Robust Probabilistic Time Series Forecasting

TaeHo Yoon[†*] **Youngsuk Park**[‡] **Ernest K. Ryu**[†] **Yuyang Wang**[‡]
[†]Department of Mathematical Sciences, Seoul National University
[‡]AWS AI Labs, Amazon Research

Abstract

Probabilistic time series forecasting has played critical role in decision-making processes due to its capability to quantify uncertainties. Deep forecasting models, however, could be prone to input perturbations, and the notion of such perturbations, together with that of robustness, has not even been completely established in the regime of probabilistic forecasting. In this work, we propose a framework for robust probabilistic time series forecasting. First, we generalize the concept of adversarial input perturbations, based on which we formulate the concept of robustness in terms of bounded Wasserstein deviation. Then we extend the randomized smoothing technique to attain robust probabilistic forecasters with theoretical robustness certificates against certain classes of adversarial perturbations. Lastly, extensive experiments demonstrate that our methods are empirically effective in enhancing the forecast quality under additive adversarial attacks and forecast consistency under supplement of noisy observations. The code for our experiments is available at `https://github.com/tetrzim/robust-probabilistic-forecasting`.

1 INTRODUCTION

Time series forecasting is among the most important tasks in the automation and optimization of business processes. In retail, for example, determining how many units of each item to purchase and where to store them depends on forecasts of future demand over different regions. In cloud computing, the estimated future usage of services and infrastructure components guides capacity planning (Park et al., 2019, 2020). The real-time forecasting is essential as a subroutine for vehicle control and planning (Kim et al., 2020) and other numerous applications (Petropoulos et al., 2022). Due to its crucial role in downstream decision making, there are two desirable properties of a forecaster: 1) the ability to generate probabilistic forecasts that allows for uncertainty estimation; 2) reliability, in the sense of being robust to (potentially adversarial) input perturbations. In the present work, we investigate robust probabilistic forecasting models which aim to satisfy the both requirements.

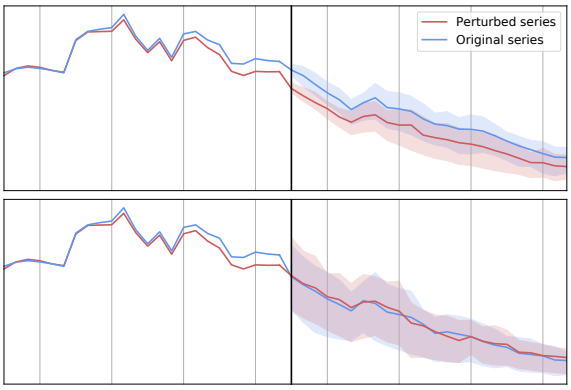


Figure 1: Predictions from vanilla DeepAR model (top) and its version with randomized smoothing we propose (bottom). Forecasts are separated from input series via the black vertical line. The smoothed model is more robust in the sense that its probabilistic outputs are less affected by adversarial input perturbation (which perturbs the blue series to red).

In the classical time series literature where statistical methods were predominant, studies on robust forecasting were mainly focused on model stability against outliers (Connor et al., 1994; Gelper et al., 2010). More

Proceedings of the 25[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151.

[*]Work done as an intern at Amazon Research. Correspondence to: TaeHo Yoon <tetrzim@snu.ac.kr>, Youngsuk Park <pyoungsu@amazon.com>.

recently, deep learning models increasingly gained popularity and has gradually become the main workhorse of scalable forecasting (Li et al., 2019b; Oreshkin et al., 2019; Sen et al., 2019; Fan et al., 2019; Chen et al., 2020; Arik et al., 2020; Zhou et al., 2021). A distinct concept of adversarial robustness has emerged as an issue because deep neural networks are notoriously prone to small adversarial input perturbations (Szegedy et al., 2013; Goodfellow et al., 2015). Particularly in the context of forecasting, in Dang-Nhu et al. (2020), the authors showed that deep autoregressive time series forecasting models with probabilistic outputs could suffer from such attacks.

