered by a forecast (similarly to *recall*). As a tradeoff balancing quality and diversity, we compute the F1 score in Eq. 20. In addition, we also use the continuous ranked probability score (CRPS) which is a standard *proper scoring rule* [44] for assessing probabilistic forecasts [41]. Intuitively, the CRPS is the pinball loss integrated over all quantile levels. A key property is that the CRPS attains its minimum when the predicted future distribution equals the true future distribution, making this metric particularly adapted to our context.

Forecasting results: We compare in Table 3 our method to 4 recent competing diversification mechanisms (variety loss [61], entropy regularisation [62], diverse DPP [1] and GDPP [68]) based on a conditional variational autoencoder (cVAE) backbone trained with DILATE. We observe that STRIPE and STRIPE++ obtain the global best performances by improving diversity by a large amount ($\text{H}_{diversity}$(DILATE)=17.6) compared to the backbone cVAE DILATE ($\text{H}_{diversity}$(DILATE)=33.9) and to other diversification schemes (the best competitor GDPP [68] attains $\text{H}_{diversity}$(DILATE)=23.9). This highlights the relevance of the structured shape and time diversity. We can also notice that, in contrast to competing diversification schemes that improve diversity at the cost of a loss in quality, STRIPE++ maintains high quality predictions. STRIPE++ is only beaten in $\text{H}_{quality}$(DILATE) by GDPP [68], but this method is significantly worse than STRIPE++ in diversity, and GDPP requires full future distribution supervision, which it not applicable in real datasets (see section 5.3.3). All in all, the F1 scores summarize the quality

TABLE 3: **STRIPE++ forecasting results on the synthetic dataset with multiple futures**, averaged over 5 runs (mean $\pm$ std). Best equivalent methods (Student t-test) shown in bold. Metrics are scaled (MSE $\times$ 1000, DILATE $\times$100, CRPS $\times$ 1000).

Methods	$\mathrm{H}_{quality}(\cdot)$ ($\downarrow$)			$\mathrm{H}_{diversity}(\cdot)$ ($\downarrow$)			$F1$ score ($\downarrow$)			CRPS ($\downarrow$)
	DTW	TDI	DILATE	DTW	TDI	DILATE	DTW	TDI	DILATE	
DeepAR [15]	42.9 $\pm$ 6.6	16.6 $\pm$ 7.6	33.5 $\pm$ 6.0	23.9 $\pm$ 3.5	**12.8 $\pm$ 2.5**	22.7 $\pm$ 2.2	30.7	14.5	27.1	62.4 $\pm$ 9.9
cVAE DILATE	**11.7 $\pm$ 1.5**	9.4 $\pm$ 2.2	**14.2 $\pm$ 1.5**	18.8 $\pm$ 1.3	48.6 $\pm$ 2.2	33.9 $\pm$ 3.9	14.4	15.7	20.0	62.2 $\pm$ 4.2
variety loss [61] DILATE	15.6 $\pm$ 3.4	10.2 $\pm$ 1.1	16.8 $\pm$ 0.9	22.7 $\pm$ 4.1	37.7 $\pm$ 4.9	30.8 $\pm$ 1.0	18.5	16.1	21.7	62.6 $\pm$ 3.0
entropy reg. [62] DILATE	13.8 $\pm$ 3.1	8.8 $\pm$ 2.2	**15.0 $\pm$ 1.6**	20.4 $\pm$ 2.8	42.0 $\pm$ 7.8	32.6 $\pm$ 2.3	16.5	14.5	20.5	62.4 $\pm$ 3.9
Diverse DPP [1] DILATE	**12.9 $\pm$ 1.2**	9.8 $\pm$ 2.1	15.1 $\pm$ 1.5	18.6 $\pm$ 1.6	42.8 $\pm$ 10.1	31.3 $\pm$ 5.7	15.2	15.9	20.4	60.7 $\pm$ 1.6
GDPP [68] DILATE	14.8 $\pm$ 2.9	11.7 $\pm$ 8.4	**14.4 $\pm$ 2.1**	20.8 $\pm$ 2.4	25.2 $\pm$ 7.2	23.9 $\pm$ 4.5	17.3	15.9	17.9	63.4 $\pm$ 6.4
STRIPE [25]	16.8 $\pm$ 0.5	**6.7 $\pm$ 0.4**	15.4 $\pm$ 0.5	16.1 $\pm$ 1.1	**13.2 $\pm$ 1.7**	**17.7 $\pm$ 0.6**	16.4	**8.8**	16.5	60.5 $\pm$ 0.4
STRIPE++	13.5 $\pm$ 0.5	9.2 $\pm$ 0.5	**15.0 $\pm$ 0.3**	**12.9 $\pm$ 0.3**	16.3 $\pm$ 1.2	**17.9 $\pm$ 0.6**	**13.2**	11.7	**16.3**	**48.6 $\pm$ 0.6**

TABLE 4: **Probabilistic forecasting results on the Traffic and Electricity datasets**, averaged over 5 runs (mean $\pm$ std). Metrics are scaled for readability. Best equivalent method(s) (Student t-test) shown in bold.

	Traffic				Electricity			
	MSE		DILATE		MSE		DILATE	
Method	mean	best	mean	best	mean	best	mean	best
Nbeats [16] MSE	-	7.8 $\pm$ 0.3	-	22.1 $\pm$ 0.8	-	24.8 $\pm$ 0.4	-	20.2 $\pm$ 0.3
Nbeats [16] DILATE	-	17.1 $\pm$ 0.8	-	17.8 $\pm$ 0.3	-	25.8 $\pm$ 0.9	-	19.9 $\pm$ 0.5
Deep AR [15]	15.1 $\pm$ 1.7	**6.6 $\pm$ 0.7**	30.3 $\pm$ 1.9	16.9 $\pm$ 0.6	67.6 $\pm$ 5.1	25.6 $\pm$ 0.4	59.8 $\pm$ 5.2	17.2 $\pm$ 0.3
cVAE DILATE	**10.0 $\pm$ 1.7**	8.8 $\pm$ 1.6	**19.1 $\pm$ 1.2**	17.0 $\pm$ 1.1	**28.9 $\pm$ 0.8**	27.8 $\pm$ 0.8	24.6 $\pm$ 1.4	22.4 $\pm$ 1.3
Variety loss [61]	**9.8 $\pm$ 0.8**	7.9 $\pm$ 0.8	**18.9 $\pm$ 1.4**	15.9 $\pm$ 1.2	29.4 $\pm$ 1.0	27.7 $\pm$ 1.0	24.7 $\pm$ 1.1	21.6 $\pm$ 1.0
Entropy regul. [62]	11.4 $\pm$ 1.3	10.3 $\pm$ 1.4	**19.1 $\pm$ 1.4**	16.8 $\pm$ 1.3	34.4 $\pm$ 4.1	32.9 $\pm$ 3.8	29.8 $\pm$ 3.6	25.6 $\pm$ 3.1
Diverse DPP [1]	11.2 $\pm$ 1.8	6.9 $\pm$ 1.0	20.5 $\pm$ 1.0	14.7 $\pm$ 1.0	31.5 $\pm$ 0.8	25.8 $\pm$ 1.3	26.6 $\pm$ 1.0	19.4 $\pm$ 1.0
STRIPE [25]	**10.1 $\pm$ 0.4**	**6.5 $\pm$ 0.2**	**19.2 $\pm$ 0.8**	**14.2 $\pm$ 0.2**	29.7 $\pm$ 0.3	**23.4 $\pm$ 0.2**	24.4 $\pm$ 0.3	**16.9 $\pm$ 0.2**
STRIPE++	**10.0 $\pm$ 0.2**	**6.7 $\pm$ 0.3**	**19.0 $\pm$ 0.2**	**14.1 $\pm$ 0.3**	**29.5 $\pm$ 0.3**	**23.6 $\pm$ 0.4**	**24.1 $\pm$ 0.2**	17.3 $\pm$ 0.4

vs. diversity tradeoffs, and STRIPE++ gets the best F1 DILATE score. Moreover, STRIPE++ outperforms all other methods with the CRPS metric, indicating that the predicted future trajectory distribution is closer to the ground truth one.

Discussion on quality regularization: As discussed above, the key difference between STRIPE and STRIPE++ consists in an additional quality constraint during the diversification stage. We observe that STRIPE++ indeed produces globally better quality forecasts ($\mathrm{H}_{quality}$(DILATE)= 15.0 for STRIPE++ vs. 15.5 for STRIPE) while maintaining the level of diversity. It leads to a better DILATE F1 score and a better CRPS for STRIPE++.

5.3.2 Ablation study

To analyze the respective roles of the quality and diversity losses, we perform an ablation study on the synthetic-prob dataset with the cVAE backbone trained with the quality loss DILATE and different DPP diversity losses. For a finer analysis, we report in Table 5 the shape (DTW, computed with Tslearn [70]) and time (TDI) components of the DILATE loss [24].

Results presented in Table 5 first reveal the crucial importance to define different criteria for quality and diversity. With the same loss for quality and diversity (as this is the case in [1]), we observe here that the DILATE DPP kernel $\exp(-\text{DILATE})$ does not bring a significant diversity gain compared to the cVAE DILATE baseline (without diversity loss). By choosing the MSE kernel instead, we even get a small diversity and quality improvement.

In contrast, our introduced shape and time kernels $\mathcal{K}_{shape}$ and $\mathcal{K}_{time}$ largely improve the diversity in DILATE without deteriorating quality. As expected, each kernel brings its own

TABLE 5: **Ablation study on the synthetic-prob dataset**. We train a backbone cVAE with the DILATE quality loss and compare different DPP kernels for diversity.

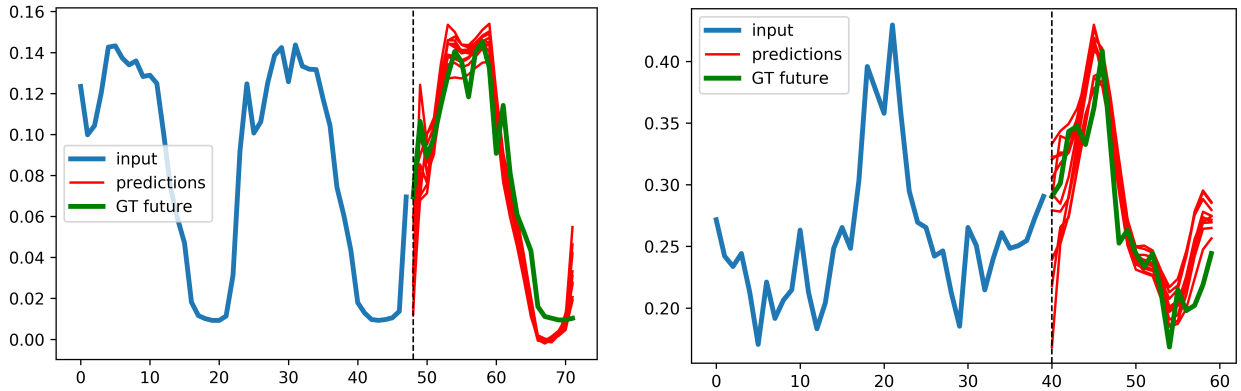
	$\mathrm{H}_{quality}$ ($\downarrow$)	$\mathrm{H}_{diversity}(\cdot)$ ($\downarrow$)		
diversity	DILATE	DTW	TDI	DILATE
None	**14.2 $\pm$ 1.5**	18.8 $\pm$ 1.3	48.6 $\pm$ 2.2	33.9 $\pm$ 3.9
DILATE	**15.1 $\pm$ 1.9**	18.6 $\pm$ 1.6	42.8 $\pm$ 10.2	31.3 $\pm$ 10.7
MSE	**15.1 $\pm$ 1.4**	18.5 $\pm$ 1.3	41.9 $\pm$ 8.8	30.8 $\pm$ 4.7
shape	**15.1 $\pm$ 0.6**	16.4 $\pm$ 1.5	15.4 $\pm$ 4.2	18.9 $\pm$ 1.8
time	**15.6 $\pm$ 0.6**	17.6 $\pm$ 0.5	**15.1 $\pm$ 3.1**	19.4 $\pm$ 1.6
STRIPE++	**15.0 $\pm$ 0.3**	**12.9 $\pm$ 0.3**	**16.3 $\pm$ 1.2**	**17.9 $\pm$ 0.6**

benefits: $\mathcal{K}_{shape}$ brings the best improvement in the shape metric DTW ($\mathrm{H}_{diversity}$(DTW) $=$ 16.4 vs. 18.8) and $\mathcal{K}_{time}$ the best improvement in the time metric TDI ($\mathrm{H}_{diversity}$(TDI) $=$ 15.1 vs. 48.6). STRIPE++ gathers the benefits of both criteria and gets the global best results in diversity and equivalent results in quality.

5.3.3 State-of-the-art comparison on real-world datasets

We evaluate here the performances of STRIPE++ on two challenging real-world datasets Traffic and Electricity commonly used as benchmarks in the time series forecasting literature [14], [15], [24], [34], [45], [71]. Contrary to the synthetic-prob dataset, we only dispose of one future trajectory sample $\mathbf{y}^*_{T+1:T+\tau}$ for each input series $\mathbf{x}_{1:T}$. In this case, the metric $\mathrm{H}_{quality}$ (resp. $\mathrm{H}_{diversity}$) defined in section 5.3.1 reduces to the mean sample (resp. best sample), which are common for evaluating stochastic forecasting models [72], [73].

Results in Table 4 reveal that STRIPE and STRIPE++ outperform all other baselines in the best sample (evaluated in MSE or DILATE). Our method even outperforms in the best sample the state-of-the-art N-Beats algorithm [16]



(a) Traffic (c) Electricity

Fig. 7: STRIPE++ qualitative predictions on real-world datasets Traffic (a) and Electricity (b).

(either trained with MSE or DILATE), which is dedicated to producing high quality deterministic forecasts. In terms of quality (evaluation with the mean sample), STRIPE++ improves over STRIPE (as expected) and gets the best (or equivalently best) results in all cases. This contrasts to competing diversification methods, e.g. Diverse DPP [1], that deteriorate quality to improve diversity. Finally we notice that STRIPE++ is consistently better in diversity and quality than the state-of-the art probabilistic deep AR method [15].

We display a few qualitative forecasting examples of STRIPE++ on Figure 7. We observe that STRIPE++ predictions are both sharp and accurate: both the shape diversity (amplitude of the peaks) and temporal diversity match the ground truth future.

5.3.4 STRIPE++ analysis: quality-diversity cooperation

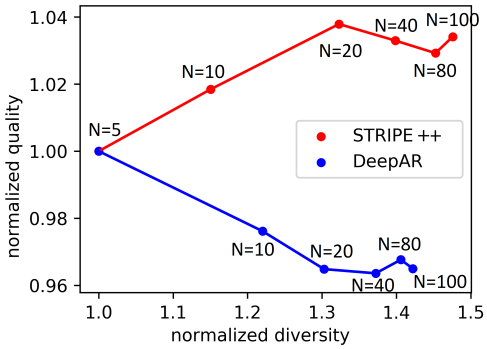


Fig. 8: Influence of the number N of trajectories on quality (higher is better) and diversity for the `synthetic-prob` dataset.

We analyze here the quality-diversity tradeoff with respect to the number N of sampled future trajectories. In Figure 8 we represent the evolution of performances when N increases from 5 to 100 on the synthetic-prob dataset. As expected, the normalized DILATE diversity $\mathrm{H}_{diversity}(5)/\mathrm{H}_{diversity}(N)$ (higher is better) increases with N for both STRIPE++ and deepAR models [15]. However we remark that STRIPE++ does not deteriorate normalized quality (which even increases slightly), in contrast to deepAR which does not have control over the targeted diversity. This again confirms the relevance of our approach that effectively combines an adequate quality loss function and a structured diversity mechanism. We provide an additional cooperation analysis in Appendix E.3 highlighting the importance to decouple the criteria used for quality and diversity.

6 Conclusion

In this work, we tackle the multi-step deep time series forecasting problem, in the challenging context of non-stationary series that can present sharp variations. In contrast to the majority of existing methods that train models with the surrogate MSE or variants, we propose to leverage shape and temporal criteria at training time. We introduce differentiable similarities and dissimilarities for characterizing shape accuracy and temporal localization error. We provide two implementations for time series forecasting: the DILATE loss function for deterministic forecasting that ensures both sharp predictions with accurate temporal localization, and the STRIPE++ model for probabilistic forecasting with shape and temporal diversity. We validate our claims with extensive experiments on synthetic and real-world datasets.

An interesting future perspective would be to incorporate seasonality and extrinsic prior knowledge (such as special events) [38] to better model the non-stationary abrupt changes and measure their impact on diversity and model confidence. Other appealing directions include diversity-promoting forecasting for exploration in reinforcement learning [74], [75], [76], and extension of shape/temporal criteria and structured diversity to spatio-temporal or video prediction tasks [73], [77], [78], [79].

References

[1] Y. Yuan and K. Kitani, "Diverse trajectory forecasting with determinantal point processes," *ICLR*, 2020.
[2] K. Rasul, A.-S. Sheikh, I. Schuster, U. Bergmann, and R. Vollgraf, "Multi-variate probabilistic time series forecasting via conditioned normalizing flows," *ICLR*, 2021.
[3] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.
[4] J. Durbin and S. J. Koopman, *Time series analysis by state space methods*. Oxford university press, 2012.

[5] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," *ICLR*, 2018.

[6] C. Snyder and M. Do, "Streets: A novel camera network dataset for traffic flow," in *NeurIPS*, 2019.

[7] S. Chauhan and L. Vig, "Anomaly detection in ecg time signals via deep long short-term memory networks," in *IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, 2015.

[8] X. Ding, Y. Zhang, T. Liu, and J. Duan, "Deep learning for event-driven stock prediction," in *IJCAI*, 2015.

[9] L. Vallance, B. Charbonnier, N. Paul, S. Dubost, and P. Blanc, "Towards a standardized procedure to assess solar forecast accuracy: A new ramp and time alignment metric," *Solar Energy*, 2017.

[10] A. Ghaderi, B. M. Sanandaji, and F. Ghaderi, "Deep forecast: Deep learning-based spatio-temporal forecasting," in *ICML Time Series Workshop*, 2017.

[11] V. Le Guen and N. Thome, "A deep physical model for solar irradiance forecasting with fisheye images," in *CVPR OmniCV workshop*, 2020.

[12] R. Yu, S. Zheng, and Y. Liu, "Learning chaotic dynamics using tensor recurrent neural networks," in *ICML Workshop on Deep Structured Prediction*, vol. 17, 2017.

[13] Y. Qin, D. Song, H. Cheng, W. Cheng, G. Jiang, and G. W. Cottrell, "A dual-stage attention-based recurrent neural network for time series prediction," in *IJCAI*, 2017.

[14] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, "Modeling long-and short-term temporal patterns with deep neural networks," in *ACM SIGIR*, 2018.

[15] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, "DeepAR: Probabilistic forecasting with autoregressive recurrent networks," *International Journal of Forecasting*, vol. 36, no. 3, pp. 1181–1191, 2020.

[16] B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio, "N-BEATS: Neural basis expansion analysis for interpretable time series forecasting," *ICLR*, 2020.

[17] 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," *AAAI*, 2021.

[18] L. Frías-Paredes, F. Mallor, M. Gastón-Romeo, and T. León, "Assessing energy forecasting inaccuracy by simultaneously considering temporal and absolute errors," *Energy Conv. and Management*, 2017.

[19] M. Cuturi and M. Blondel, "Soft-DTW: a differentiable loss function for time-series," in *ICML*, 2017.

[20] A. Florita, B.-M. Hodge, and K. Orwig, "Identifying wind and solar ramping events," in *IEEE GreenTech Conference*, 2013.

[21] Z. Wang and A. C. Bovik, "Mean squared error: Love it or leave it? a new look at signal fidelity measures," *IEEE signal processing magazine*, vol. 26, no. 1, pp. 98–117, 2009.