Despite its importance, to the best of our knowledge, there is no prior work which has developed the formal concept of adversarial robustness for probabilistic forecasting models. This leaves the forecasters under the threat of adversarial attacks, endangering the decision making of mission-critical tasks. Furthermore, the time series data possess a unique characteristic (i.e. the time dimension), which allows for robustness notions that are fundamentally different from l_p-adversarial robustness, e.g., forecasts' stability under temporal window shift (translation) or the presence of statistical outliers. Therefore, it is necessary to establish a notion of robustness for probabilistic time series forecasting models that is general enough to encompass diverse classes of input changes and the corresponding practical requirements.

Contributions. In this paper, we develop a framework of robust probabilistic time series forecasting, handling both theory and practice. To this aim, we first generalize the concept of adversarial perturbations in terms of abstract input and output transformations and provide a formal mathematical notion of robustness in terms of distributional stability of outputs when inputs are perturbed. With these foundations established, we propose randomized smoothing for probabilistic forecasters, which enjoys theoretical robustness guarantees for distinct classes of input perturbations and potentially improves a baseline model's robustness without requiring separate learning procedures. To establish even more pragmatic robust methods, we combine the smoothing techniques with randomized training, i.e., augmentation of training data with random noises, which is generally known to be effective in generating robust base learners. Finally, we empirically verify that the randomizing procedures are indeed effective in rendering robust probabilistic forecasting models through extensive experiments on multiple real datasets.

1.1 Related Work

Classically robust forecasting via statistical methods. Earlier ideas on robust forecasting have mostly focused on adapting the classical techniques to deal with outliers, missing data, or change points. A plethora of works have been developed in multiple directions, including robust versions of exponential and Holt–Winters smoothing (Cipra, 1992; Cipra et al., 1995; Gelper et al., 2010), outlier-robust Kalman filters (Cipra and Romera, 1997; Ting et al., 2007; Agamennoni et al., 2011; Chang, 2014), and statistical frameworks based on detection or filtering of anomalies (Connor et al., 1994; Garnett et al., 2009; Ristanoski et al., 2013; Anava et al., 2015; Guo et al., 2016).

Deep learning for time series forecasting. Neural network has been applied to time series forecasting for more than half a century, and the earliest work dates back to 1960s (Hu and Root, 1964). Despite the early start, neural networks found little success in the time series forecasting literature until recently. With the explosive production of time series data and advances in neural architectures, deep learning has become increasingly popular. The strong performance of the deep forecasting models is especially prominent in the fields where a large collection of time series is available, such as demand forecasting in large retailers. Among the deep forecasting models, more relevant to the present work are the approaches that generate probabilistic forecasts. This is typically achieved by two avenues. The first approach, including Salinas et al. (2020, 2019); de Bézenac et al. (2020); Rangapuram et al. (2018); Wang et al. (2019), uses neural networks as backbone sequential model and the last layer is customized via a likelihood function. An alternative approach directly generates the desired quantile forecasts; see for example Wen et al. (2017); Lim et al. (2021); Park et al. (2021); Eisenach et al. (2022). Other classes of works include variance reduced training (Lu et al., 2021) or domain adaptaton based techniques (Jin et al., 2022). For comprehensive study of deep forecasting models, we refer interested readers to Benidis et al. (2020); Hewamalage et al. (2021); Alexandrov et al. (2020).

Adversarial attacks and time series. The seminal work of Szegedy et al. (2013) demonstrated that image classification models based on deep neural networks, in spite of their high test accuracy, tend to be susceptible to hardly human-perceptible changes, which cause them to completely misclassify the inputs. This inspired a number of works to further study effective attack schemes (Goodfellow et al., 2015; Madry et al., 2018; Papernot et al., 2017; Athalye et al., 2018). In the time series domain, earlier works (Fawaz et al., 2019; Karim et al., 2021) mainly focused on attacking time series classification models, and attack against probabilistic forecasting models was first devised by Dang-Nhu et al. (2020) using reparametrization tricks.

Certified adversarial defenses. While the adversarial training (Kurakin et al., 2017; Madry et al., 2018) has been arguably the most successful defense scheme against adversarial attacks on the empirical side, its theoretical performance guarantee over perturbed data has not been established. Towards developing defense scheme with certificates, a series of works (Dvijotham et al., 2018; Wong and Kolter, 2018; Wong et al., 2018; Raghunathan et al., 2018b,a; Gowal et al., 2018) suggested to directly control the local Lipschitz constant of feedforward neural networks, but such approaches were constrained to moderate-sized models. Another line of works (Lecuyer et al., 2019; Cohen et al., 2019; Salman et al., 2019; Li et al., 2019a) studied randomized smoothing as a more scalable and model-agnostic approach, and successfully provided practical accuracy guarantees on classification problems up to the ImageNet scale under adversarial perturbations. Randomized smoothing have also had applications in regression problems in the context of certifiably robust object detection (Chiang et al., 2020). However, we are not aware of any prior works along this direction which have considered adversarial defenses for models with probabilistic outputs, which is the standard framework for time series forecasting.

Exposure bias and translation robustness. An autoregressive (or conditional) sequence generation model may behave significantly differently in training and inference stages because in the test time, it generates outputs based on its own previous outputs, whose distribution may deviate from the ground-truth and the resulting error can be propagated (Bengio et al., 2015; Bowman et al., 2016; Ranzato et al., 2016). This phenomenon, often referred to as exposure bias, has been studied and empirically addressed by a number of prior works on language models (Norouzi et al., 2016; Schmidt, 2019) and more recently on time series forecasting (Sangiorgio and Dercole, 2020). These works are partially related to our translation robustness framework where we control propagation of errors caused by appending noisy or adversarial observations, but our approach is fundamentally distinct in that we focus on worst-case perturbations (rather than regarding the data distributions) and theoretically guaranteed solutions (rather than empirical remedies).

2 PRELIMINARIES

2.1 Probabilistic Time Series Forecasting

Suppose we are given a dataset of N time series, where the i-th time series consist of observation $x_{i,t} \in \mathbb{R}$ with (optional) input covariates $z_{i,t} \in \mathbb{R}^d$ at time t. We drop the time series index whenever the context is clear. Examples of the input covariates include price and promotion at a certain time with the observations being the sales. For each time series, we observe T past targets $\boldsymbol{x} = x_{1:T} \in \mathcal{X} = \bigcup_{T=1}^{\infty} \mathbb{R}^T$ and all covariates $z_{1:T+\tau} \in \mathcal{Z}$ to predict τ future targets $x_{T+1:T+\tau} \in \mathcal{Y} = \mathbb{R}^\tau$. We denote a global[1] probabilistic forecaster $f : \mathcal{X} \times \mathcal{Z} \to \mathcal{P}(\mathcal{Y})$ where $\mathcal{P}(\mathcal{Y})$ denotes the probability distribution on the prediction space $\mathcal{Y}$. With a slight abuse of notation, we describe the forecaster as

$$(Y_1, \dots, Y_\tau) = f(x_1, \dots, x_T, z_1 \dots, z_{T+\tau}),$$

where $(Y_1, \dots, Y_\tau)$ are random variables associated with future targets $(x_{T+1}, \dots, x_{T+\tau})$ and their full distributions are specified in terms of quantiles (Wen et al., 2017; Park et al., 2021) or parametric forms (Salinas et al., 2020), e.g., Gaussian, Student's t, or negative binomial. For notational simplicity, we omit the covariates $z_{1:T+\tau}$ and concisely write

$$\mathbf{Y} = (Y_1, \dots, Y_\tau) = f(\boldsymbol{x}). \tag{1}$$

We respectively denote as $x_{T+1}, \dots, x_{T+\tau}$ the ground-truth future targets and $\boldsymbol{y} = (y_1, \dots, y_\tau) = (\hat{x}_{T+1}, \dots, \hat{x}_{T+\tau})$ the sampled prediction (or realization) of future targets from probabilistic forecaster f. Since probabilistic forecast $f(\boldsymbol{x})$ essentially carries distributional information, we abuse notation to allow sampling from it: $(\hat{x}_{T+1}, \dots, \hat{x}_{T+\tau}) \sim f(\boldsymbol{x})$.