[22] D. Yang, S. Alessandrini, J. Antonanzas, F. Antonanzas-Torres, V. Badescu, H. G. Beyer, R. Blaga, J. Boland, J. M. Bright, C. F. Coimbra *et al.*, "Verification of deterministic solar forecasts," *Solar Energy*, vol. 210, pp. 20–37, 2020.

[23] H. Verbois, P. Blanc, R. Huva, Y.-M. Saint-Drenan, A. Rusydi, and A. Thiery, "Beyond quadratic error: Case-study of a multiple criteria approach to the performance assessment of numerical forecasts of solar irradiance in the tropics," *Renewable and Sustainable Energy Reviews*, vol. 117, p. 109471, 2020.

[24] V. Le Guen and N. Thome, "Shape and time distortion loss for training deep time series forecasting models," in *NeurIPS*, 2019.

[25] V. Le-Guen and N. Thome, "Probabilistic time series forecasting with structured shape and temporal diversity," in *NeurIPS*, 2020.

[26] S. Kohl, B. Romera-Paredes, C. Meyer, J. De Fauw, J. R. Ledsam, K. Maier-Hein, S. M. A. Eslami, D. Jimenez Rezende, and O. Ronneberger, "A probabilistic u-net for segmentation of ambiguous images," in *NeurIPS*, 2018.

[27] M. Blondel, A. Mensch, and J.-P. Vert, "Differentiable divergences between time series," *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021.

[28] R. Hyndman, A. B. Koehler, J. K. Ord, and R. D. Snyder, *Forecasting with exponential smoothing: the state space approach*. Springer Science & Business Media, 2008.

[29] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, 1997.

[30] S. B. Taieb and A. F. Atiya, "A bias and variance analysis for multistep-ahead time series forecasting," *IEEE Transactions on Neural Networks and Learning Systems*, 2016.

[31] I. Fox, L. Ang, M. Jaiswal, R. Pop-Busui, and J. Wiens, "Deep multi-output forecasting: Learning to accurately predict blood glucose trajectories," in *ACM SIGKDD*, 2018.

[32] C. Fan, Y. Zhang, Y. Pan, X. Li, C. Zhang, R. Yuan, D. Wu, W. Wang, J. Pei, and H. Huang, "Multi-horizon time series forecasting with temporal attention learning," in *ACM SIGKDD*, 2019.

[33] V. Kuznetsov and Z. Mariet, "Foundations of sequence-to-sequence modeling for time series," in *AISTATS*, 2019.

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

[35] S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y.-X. Wang, and X. Yan, "Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting," in *NeurIPS*, 2019.

[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, 2017.

[37] L. Zhu and N. Laptev, "Deep and confident prediction for time series at Uber," in *ICDM Workshop*. IEEE, 2017, pp. 103–110.

[38] N. Laptev, J. Yosinski, L. E. Li, and S. Smyl, "Time-series extreme event forecasting with neural networks at Uber," in *ICML*, 2017.

[39] S. Smyl and N. G. Hua, "Machine learning methods for gefcom2017 probabilistic load forecasting," *International Journal of Forecasting*, 2019.

[40] R. Wen, K. Torkkola, B. Narayanaswamy, and D. Madeka, "A multi-horizon quantile recurrent forecaster," *NeurIPS Time Series Workshop*, 2017.

[41] J. Gasthaus, K. Benidis, Y. Wang, S. S. Rangapuram, D. Salinas, V. Flunkert, and T. Januschowski, "Probabilistic forecasting with spline quantile function RNNs," in *AISTATS*, 2019.

[42] R. Wen and K. Torkkola, "Deep generative quantile-copula models for probabilistic forecasting," *ICML Time Series Workshop*, 2019.

[43] R. Koenker and K. F. Hallock, "Quantile regression," *Journal of economic perspectives*, vol. 15, no. 4, pp. 143–156, 2001.

[44] T. Gneiting, F. Balabdaoui, and A. E. Raftery, "Probabilistic forecasts, calibration and sharpness," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2007.

[45] S. S. Rangapuram, M. W. Seeger, J. Gasthaus, L. Stella, Y. Wang, and T. Januschowski, "Deep state space models for time series forecasting," in *NeurIPS*, 2018.

[46] R. Kurle, S. S. Rangapuram, E. de Bézenac, S. Günnemann, and J. Gasthaus, "Deep rao-blackwellised particle filters for time series forecasting," *NeurIPS*, 2020.

[47] D. Salinas, M. Bohlke-Schneider, L. Callot, R. Medico, and J. Gasthaus, "High-dimensional multivariate forecasting with low-rank gaussian copula processes," in *NeurIPS*, 2019.

[48] A. Koochali, A. Dengel, and S. Ahmed, "If you like it, gan it. probabilistic multivariate times series forecast with gan," *arXiv preprint arXiv:2005.01181*, 2020.

[49] E. de Bézenac, S. S. Rangapuram, K. Benidis, M. Bohlke-Schneider, R. Kurle, L. Stella, H. Hasson, P. Gallinari, and T. Januschowski, "Normalizing kalman filters for multivariate time series analysis," *NeurIPS*, 2020.

[50] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *Readings in speech recognition*, vol. 159, p. 224, 1990.

[51] Y.-S. Jeong, M. K. Jeong, and O. A. Omitaomu, "Weighted dynamic time warping for time series classification," *Pattern Recognition*, 2011.

[52] Z. Zhang, R. Tavenard, A. Bailly, X. Tang, P. Tang, and T. Corpetti, "Dynamic time warping under limited warping path length," *Information Sciences*, 2017.

[53] C. Truong, L. Oudre, and N. Vayatis, "Supervised kernel change point detection with partial annotations," in *ICASSP*. IEEE, 2019.

[54] A. Mensch and M. Blondel, "Differentiable dynamic programming for structured prediction and attention," *ICML*, 2018.

[55] A. Abid and J. Zou, "Learning a warping distance from unlabeled time series using sequence autoencoders," in *NeurIPS*, 2018.

[56] T. Vayer, L. Chapel, N. Courty, R. Flamary, Y. Soullard, and R. Tavenard, "Time series alignment with global invariances," *arXiv preprint arXiv:2002.03848*, 2020.

[57] F. Rivest and R. Kohar, "A new timing error cost function for binary time series prediction." *IEEE transactions on neural networks and learning systems*, 2019.

[58] J. Yoon, D. Jarrett, and M. van der Schaar, "Time-series generative adversarial networks," *NeurIPS*, 2019.

[59] S. Wu, X. Xiao, Q. Ding, P. Zhao, Y. Wei, and J. Huang, "Adversarial sparse transformer for time series forecasting," *NeurIPS*, 2020.
[60] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social GAN: Socially acceptable trajectories with generative adversarial networks," in *CVPR*, 2018.
[61] L. A. Thiede and P. P. Brahma, "Analyzing the variety loss in the context of probabilistic trajectory prediction," in *ICCV*, 2019.
[62] A. B. Dieng, F. J. Ruiz, D. M. Blei, and M. K. Titsias, "Prescribed generative adversarial networks," *arXiv:1910.04302*, 2019.
[63] D. Wang and Q. Liu, "Nonlinear Stein variational gradient descent for learning diversified mixture models," in *ICML*, 2019.
[64] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.
[65] Y. Yuan and K. Kitani, "Dlow: Diversifying latent flows for diverse human motion prediction," in *ECCV*, 2020.
[66] B. Gong, W.-L. Chao, K. Grauman, and F. Sha, "Diverse sequential subset selection for supervised video summarization," in *NeurIPS*, 2014.
[67] J. A. Gillenwater, A. Kulesza, E. Fox, and B. Taskar, "Expectation-maximization for learning determinantal point processes," in *NeurIPS*, 2014.
[68] M. Elfeki, C. Couprie, M. Riviere, and M. Elhoseiny, "GDPP: learning diverse generations using determinantal point process," *ICML*, 2019.
[69] M. Cuturi, J.-P. Vert, O. Birkenes, and T. Matsui, "A kernel for time series based on global alignments," in *ICASSP*. IEEE, 2007.
[70] R. Tavenard, J. Faouzi, G. Vandewiele, F. Divo, G. Androz, C. Holtz, M. Payne, R. Yurchak, M. Rußwurm, K. Kolar *et al.*, "Tslearn, a machine learning toolkit for time series data," *JMLR*, 2020.
[71] H.-F. Yu, N. Rao, and I. S. Dhillon, "Temporal regularized matrix factorization for high-dimensional time series prediction," in *NeurIPS*, 2016.
[72] M. Babaeizadeh, C. Finn, D. Erhan, R. H. Campbell, and S. Levine, "Stochastic variational video prediction," *ICLR*, 2018.
[73] J.-Y. Franceschi, E. Delasalles, M. Chen, S. Lamprier, and P. Gallinari, "Stochastic latent residual video prediction," *ICML*, 2020.
[74] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, "Curiosity-driven exploration by self-supervised prediction," in *CVPR workshop*, 2017.
[75] B. Eysenbach, A. Gupta, J. Ibarz, and S. Levine, "Diversity is all you need: Learning skills without a reward function," *ICLR*, 2019.
[76] E. Leurent, D. Efimov, and O.-A. Maillard, "Robust estimation, prediction and control with linear dynamics and generic costs," in *NeurIPS*, 2020.
[77] S. Xingjian, Z. Chen, and H. e. a. Wang, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *NeurIPS*, 2015.
[78] V. Le Guen and N. Thome, "Disentangling physical dynamics from unknown factors for unsupervised video prediction," in *CVPR*, 2020.
[79] Y. Yin, V. L. Guen, J. Dona, E. de Bézenac, I. Ayed, , N. Thome, and P. Gallinari, "Augmenting physical models with deep networks for complex dynamics forecasting," *ICLR*, 2021.

Vincent Le Guen is a Ph.D. student in deep learning and computer vision at Conservatoire National des Arts et Métiers (CNAM Paris) and EDF research lab (France). He received a M.Eng. degree from Télécom Paris and a M.Sc degree in applied mathematics from Ecole Normale Supérieure Paris Saclay in 2013. His research interests include deep learning for spatio-temporal phenomena forecasting and physics-inspired machine learning.