2.2 Adversarial Attacks on Probabilistic Autoregressive Forecasting Models

In the probabilistic forecasting setting, the adversarial perturbation (or attack) $\boldsymbol{\delta}$ on the input $\boldsymbol{x}$ with given adversarial target values $\mathbf{t}_{\text{adv}} \in \mathbb{R}^m$ and a statistic[2] $\chi : \mathbb{R}^\tau \to \mathbb{R}^m$ can be found by minimizing

$$\underset{\boldsymbol{\delta}:\|\boldsymbol{\delta}\| \leqslant \eta}{\operatorname{argmin}} \left\| \mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}+\boldsymbol{\delta})}[\chi(Y_1, \dots, Y_\tau)] - \mathbf{t}_{\text{adv}} \right\|_2^2 \tag{2}$$

where $\eta \geqslant 0$ is the attack threshold, and the norm $\|\boldsymbol{\delta}\|$ is chosen depending on the context. The expectation $\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}+\boldsymbol{\delta})}$ is taken over the randomness in $(Y_1, \dots, Y_\tau) = f(\boldsymbol{x} + \boldsymbol{\delta})$, the output of the probabilistic forecaster (1) on the input $\boldsymbol{x} + \boldsymbol{\delta}$. The target value $\mathbf{t}_{\text{adv}}$ is chosen to be significantly different from $\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x})}[\chi(Y_1, \dots, Y_\tau)]$. For the case of stock price predictions, the choice of χ may be varied to express financial quantities such as buy- or sell-option prices; see Dang-Nhu et al. (2020) for details.

For practical experiments, we focus on attacking subsets of prediction outputs. In other words, we consider

[1] Forecast models are the same across all N time series.

[2] Dang-Nhu et al. (2020) limits χ and $\mathbf{t}_{\text{adv}}$ to scalar ones ($m = 1$).

statistics of the form

$$\chi_H(Y_1, \dots, Y_\tau) = (Y_{h_1}, \dots, Y_{h_m}) \tag{3}$$

in (2), where H is a subset of prediction indices with size m, i.e., $H := \{h_1, \dots, h_m\} \subset \{1, \dots, \tau\}$. In this case, the adversary searches for a minimal norm perturbation $\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{\delta}$ for which the subset of perturbed forecasts is significantly different from the original forecasts at corresponding indices, and potentially from the ground-truth values as well.

Constrained optimization (2) can be relaxed into a regularized optimization problem as follows:

$$\min_{\boldsymbol{\delta}} L(\boldsymbol{\delta}) := \|\boldsymbol{\delta}\|^2 + \lambda \cdot \left\|\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}+\boldsymbol{\delta})}[Y_H] - \boldsymbol{t}_{\text{adv}}\right\|_2^2 \tag{4}$$

where $\lambda > 0$ is a hyperparameter. The derivative of the relaxed objective (4) with respect to $\boldsymbol{\delta}$ could be computed via the reparametrization trick[3] as in Dang-Nhu et al. (2020), which allows us to solve the regularized problem using any first-order optimizer.

3 DEFINING ROBUSTNESS FOR PROBABILISTIC TIME SERIES FORECASTING

The adversarial attack in time series forecasting has been proposed only in terms of additive input perturbation with respect to l_p norm as in (2). However, the unique properties of time series data including the existence of time dimension, periodicity, or seasonality potentially allow for a number of distinct types of perturbation. In this section, we generalize the notion of adversarial input perturbations in probabilistic forecasting, in order to incorporate distinct classes of input changes. Then we define the corresponding notion of robustness which properly quantifies model sensitivity to those input perturbations.

3.1 Generalized Input Perturbations in Time Series Forecasting

We consider abstract input perturbation $T_{\mathcal{X}} \colon \mathcal{X} \to \mathcal{X}$ and output transformation $T_{\mathcal{Y}} \colon \mathcal{Y} \to \mathcal{Y}$. Given a probabilistic forecaster f and an input $\boldsymbol{x} \in \mathcal{X}$, we describe the forecast output from f on $\boldsymbol{x}$ under the input perturbation $T_{\mathcal{X}}$ as $f(T_{\mathcal{X}}(\boldsymbol{x}))$, and the original forecast output with the output transformation $T_{\mathcal{Y}}$ applied as $(T_{\mathcal{Y}})_{\#} f(\boldsymbol{x})$. Ultimately, we want to have

$$f \circ T_{\mathcal{X}} \approx T_{\mathcal{Y}} \circ f \tag{5}$$

[3]The forecaster should support sampling from a distribution with location and/or scale parameters, which is the case for autoregressive forecasting models.

in probabilistic sense, toward achieving robust forecasters. Before formally providing the detailed concept of robustness (in Section 3.2), we first walk through two example classes of perturbations: additive adversarial attack and time shift with new noisy observations, and demonstrate how the transformations $T_{\mathcal{X}}$ and $T_{\mathcal{Y}}$ can be specified.

3.1.1 Additive Adversarial Perturbation

Consider the additive advsersarial perturbation which deceives the forecaster to deviate from its original forecasts on the subset H of prediction indices. We model the corresponding $T_{\mathcal{X}}$ as

$$T_{\mathcal{X}}(\boldsymbol{x}) = \boldsymbol{x} + \boldsymbol{\delta}^\star(\boldsymbol{x}), \tag{6}$$

where

$$\boldsymbol{\delta}^\star(\boldsymbol{x}) = \underset{\|\boldsymbol{\delta}\| \leqslant \eta}{\operatorname{argmax}} \left\|\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}+\boldsymbol{\delta})}[\mathbf{Y}_H] - \mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x})}[\mathbf{Y}_H]\right\|^2. \tag{7}$$

Simply taking the output transformation as the identity map, i.e., $T_{\mathcal{Y}} = \text{Id}$, the requirement $f \circ T_{\mathcal{X}} \approx T_{\mathcal{Y}} \circ f$ reduces to

$$f(\boldsymbol{x} + \boldsymbol{\delta}^\star(\boldsymbol{x})) \approx f(\boldsymbol{x}).$$

That is, we want our forecaster to be insensitive to adversarially constructed additive noise. If f is both robust in this sense and also has good prediction performance over the clean (unattacked) test dataset, its forecast quality will be retained even when it is given with adversarially perturbed time series data.

3.1.2 Time Shift with New Noisy Observations

Consider the scenario where one initially has an input series $\boldsymbol{x} = (x_1, \dots, x_T)$, and later a set of $k \ll \tau$ new observations $\{\tilde{x}_{T+1}, \dots, \tilde{x}_{T+k}\}$ arrives and the entire prediction task is shifted by k time steps. Suppose that we want the initial forecasts

$$f(\boldsymbol{x}) = (Y_1, Y_2, \dots, Y_{k+1}, Y_{k+2}, \dots)$$

to be consistent with the forecasts

$$f(\boldsymbol{x}; \tilde{x}_{T+1}, \dots, \tilde{x}_{T+k}) = (Y'_{k+1}, Y'_{k+2}, \dots)$$

produced after given the new observations. In this case, we let $T_{\mathcal{X}}(\boldsymbol{x}) = (\boldsymbol{x}; \tilde{x}_{T+1}, \dots, \tilde{x}_{T+k})$ be the augmentation by new observations, and

$$T_{\mathcal{Y}}(y_1, y_2, \dots, y_{k+1}, y_{k+2}, \dots) = (y_{k+1}, y_{k+2}, \dots)$$

be the left translation operator by k time steps. Then $f \circ T_{\mathcal{X}} \approx T_{\mathcal{Y}} \circ f$ requires that

$$Y_{k+1} \approx Y'_{k+1},\ Y_{k+2} \approx Y'_{k+2},\ \dots,$$

i.e., the two sets of forecasts should be consistent.