Nicolas Thome is a full professor at Conservatoire Nationnal des Arts et Métiers (CNAM Paris). His research interests include machine learning and deep learning for understanding low-level signals, e.g.. vision, time series, acoustics, etc. He also explores solutions for combining low-level and more higher-level data for multi-modal data processing. His current application domains are essentially targeted towards healthcare, autonomous driving and physics. He is involved in several French, European and international collaborative research projects on artificial intelligence and deep learning.

Deep Time Series Forecasting with Shape and Temporal Criteria Appendix

Vincent Le Guen, Nicolas Thome

APPENDIX A
PROOF THAT THE TEMPORAL KERNEL IS PSD

The DTW score between two time series $\mathbf{y} \in \mathbb{R}^{d\times n}$ and $\mathbf{z} \in \mathbb{R}^{d\times m}$ can be written $S(\pi) = \sum_{i=1}^{|\pi|} \mathbf{\Delta}(\mathbf{y}_{\pi_1(i)}, \mathbf{z}_{\pi_2(i)})$ where $\pi = (\pi_1, \pi_2)$ is a valid alignment between both series. Equivalently we can write the DTW score $S(\pi) = S(\mathbf{A}) = \langle \mathbf{A}, \mathbf{\Delta}(\mathbf{y}, \mathbf{z})\rangle$, where $\mathbf{A} \subset \{0,1\}^{n\times m}$ is the warping path in matrix form ($\mathbf{A}_{ij} = 1$ if $\mathbf{y}_i$ is associated to $\mathbf{z}_j$ and 0 otherwise).

Let $w : \mathcal{A}_{n,m} \longrightarrow \mathbb{R}_+^*$ be a strictly positive weighting function on alignment paths and let's consider the following kernel:

$$\begin{aligned}
\mathcal{K}_w(\mathbf{y}, \mathbf{z}) &= \sum_{\mathbf{A}\in\mathcal{A}_{n,m}} w(\mathbf{A})\ e^{-\frac{S(\mathbf{A})}{\gamma}} && (1)\\
&= \sum_{\mathbf{A}\in\mathcal{A}_{n,m}} w(\mathbf{A})\ e^{-\frac{\langle \mathbf{A}, \mathbf{\Delta}(\mathbf{y},\mathbf{z})\rangle}{\gamma}} && (2)\\
&= \sum_{\pi\in\mathcal{A}_{n,m}} w(\pi)\ e^{-\frac{\sum_{j=1}^{|\pi|} \mathbf{\Delta}\left(\mathbf{y}_{\pi_1(j)}, \mathbf{z}_{\pi_2(j)}\right)}{\gamma}} && (3)\\
&= \sum_{\pi\in\mathcal{A}_{n,m}} w(\pi) \prod_{j=1}^{|\pi|} e^{-\frac{\mathbf{\Delta}\left(\mathbf{y}_{\pi_1(j)}, \mathbf{z}_{\pi_2(j)}\right)}{\gamma}} && (4)\\
&= \sum_{\pi\in\mathcal{A}_{n,m}} w(\pi) \prod_{j=1}^{|\pi|} k(\mathbf{y}_{\pi_1(j)}, \mathbf{z}_{\pi_2(j)}) && (5)
\end{aligned}$$

where we denote $k = e^{-\frac{\mathbf{\Delta}}{\gamma}}$. We prove the following result:

Proposition 1. *If k is a PSD kernel such that $\frac{k}{1+k}$ is also PSD, the kernel $\mathcal{K}_w$ defined in Eq. 5 is also PSD.*

Proof. The proof is adapted from [7]. First, for any time series $\mathbf{y} = (\mathbf{y}_1, \dots, \mathbf{y}_n) \in \mathbb{R}^{d\times n}$ of length n and for any sequence $a \in \mathbb{N}^n$, we introduce the notation:

$$\mathbf{y}_a = (\underbrace{\mathbf{y}_1, \dots, \mathbf{y}_1}_{a_1 \text{ times}}, \dots, \underbrace{\mathbf{y}_n, \dots, \mathbf{y}_n}_{a_n \text{ times}}) \tag{6}$$

Let χ be any PSD kernel defined on $\mathbb{R}^d$ with the following condition $|\chi| < 1$, we introduce the kernel κ defined as:

$$\kappa(\mathbf{y}, \mathbf{z}) = \begin{cases} \prod_{i=1}^{|x|} \chi(\mathbf{y}_i, \mathbf{z}_j) & \text{if } |\mathbf{y}| = |\mathbf{z}| \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

Then, given a strictly positive weighting function $w(a, b) > 0$, the following kernel $\mathcal{K}_w$ defined in Eq. 8 is PSD by construction:

$$\mathcal{K}_w(\mathbf{y}, \mathbf{z}) = \sum_{a\in\mathbb{N}^n} \sum_{b\in\mathbb{N}^m} w(a, b)\ \kappa(\mathbf{y}_a, \mathbf{z}_b) \tag{8}$$

where we recall that $n = |\mathbf{y}|$ and $m = |\mathbf{z}|$. We denote $\epsilon_a = (\underbrace{1, \dots, 1}_{a_1 \text{ times}}, \dots, \underbrace{p, \dots, p}_{a_p \text{ times}})$ for any $a \in \mathbb{N}^p$. We also write for any sequences u and v of common length p: $u \otimes v = ((u_1, v_1), \dots, (u_p, v_p))$. With these notations, we can rewrite $\mathcal{K}_w$ as:

$$\mathcal{K}_w(\mathbf{y}, \mathbf{z}) = \sum_{\substack{a\in\mathbb{N}^n, b\in\mathbb{N}^m \\ \|a\|=\|b\|}} w(a, b) \prod_{i=1}^{\|a\|} \chi((\mathbf{y}, \mathbf{z})_{\epsilon_a \otimes \epsilon_b(i)}) \tag{9}$$

Notice now for each couple (a, b) there exists a unique alignment path π and an integral vector v verifying $\pi_v = \epsilon_a \otimes \epsilon_b$. Conversely, for each couple (π, v) there exists a unique pair (a, b) verifying $\pi_v = \epsilon_a \otimes \epsilon_b$. Therefore the kernel $\mathcal{K}_w$ in Eq. 9 can be written equivalently with a parameterization on (π, v) for w:

$$\mathcal{K}_w(\mathbf{y}, \mathbf{z}) = \sum_{\pi\in\mathcal{A}_{n,m}} \sum_{v\in\mathbb{N}^{|\pi|}} w(\pi, v) \prod_{j=1}^{|\pi|} \chi((\mathbf{y}, \mathbf{z})_{\pi_v(j)}) \tag{10}$$

where $\chi_{\pi(j)}$ is a shortcut for $\chi(\mathbf{y}_{\pi_1(j)}, \mathbf{z}_{\pi_2(j)})$.

Now we assume that the weighting function w depends only on π: $w(\pi, v) = w(\pi)$. Then we have:

$$\begin{aligned}
\mathcal{K}_w(\mathbf{y}, \mathbf{z}) &= \sum_{\pi\in\mathcal{A}_{n,m}} w(\pi) \sum_{v\in\mathbb{N}^{|\pi|}} \prod_{j=1}^{|\pi|} \chi_{\pi(j)}^{v_j} \\
&= \sum_{\pi\in\mathcal{A}_{n,m}} w(\pi) \prod_{j=1}^{|\pi|} \left(\chi_{\pi(j)} + \chi_{\pi(j)}^2 + \chi_{\pi(j)}^3 + \dots\right) \\
&= \sum_{\pi\in\mathcal{A}_{n,m}} w(\pi) \prod_{j=1}^{|\pi|} \frac{\chi_{\pi(j)}}{1 - \chi_{\pi(j)}}
\end{aligned}$$

By setting now $\chi = \frac{k}{1+k}$ which is PSD by hypothesis and verifies $|\chi| < 1$ (recall that $k = e^{-\frac{\mathbf{\Delta}}{\gamma}}$), we get:

$$\begin{aligned}\mathcal{K}_w(\mathbf{y},\mathbf{z}) &= \sum_{\pi\in\mathcal{A}_{n,m}} w(\pi)\prod_{j=1}^{|\pi|} k_{\pi(j)} \\ &= \sum_{\pi\in\mathcal{A}_{n,m}} w(\pi)\prod_{j=1}^{|\pi|} k(\mathbf{y}_{\pi_1(j)},\mathbf{z}_{\pi_2(j)})\end{aligned}$$

which corresponds exactly to the kernel $\mathcal{K}_w$ defined in Eq. 5. This proves that $\mathcal{K}_w$ in Eq. 5 is a well defined PSD kernel.

With the particular choice $w(\mathbf{A}) = \langle \mathbf{A}, \mathbf{\Omega_{sim}}\rangle$, we recover:

$$\begin{aligned}\mathcal{K}_w(\mathbf{y},\mathbf{z}) &= \sum_{\mathbf{A}\in\mathcal{A}} \langle \mathbf{A}, \mathbf{\Omega_{sim}}\rangle \;\; e^{-\frac{\langle \mathbf{A},\mathbf{\Delta}(\mathbf{y},\mathbf{z})\rangle}{\gamma}} \\ &= Z \times \mathrm{TDI}_\gamma^{\mathbf{\Delta},\mathbf{\Omega_{sim}}}(\mathbf{y},\mathbf{z}) \\ &= e^{-\mathrm{DTW}_\gamma^{\mathbf{\Delta}}(\mathbf{y},\mathbf{z})/\gamma} \times \mathrm{TDI}_\gamma^{\mathbf{\Delta},\mathbf{\Omega_{sim}}}(\mathbf{y},\mathbf{z}) \\ &= \mathcal{K}_{time}(\mathbf{y},\mathbf{z})\end{aligned}$$

which finally proves that $\mathcal{K}_{time}$ defined in paper Eq. 9 is a valid PSD kernel. □

The particular choice $k(u,v) = \dfrac{\frac{1}{2}e^{-\|u-v\|_2^2}}{1-\frac{1}{2}e^{-\|u-v\|_2^2}}$ fullfills Proposition 1 requirements: k is indeed PSD as the infinite limit of a sequence of PSD kernels $\sum_{i=1}^{\infty} g^i = \frac{g}{1-g} = k$, where g is a halved Gaussian PSD kernel: $g(u,v) = \frac{1}{2}e^{-\|u-v\|_2^2}$. For this choice of k, the corresponding pairwise cost matrix writes (it is the half-Gaussian cost defined in paper section 3.1.3):

$$\mathbf{\Delta}(\mathbf{y}_i,\mathbf{z}_j) = \gamma\left[\|\mathbf{y}_i-\mathbf{z}_j\|_2^2 - \log\left(2-e^{-\|\mathbf{y}_i-\mathbf{z}_j\|_2^2}\right)\right] \quad (11)$$

Appendix B
Efficient forward and backward computation

The direct computation of the shape loss $\mathrm{DTW}_\gamma^{\mathbf{\Delta}}$ (paper Eq. 3) and the temporal loss $\mathrm{TDI}_\gamma^{\mathbf{\Delta},\mathbf{\Omega_{dissim}}}$ (paper Eq. 8) is intractable, due to the exponential growth of cardinal of $\mathcal{A}_{n,m}$. We provide a careful implementation of the forward and backward passes in order to make learning efficient.

Shape loss: Regarding $\mathrm{DTW}_\gamma^{\mathbf{\Delta}}$, we rely on [6] to efficiently compute the forward pass with a variant of the Bellmann dynamic programming approach [1]. For the backward pass, we implement the recursion proposed in [6] in a custom Pytorch loss. This implementation is much more efficient than relying on vanilla auto-differentiation, since it reuses intermediate results from the forward pass.

Temporal loss: For $\mathrm{TDI}_\gamma^{\mathbf{\Delta},\mathbf{\Omega_{dissim}}}$, note that the bottleneck for the forward pass in Eq. 8 is to compute $\mathbf{A}_\gamma^* = \nabla_\Delta \mathrm{DTW}_\gamma^{\mathbf{\Delta}}(\mathbf{y},\mathbf{z})$, which we implement as explained for the $\mathrm{DTW}_\gamma^{\mathbf{\Delta}}$ backward pass. Regarding $\mathrm{TDI}_\gamma^{\mathbf{\Delta},\mathbf{\Omega_{dissim}}}$ backward pass, we need to compute the Hessian $\nabla^2\mathrm{DTW}_\gamma^{\mathbf{\Delta}}(\mathbf{y},\mathbf{z})$. We use the method proposed in [11], based on a dynamic programming implementation that we embed in a custom Pytorch loss. Again, our back-prop implementation allows a significant speed-up compared to standard auto-differentiation (see paper section 5.2.3). The resulting time complexity of both shape and temporal losses for forward and backward is $\mathcal{O}(nm)$.

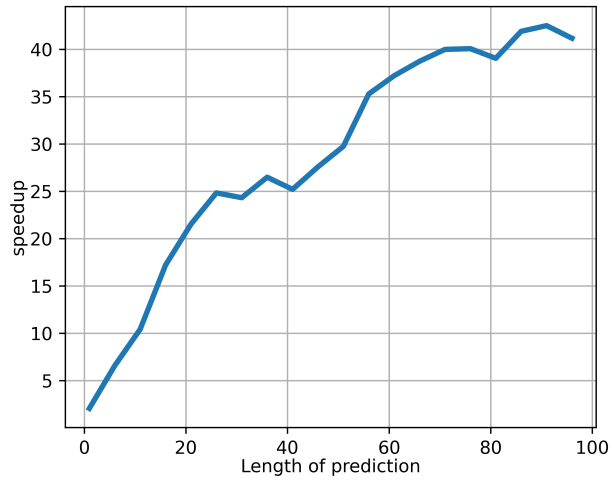


Fig. 1: Speedup of the custom forward and backward implementation of DILATE.

Custom backward implementation speedup: We compare in Figure 1 the computational time between the standard Pytorch auto-differentiation mechanism and our custom backward pass implementation. We plot the speedup of our implementation with respect to the prediction length τ (averaged over 10 random target/prediction tuples). We notice the increasing speedup with respect to τ: speedup of $\times$ 20 for 20 steps ahead and up to $\times$ 35 for 100 steps ahead predictions.

Appendix C
Datasets

We provide here a detailed description of the datasets used in the paper. We precise for each case the length of the context window T and the prediction horizon τ.

Synthetic-det ($T = 20, \tau = 20$): dataset consisting in predicting sudden changes (step functions) based on an input signal composed of two peaks. This controlled setup was designed to measure precisely the shape and time errors of predictions. We generate 500 times series for train, 500 for validation and 500 for test, with 40 time steps: the first 20 are the inputs, the last 20 are the targets to forecast. In each series, the input range is composed of 2 peaks of random temporal position i_1 and i_2 and random amplitude j_1 and j_2 between 0 and 1, and the target range is composed of a step of amplitude $j_2 - j_1$ and stochastic position $i_2 + (i_2 - i_1) + randint(-3, 3)$. All time series are corrupted by an additive Gaussian white noise of variance 0.01.

Synthetic-prob ($T = 20, \tau = 20$): this is a variant of Synthetic-det where for each input series, we generate 10 different future series of length 20 by adding noise on the step amplitude and localisation. The dataset is composed of $100 \times 10 = 1000$ time series for each train/valid/test split.

ECG5000 ($T = 84, \tau = 56$): this dataset comes from the UCR Time Series Classification Archive [4], and is composed of 5000 electrocardiograms (ECG) (500 for training, 4500 for testing) of length 140. We take the first 84 time steps (60 %) as input and predict the last 56 steps (40 %) of each time series (same setup as in [6]).

Traffic ($T = 168, \tau = 24$): this dataset is composed of road occupancy rates (between 0 and 1) from the California Department of Transportation (48 months from 2015-2016) measured every 1h. We work on the first univariate series of length 17544 (with the same 60/20/20 train/valid/test split as in [10]), and we train models to predict the 24 future points given the past 168 points (past week).

Electricity ($T = 168, \tau = 24$): this dataset consists in hourly electricity consumption measurements (kWh) from 370 customers (see Figure 2 (a)).

ETTh1 [19] ($T = 96, \tau = 96$): dataset of hourly Electricity Transformer Temperature measurements, which is an important indicator for electricity grids. This dataset enables to assess the generalization of our approach on much longer term predictions (see Figure 2 (b)).

APPENDIX D
DILATE ADDITIONAL DETAILS

D.1 External shape and temporal metrics

We detail here the two external metrics used in our experiments to evaluate the shape and temporal errors.

Ramp score: The notion of *ramping event* is a major issue for intermittent renewable energy production that needs to be anticipated for electricity grid management. For assessing the performance of trained forecasting models in presence of ramps, the Ramp Score was proposed in [16]. This score is based on a piecewise linear approximation on both input and target time series by the Swinging Door algorithm [3], [8]. The Ramp Score described in [16] is computed as the integral between the unsigned difference of derivatives of both linear approximated series. For assessing only the shape error component, we apply in our experiments the ramp score on the target and prediction series after alignment by the optimal DTW path.

Hausdorff distance: Given a set of change points $\mathcal{T}^*$ in the target signal and change points $\hat{\mathcal{T}}$ in the predicted signal, the Hausdorff distance is defined as:

$$\text{Hausdorff}(\mathcal{T}^*, \hat{\mathcal{T}}) := \max(\max_{\hat{t}\in\hat{\mathcal{T}}} \min_{t^*\in\mathcal{T}^*} |\hat{t}-t^*|, \max_{t^*\in\mathcal{T}^*} \min_{\hat{t}\in\hat{\mathcal{T}}} |\hat{t}-t^*|) \tag{12}$$

It corresponds to the greatest temporal distance between a change point and its prediction.

We now explain how the change points are computed for each dataset: for Synthetic, we know exactly by construction the positions of the change points in the target signals. For the predictions, we look for a single change point corresponding to the location of the predicted step function. We use the exact segmentation method by dynamic programming described in [15] with the Python toolbox http://ctruong.perso.math.cnrs.fr/ruptures-docs/build/html/index.html# .

For ECG5000 and Traffic datasets which present sharp peaks, this change point detection algorithm is not suited (detected change points are often located at the inflexion points of peaks and not at the exact peak location). We thus use a simple peak detection algorithm based on first order finite differences. We tune the threshold parameter for outputting a detection and the min distance between detections parameter experimentally for each dataset.

D.2 DILATE implementation details

Neural networks architectures: For the generic neural network architectures, we use a fully connected network (1 layer of 128 neurons), which does not make any assumption on data structure, and a more specialized Seq2Seq model [14] with Gated Recurrent Units (GRU) [5] with 1 layer of 128 units. Each model is trained with PyTorch for a max number of 1000 epochs with Early Stopping with the ADAM optimizer. The smoothing parameter γ of DTW and TDI is set to 10^{-2}.

DILATE hyperparameters: the hyperparameter α balancing $\mathcal{L}_{shape}$ and $\mathcal{L}_{time}$ is determined on a validation set to get comparable DTW shape performance than the $\text{DTW}_\gamma^{\mathbf{\Delta}}$ trained model: $\alpha = 0.5$ for Synthetic and ECG5000, and 0.8 for Traffic, Electricity and ETTh1. The DTW smoothing parameter γ is fixed to 10^{-2}, as discussed in paper section 5.2.3.

Our code implementing DILATE is available on line from https://github.com/vincent-leguen/DILATE.

D.3 Visualizations for Electricity and ETTh1

We provide in Figure 2 qualitative predictions of the N-Beats model [12] on the Electricity (a) and ETTh1 (b) datasets. For Electricity (a), DILATE leads to a much sharper prediction than with the MSE, as expected. For ETTH1 (b), we extend the experiments to a much longer prediction horizon ($\tau = 96$), which makes the extrapolation task much more challenging. We observe that the MSE loss only gives a future trend trajectory, while DILATE better captures the future sharp patterns.