Note that $\tilde{x}$'s are inherited in $T_{\mathcal{X}}$, and when some of $\tilde{x}$'s are highly anomalous or have been manipulated by an adversary, $T_{\mathcal{X}}$ may largely impact the original forecasts. In our experiments, we consider the specific case of appending a single adversarial observation

$$\tilde{x}_{T+1} := (1+\rho)x_{T+1}, \tag{8}$$

which is (de-)amplified relative to the ground truth according to the adversarial parameter $\rho > -1$.

3.2 Formal Mathematical Definition of Robustness

Given a probabilistic forecaster f and a transformation pair $T_{\mathcal{X}}, T_{\mathcal{Y}}$ suitable for time series setting, both $f(T_{\mathcal{X}}(\boldsymbol{x}))$ and $(T_{\mathcal{Y}})_{\#} f(\boldsymbol{x})$ are random variables in $\mathbb{R}^{\tau}$ whose distributions are specified. Let us denote, for each prediction time point $t = 1, \dots, \tau$, the associated marginal distributions by μ_t and μ_t' respectively, i.e., $(f(T_{\mathcal{X}}(\boldsymbol{x})))_t \sim \mu_t$ and $((T_{\mathcal{Y}})_{\#} f(\boldsymbol{x}))_t \sim \mu_t'$. We aim to formally quantify the informal notion (5) in terms of certain metric between these distributions.

As a final ingredient for establishing the precise definition, we define $d(\boldsymbol{x}; T_{\mathcal{X}})$, which is a measure of how significant the change due to the transformation $T_{\mathcal{X}}$ is, or in other words, a dissimilarity measure between $\boldsymbol{x}$ and $T_{\mathcal{X}}(\boldsymbol{x})$. For the case of additive adversarial perturbation in Section 3.1.1, a natural choice for d would be $d(\boldsymbol{x}; T_{\mathcal{X}}) = \|\boldsymbol{\delta}^\star(\boldsymbol{x})\|$. For the time-shift setup in Section 3.1.2, we take d of the form

$$d(\boldsymbol{x}; T_{\mathcal{X}}) = D\left(\tilde{\boldsymbol{x}}_{T+1:T+k}; \hat{\boldsymbol{x}}_{T+1:T+k}\right)$$

where $\hat{x}_{T+1:T+k}$ are the model's initial point forecasts on the first k future time points, and $D \geqslant 0$ satisfies $D(\boldsymbol{y}'; \boldsymbol{y}) = 0$ iff $\boldsymbol{y}' = \boldsymbol{y}$. We take $D(\boldsymbol{y}'; \boldsymbol{y}) = \|\boldsymbol{y}' - \boldsymbol{y}\|_2$ in this paper for the sake of establishing theoretical guarantees, but D can also be chosen in ways that involve likelihood to less penalize large deviations, e.g., $D(\boldsymbol{y}'; \boldsymbol{y}) = -\log\left(q(\boldsymbol{y}')/q(\boldsymbol{y})\right)$, where q is the joint density function for the model's k-step ahead predictions.

Definition 1. Let f be a probabilistic forecaster, $\boldsymbol{x} \in \mathcal{X}$, and $T_{\mathcal{X}}, T_{\mathcal{Y}}$ the input, output transformations with marginal distributions μ_t, μ_t', respectively. Then, f is ε–η robust at $\boldsymbol{x}$ with respect to the transformation pair $(T_{\mathcal{X}}, T_{\mathcal{Y}})$ if, provided that $d(\boldsymbol{x}; T_{\mathcal{X}}) < \eta$, for any $t = 1, \dots, \tau$, we have

$$W_1\left(\mu_t, \mu_t'\right) < \varepsilon. \tag{9}$$

Connection to adversarial attacks. In the additive adversarial attack we detailed in Section 2.2, it is assumed that χ is chosen by the adversary. Therefore, a defense scheme against the attack (2) would naturally involve minimizing an objective of the form

$$\sup_{\|\boldsymbol{\delta}\| \leqslant \eta} \sup_{\chi \in \mathcal{F}} \left(\mathbb{E}[\chi(f(\boldsymbol{x}+\boldsymbol{\delta}))] - \mathbb{E}[\chi(f(\boldsymbol{x}))]\right), \tag{10}$$