D.4 Comparison to DILATE divergence variant

Blondel *et al.* [2] point out two limitations for using $\text{DTW}_\gamma^{\mathbf{\Delta}}$ as a loss function: first, it can take negative values and second, $\text{DTW}_\gamma^{\mathbf{\Delta}}(\mathbf{y}, \mathbf{z})$ does not reach its minimum when $\mathbf{y} = \mathbf{z}$. To address these issues, they propose a proper divergence defined as follows [2]:

$$\text{DTW-div}_\gamma^{\mathbf{\Delta}}(\mathbf{y}, \mathbf{z}) = \text{DTW}_\gamma^{\mathbf{\Delta}}(\mathbf{y}, \mathbf{z}) - \frac{1}{2}(\text{DTW}_\gamma^{\mathbf{\Delta}}(\mathbf{y}, \mathbf{y}) + \text{DTW}_\gamma^{\mathbf{\Delta}}(\mathbf{z}, \mathbf{z})) \tag{13}$$

This divergence is non-negative and satisfies $\text{DTW-div}_\gamma^{\mathbf{\Delta}}(\mathbf{y}, \mathbf{y}) = 0$. However, it is still not a distance function since the triangle inequality is not verified (as for

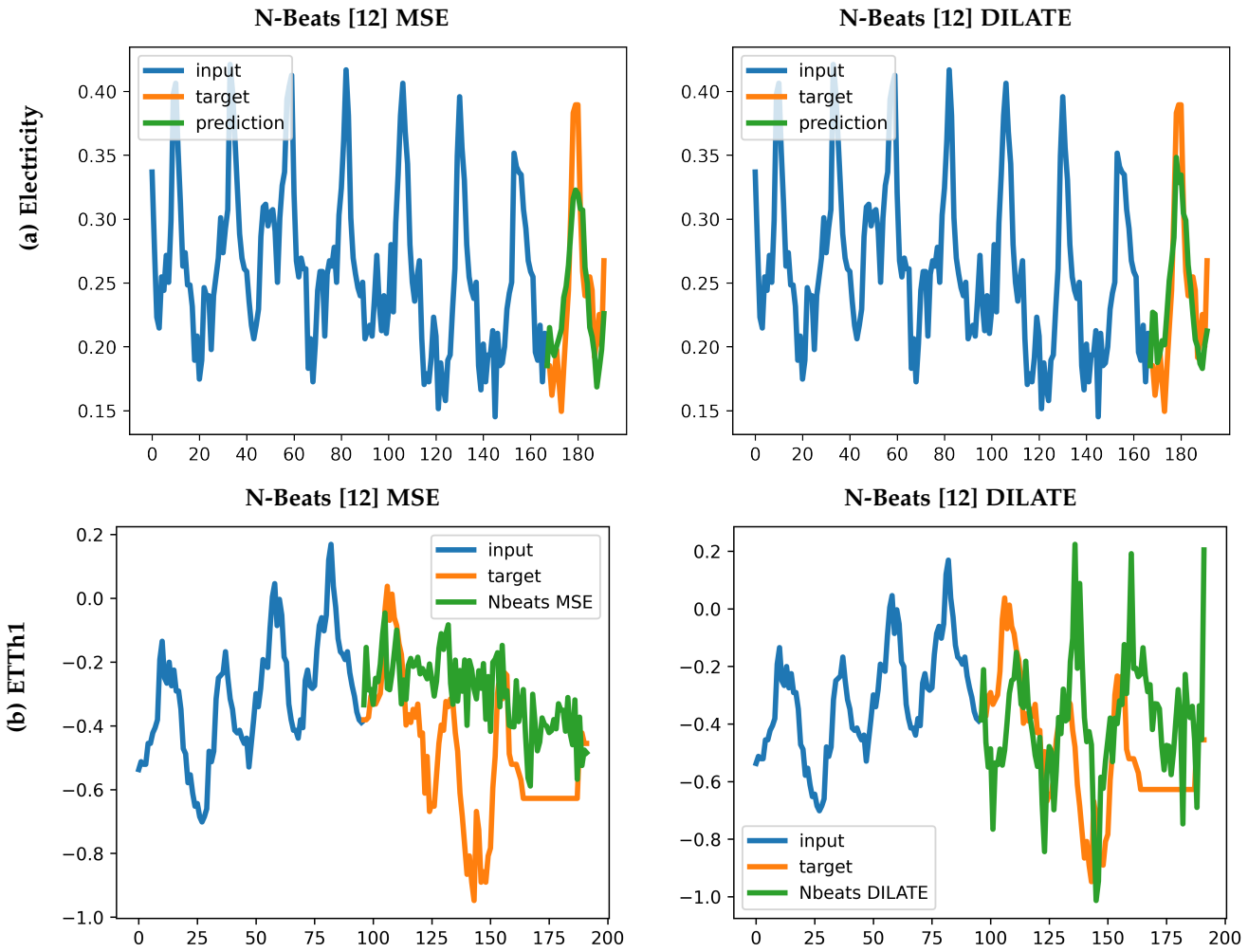


Fig. 2: Qualitative forecasting results comparing the N-Beats model [12] trained with MSE and DILATE loss on the Electricity and ETTH1 datasets.

the true DTW).

These limitations also hold for DILATE. Consequently, we use the same normalization trick to define a proper DILATE-divergence. Forecasting results in Table 1 show that DILATE-div is equivalent to DILATE with the Seq2Seq and N-Beats [12] models, and inferior to DILATE with the Informer model [19]. It confirms the good behaviour of the DILATE loss that does not require this renormalization.

TABLE 1: Comparison between DILATE and DILATE-div on the synthetic-det dataset.

Model	MSE	DILATE
Seq2Seq DILATE	**13.1** $\pm$ **1.8**	**33.7** $\pm$ **3.1**
Seq2Seq DILATE-div	**13.6** $\pm$ **0.9**	**33.6** $\pm$ **2.1**
N-Beats [12] DILATE	**13.3** $\pm$ **0.7**	**37.9** $\pm$ **1.6**
N-Beats [12] DILATE-div	**13.8** $\pm$ **0.9**	**38.5** $\pm$ **1.4**
Informer [19] DILATE	**11.8** $\pm$ **0.7**	**30.1** $\pm$ **1.3**
Informer [19] DILATE-div	12.9 $\pm$ 0.1	31.8 $\pm$ 6.5

D.5 Additional visualizations

We provide additional qualitative predictions with DILATE for the Synthetic-det in Figure 3, for ECG5000 in Figure 4 and for Traffic in Figure 5.

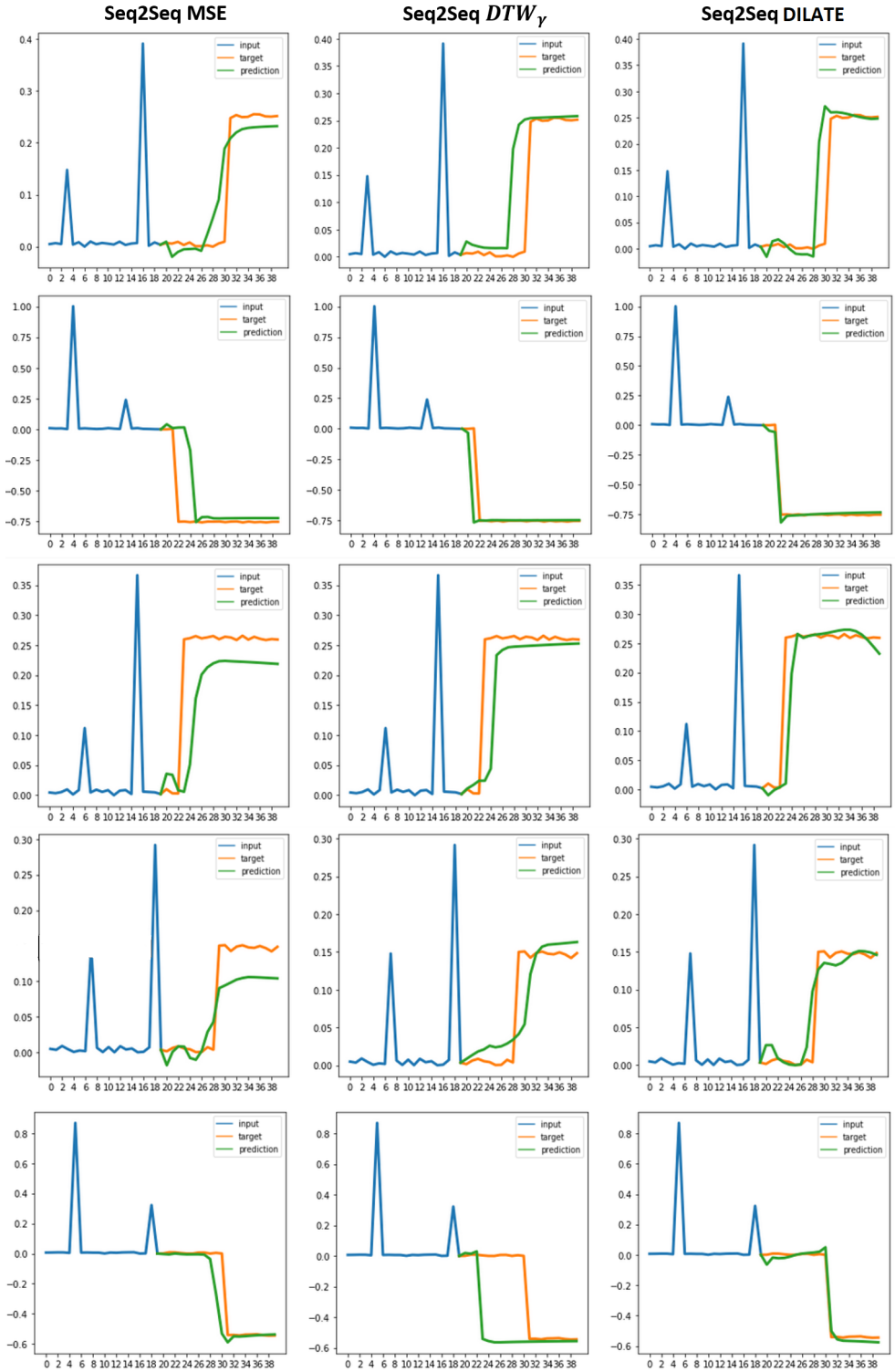


Fig. 3: Qualitative predictions for the Synthetic-det dataset.