where $\mathcal{F}$ denotes the collection of χ's which the adversary could choose from. Note that if $\|\cdot\| = \|\cdot\|_2$, and if $\mathcal{F}$ consists of L-Lipschitz continuous functions for some $L > 0$, the inner maximization in (10) results in a constant multiple of 1-Wasserstein (or W_1) distance between the distributions of $f(\boldsymbol{x}+\boldsymbol{\delta})$ and $f(\boldsymbol{x})$, due to Kantorovich-Rubinstein duality. This interpretation motivates our choice of 1-Wasserstein distance as the measure of local distributional change in the robustness definition (9).

Connection to quantile forecasts. Here we provide another, more general perspective on the reason for formulating Definition 1 in terms of W_1 distance. Probabilistic forecasts are often characterized via quantiles; MQ-RNN (Wen et al., 2017) directly performs quantile regression, and for sampling-based forecasters such as DeepAR (Salinas et al., 2020), sample quantiles are used to compute the prediction intervals. This practice of using quantiles as important quantities is fortuitously aligned with the following interpretation of W_1 distance as the average quantile difference; if F, G are respectively the cumulative distribution function of a real-valued random variable and μ, ν are the corresponding probability distributions, then

$$W_1(\mu, \nu) = \int_0^1 |F^{-1}(u) - G^{-1}(u)|\, du. \tag{11}$$

That is, our robustness definition requires that a model's quantile estimates, in the average sense over probability levels, should not be significantly affected by small input perturbations with respect to $d_{\mathcal{X}}$.

4 THEORY AND FRAMEWORKS

In this section, we develop methodologies for robust forecasting based on randomized smoothing, covering the two classes of adversarial perturbations we considered in Section 3.2. We establish and discuss the theoretical robustness guarantees of these smoothing-based methods. Additionally, we revisit the randomized training (data augmentation with noises) widely adopted by practitioners as a strategy for enhancing base forecasters for randomized smoothing techniques.

4.1 Randomized Smoothing for Guaranteed Robustness Against l_2 Perturbations

We first develop a robustness framework using randomized smoothing for additive adversarial perturbations (covered in Section 3.1.1) with $d(\boldsymbol{x}; T_{\mathcal{X}}) = \|\boldsymbol{\delta}\|_2$.

Algorithm 1: Randomized smoothing for probabilistic forecasters

Input: Multi-horizon sample-based forecaster f, Input series $\boldsymbol{x} = (x_1, \ldots, x_T)$, τ, n, σ^2
Output: n sample paths $\hat{\boldsymbol{x}}^{(j)}_{T+1:T+\tau}$ from g_σ $(j = 1, \ldots, n)$
for $j = 1, \ldots, n$ **do**
  $\zeta_1, \ldots, \zeta_T \sim \mathcal{N}(0, \sigma^2)$ i.i.d.
  $\tilde{\boldsymbol{x}} \leftarrow (x_1 + \zeta_1, \ldots, x_T + \zeta_T)$
  $\hat{\boldsymbol{x}}^{(j)}_{T+1:T+\tau} \sim f(\tilde{\boldsymbol{x}})$
end

One-step ahead predictors. Consider the simple setting where f is a random function from $\mathbb{R}^T$ to $\mathbb{R}$, which we will extend to multivariate cases later. Let us denote by $F_{\boldsymbol{x}}$ the cumulative distribution function (cdf) of the random variable $f(\boldsymbol{x})$, i.e.,

$$F_{\boldsymbol{x}}(r) := \Pr\left[f(\boldsymbol{x}) \leqslant r\right]$$

for $r \in \mathbb{R}$. Given the smoothing paramter $\sigma > 0$, we define the smoothed version g_σ of f as the random function from $\mathbb{R}^T$ to $\mathbb{R}$ with cdf