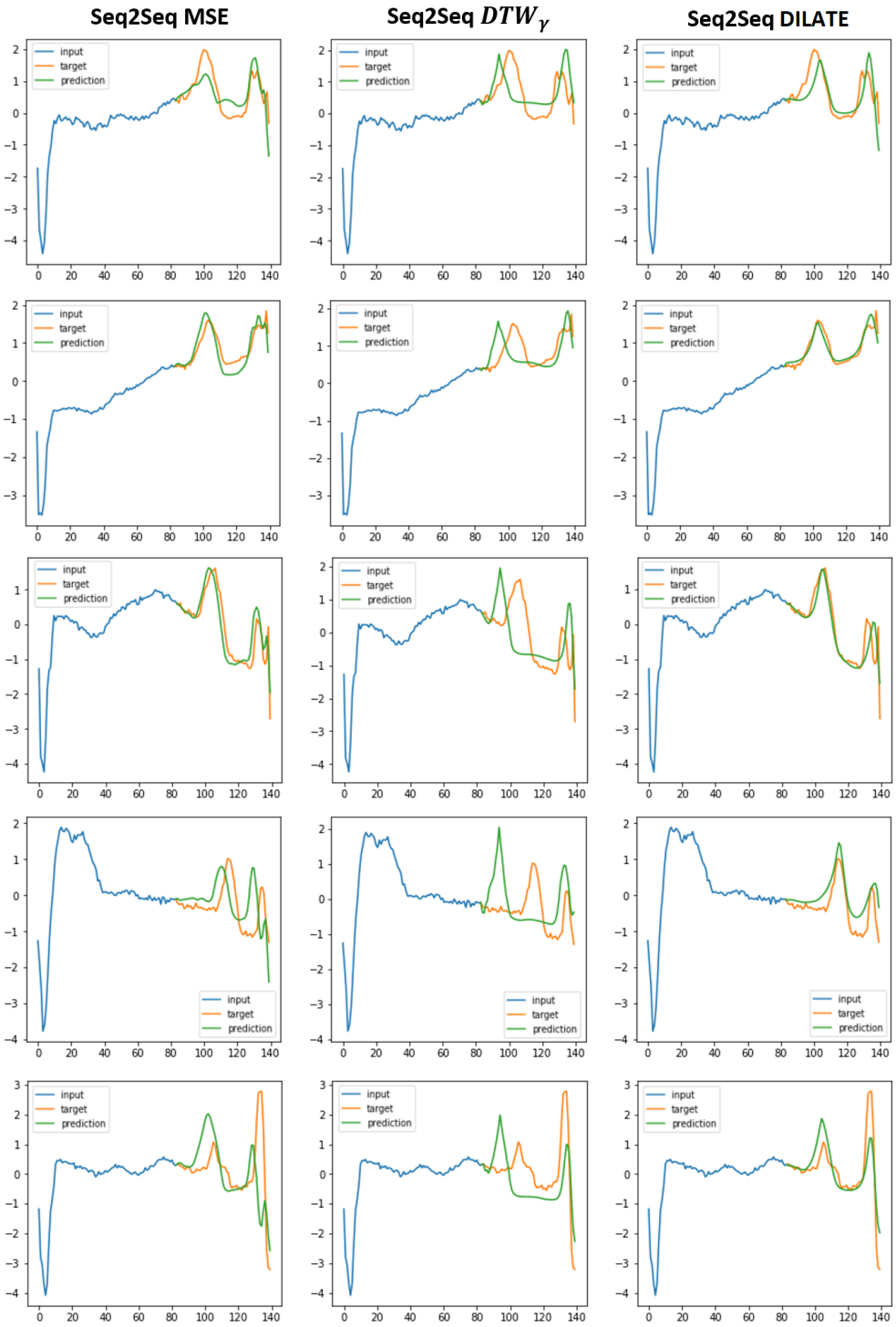


Fig. 4: Qualitative predictions for the ECG5000 dataset.

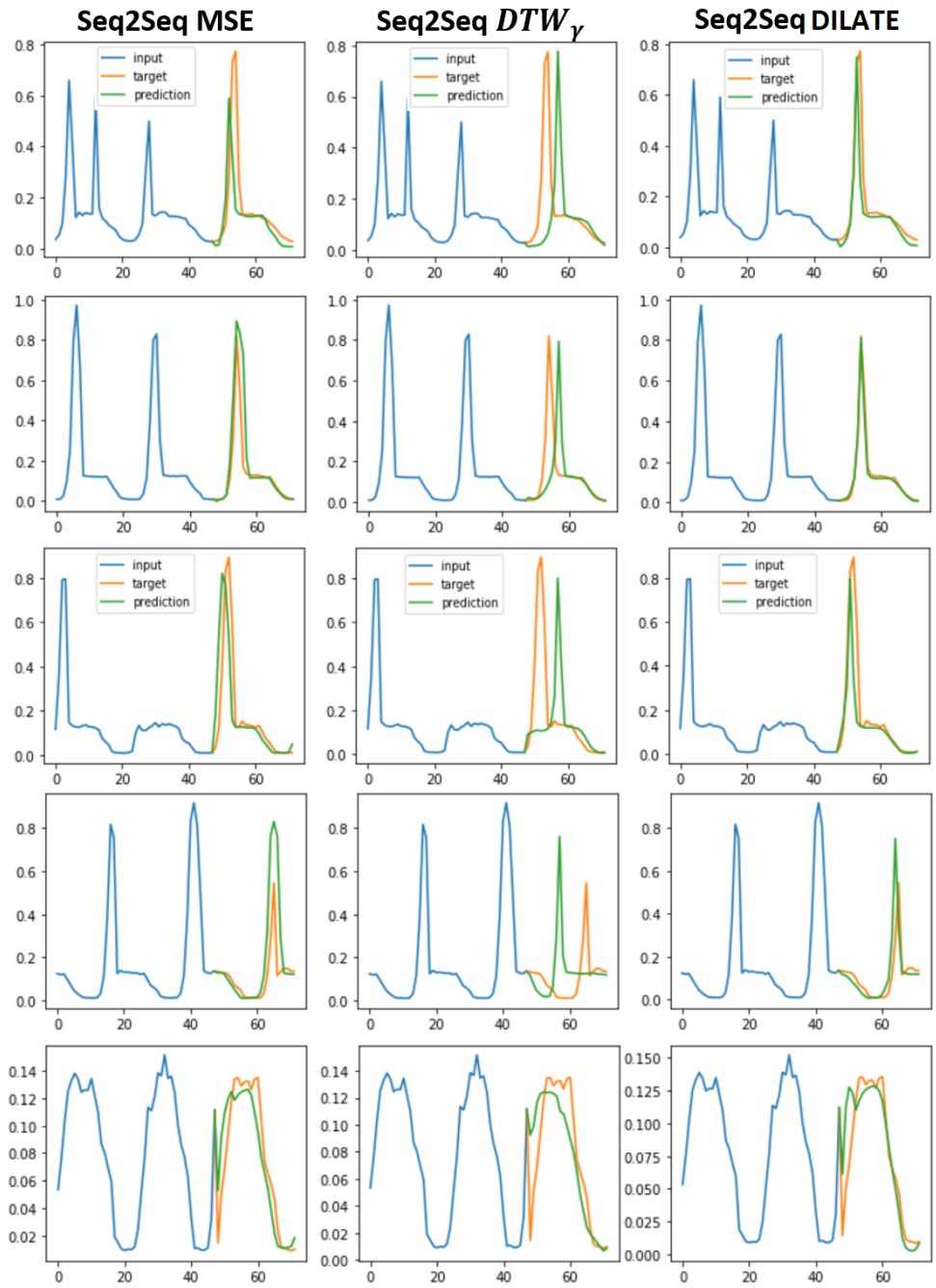


Fig. 5: Qualitative predictions for the Traffic dataset.

APPENDIX E
STRIPE++ ADDITIONAL DETAILS

E.1 Derivation of the DPP diversity loss

Determinantal Point Processes (DPPs) [9] are a probabilistic tool for describing the diversity of a ground set of items $\mathcal{S} = \{\mathbf{y}_1, ..., \mathbf{y}_N\}$. Diversity is controlled via the choice of a positive semi-definite (PSD) kernel $\mathcal{K}$ for comparing items. A DPP is a probability distribution over all subsets of $\mathcal{S}$ that assigns the following probability to a random subset $\mathbf{Y}$:

$$\mathcal{P}_{\mathbf{K}}(\mathbf{Y} = Y) = \frac{\det(\mathbf{K}_Y)}{\sum_{Y' \subseteq \mathcal{S}} \det(\mathbf{K}'_Y)} = \frac{\det(\mathbf{K}_Y)}{\det(\mathbf{K} + \mathbf{I})} \tag{14}$$

where $\mathbf{K}$ denotes the kernel in matrix form and $\mathbf{K}_A$ is its restriction to the elements indexed by $A : \mathbf{K}_A = [\mathbf{K}_{i,j}]_{i,j \in A}$.

Intuitively, a DPP encourages the selection of diverse elements from the ground set $\mathcal{Y}$. If $\mathcal{Y}$ is more diverse, a random subset $Y \sim \text{DPP}(\mathcal{K})$ sampled from the DPP will select more items, i.e. will have a larger cardinality. This idea is embedded into the diversity loss $\mathcal{L}_{diversity}$ proposed in [18]:

$$\mathcal{L}_{diversity}(\mathcal{Y}; \mathbf{K}) = -\mathbb{E}_{Y \sim \text{DPP}(\mathbf{K})}|Y| = -Trace(\mathbf{I} - (\mathbf{K} + \mathbf{I})^{-1}) \tag{15}$$

E.2 STRIPE++ implementation details

Neural network architectures: STRIPE++ is composed of a Sequence To Sequence predictive model. The encoder is a recurrent neural network (RNN) with 1 layer of 128 Gated Recurrent Units (GRU) [5] units, producing a latent state h of dimension 128. We fixed by cross-validation the dimension of each diversifying variable z_s or z_t to be $k = 8$. The decoder is another RNN with $128 + 8 + 8 = 144$ GRU units followed by fully connected layers responsible for producing the future trajectory.

The Posterior network has a similar architecture as the encoder: it is a RNN with 1 layer of 128 GRU units that takes as input the full series $(\mathbf{x}_{1:T}, \mathbf{y}^*_{T+1:T+\tau})$, followed by two multi-layer perceptrons (MLP) dedicated to output the parameters (μ^*_s, σ^*_s) and (μ^*_t, σ^*_t) of the Gaussian distribution from which to sample the posterior diversifying variables z^*_s and z^*_t.

The $\text{STRIPE}^{++}_{\text{shape}}$ and $\text{STRIPE}^{++}_{\text{time}}$ proposal mechanisms build on top of the encoder (that produces h) with a MLP with 3 layers of 512 neurons (with Batch Normalization and LeakyReLU activations) and a final linear layer to produce $N = 10$ latent codes of dimension $k = 8$ (corresponding to the proposals for z_s or z_t).

STRIPE++ hyperparameters: We cross-validated the relevant hyperparameters of STRIPE++:

- k: dimension of the diversifying latent variables z. This dimension should be chosen relatively to the hidden size of the RNN encoders and decoders (128 in our experiments). We fixed $k = 8$ in all cases.
- N: the number of future trajectories to sample. We fixed $N = 10$. We performed a sensibility analysis to this parameter in paper Figure 8.
- $\mu = 20$: quality constraint hyperparameter in the DPP kernels.

E.3 STRIPE++ additional visualizations

In paper Figure 7, we have shown qualitative prediction results of the STRIPE++ model on the `Traffic` and `Electricity` datasets. To complement these visualizations, we represent in Figure 6 for the same examples the 10th, 90th quantiles and the mean STRIPE++ prediction, computed over a set of 100 predictions. We observe that the 10th and 90th of STRIPE++ predictions offer a realistic and sharp cover of possible trajectories, that includes the ground truth future.

This contrasts with state-of-the-art deep probabilistic forecasting methods trained with the MSE or the quantile loss [17], [18]; the predictions of these models are often smooth and non-sharp, as illustrated in paper Figure 1 (b). Therefore they do not correspond to realistic scenarios.

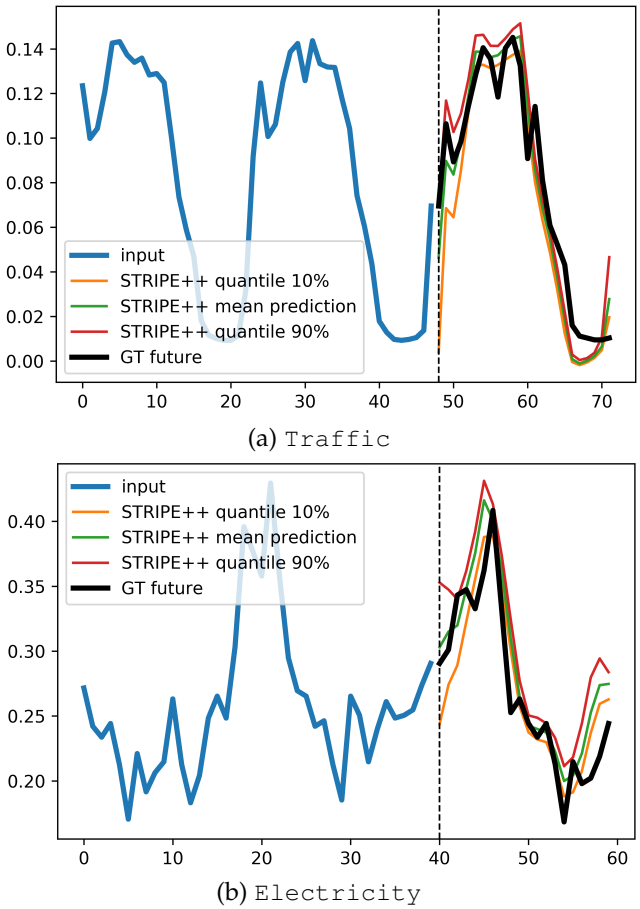


Fig. 6: STRIPE++ qualitative predictions on real-world datasets `Traffic` (a) and `Electricity` (b).

E.4 STRIPE++ quality/diversity cooperation analysis

We highlight here the importance to separate the criteria for enforcing quality and diversity. In Figure 7, we represent 50 predictions from the models Diverse DPP DILATE [18] and STRIPE++ in the plane (DTW,TDI). Diverse DPP DILATE [18] uses a DPP diversity loss based on the DILATE kernel, which is the same than for quality. We clearly see that the two objectives conflict: this model increases the DILATE diversity (by increasing the variance in the shape (DTW) or the time TDI) components) but a lot of these predictions have a high DILATE loss (worse quality). In contrast, STRIPE++ predictions are diverse in DTW and TDI, and maintain an overall low DILATE loss. STRIPE++ succeeds in recovering a set of good tradeoffs between shape and time leading a low DILATE loss.

We also display in Figure 8 the unnormalized results of the comparison between DeepAR [13] and STRIPE++ with the number of sampled trajectories N. The quality is measured with $-\text{H}_{quality}(\text{DILATE})$ and the diversity with $-\text{H}_{diversity}(\text{DILATE})$ (higher is better in both cases). These results confirm the conclusions from paper Figure **??**. The diversity increases for deepAR and STRIPE++ with N; however, this gain in diversity comes at the cost of a loss of quality for deepAR, in contrast to STRIPE++ which does not deteriorate quality.

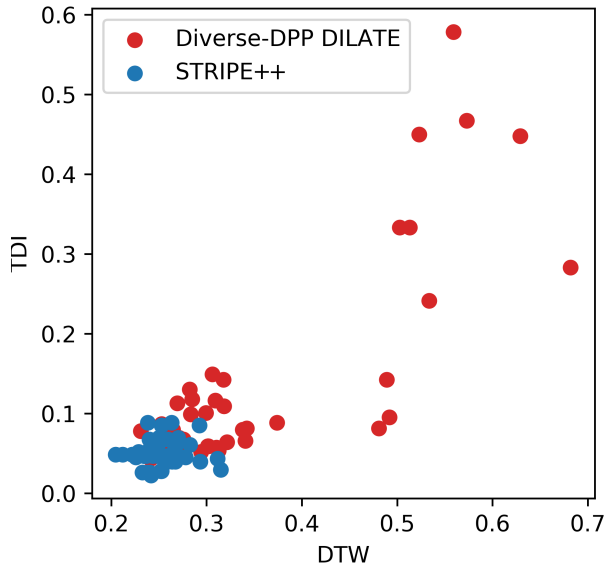


Fig. 7: Scatterplot of 50 predictions in the plane (DTW,TDI), comparing STRIPE++ v.s. Diverse DPP DILATE [18].

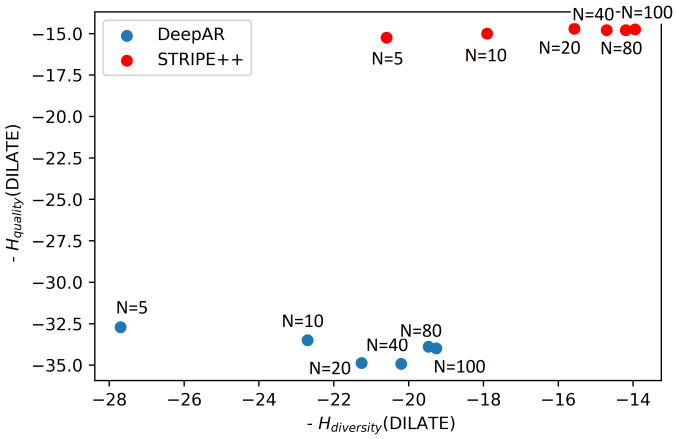


Fig. 8: Influence of the number N of trajectories on quality ($-\mathrm{H}_{quality}$(DILATE), higher is better) and diversity ($-\mathrm{H}_{diversity}$(DILATE), higher is better) for the `synthetic-prob` dataset.

References

[1] Richard Bellman. On the theory of dynamic programming. *Proceedings of the National Academy of Sciences of the United States of America*, 38(8):716, 1952.

[2] Mathieu Blondel, Arthur Mensch, and Jean-Philippe Vert. Differentiable divergences between time series. *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021.

[3] EH Bristol. Swinging door trending: Adaptive trend recording? In *ISA National Conf. Proc., 1990*, pages 749–754, 1990.

[4] Yanping Chen, Eamonn Keogh, Bing Hu, Nurjahan Begum, Anthony Bagnall, Abdullah Mueen, and Gustavo Batista. The UCR time series classification archive. 2015.

[5] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[6] Marco Cuturi and Mathieu Blondel. Soft-DTW: a differentiable loss function for time-series. In *ICML*, 2017.

[7] Marco Cuturi, Jean-Philippe Vert, Oystein Birkenes, and Tomoko Matsui. A kernel for time series based on global alignments. In *ICASSP*. IEEE, 2007.

[8] Anthony Florita, Bri-Mathias Hodge, and Kirsten Orwig. Identifying wind and solar ramping events. In *IEEE GreenTech Conference*, 2013.

[9] Alex Kulesza, Ben Taskar, et al. Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning*, 5(2–3):123–286, 2012.

[10] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *ACM SIGIR*, 2018.

[11] Arthur Mensch and Mathieu Blondel. Differentiable dynamic programming for structured prediction and attention. *ICML*, 2018.

[12] Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. *ICLR*, 2020.

[13] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191, 2020.

[14] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *NeurIPS*, 2014.

[15] Charles Truong, Laurent Oudre, and Nicolas Vayatis. A review of change point detection methods. *arXiv preprint arXiv:1801.00718*, 2018.

[16] Loïc Vallance, Bruno Charbonnier, Nicolas Paul, Stéphanie Dubost, and Philippe Blanc. Towards a standardized procedure to assess solar forecast accuracy: A new ramp and time alignment metric. *Solar Energy*, 2017.

[17] Ruofeng Wen, Kari Torkkola, Balakrishnan Narayanaswamy, and Dhruv Madeka. A multi-horizon quantile recurrent forecaster. *NeurIPS Time Series Workshop*, 2017.

[18] Ye Yuan and Kris Kitani. Diverse trajectory forecasting with determinantal point processes. *ICLR*, 2020.

[19] 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. *AAAI*, 2021.