$$\begin{aligned} G_{\boldsymbol{x},\sigma}(r) &= \Pr\left[g_\sigma(\boldsymbol{x}) \leqslant r\right] \qquad (12)\\ &= \mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathcal{N}(0,\sigma^2 I)}\left[F_{\boldsymbol{x}+\boldsymbol{z}}(r)\right] = \int F_{\boldsymbol{x}+\boldsymbol{z}}(r) p_\sigma(\boldsymbol{z})\, d\boldsymbol{z}, \end{aligned}$$

where $p_\sigma(\boldsymbol{z})$ is the density function of the multivariate Gaussian distribution $\mathcal{N}(0, \sigma^2 I)$. Below, we provide a theoretical guarantee on the robustness of the smoothed probabilistic predictor g_σ, analogous to prior results (Cohen et al., 2019; Salman et al., 2019; Chiang et al., 2020) for deterministic setups.

Theorem 1. *Let f be a random function from $\mathbb{R}^T$ to $\mathbb{R}$, and let g_σ be as in* (12)*. Given $\boldsymbol{x} \in \mathbb{R}^T$, we have the inequality*

$$\begin{aligned} \mathrm{Ro}(\boldsymbol{x};\sigma) &:= \limsup_{\|\boldsymbol{\delta}\|_2 \to 0} \frac{W_1(G_{\boldsymbol{x},\sigma}, G_{\boldsymbol{x}+\boldsymbol{\delta},\sigma})}{\|\boldsymbol{\delta}\|_2}\\ &\leqslant \frac{1}{\sigma}\int_{-\infty}^{\infty} \phi\left(\Phi^{-1}(G_{\boldsymbol{x},\sigma}(r))\right) dr \end{aligned} \qquad (13)$$

provided that the integral on the right hand side is locally bounded at $\boldsymbol{x}$, where ϕ, Φ denote the pdf and cdf of the standard normal distribution.

Note that if $\mathrm{Ro}(\boldsymbol{x};\sigma) < \infty$, then f is $O(\eta)$–η robust for η small enough, in the sense of Definition 1 with respect to adversarial perturbations, in the one-step ahead prediction case.

Finiteness of $\mathrm{Ro}(\boldsymbol{x};\sigma)$. Provided that the cdf $G_{\boldsymbol{x},\sigma}(r)$ of the smoothed random variable $g_\sigma(\boldsymbol{x})$ has nonzero derivative at all $r \in \mathbb{R}$, we can make the change of variable $u = G_{\boldsymbol{x},\sigma}(r)$ to rewrite (13) as

$$\mathrm{Ro}(\boldsymbol{x};\sigma) \leqslant \frac{1}{\sigma}\int_0^1 \phi(\Phi^{-1}(u)) \left(G^{-1}_{\boldsymbol{x},\sigma}\right)'(u)\, du.$$

Note that the quantity $\left(G^{-1}_{\boldsymbol{x},\sigma}\right)'(u)$ is the quantile density function of $g_\sigma(\boldsymbol{x})$, while $\phi\left(\Phi^{-1}(u)\right)$ is the inverse quantile density of the standard Gaussian. Therefore, intuitively speaking, we expect $\mathrm{Ro}(\boldsymbol{x};\sigma)$ to be finite unless the quantiles of $g_\sigma(\boldsymbol{x})$ blow up too quickly compared to those of the normal distribution as $u \to 0, 1$. Indeed, the following Lemmas show that $\mathrm{Ro}(\boldsymbol{x};\sigma) < \infty$ holds under mild, practical assumptions on f.

Lemma 2. *Let f and g_σ be as in Theorem 1. Suppose that there exists a function $\varphi : \mathbb{R}_+ \to \mathbb{R}_+$ such that $\Pr\left[|f(\boldsymbol{x})| \geqslant r\right] \leqslant \varphi(r)$ for any $\boldsymbol{x} \in \mathbb{R}^T$ and $r > 0$, and*

$$\int_0^\infty \varphi(r)\, dr < \infty. \qquad (14)$$

Then there exists $C > 0$, depending only on φ and σ, such that $\mathrm{Ro}(\boldsymbol{x};\sigma) < C$ for all $\boldsymbol{x} \in \mathbb{R}^T$.

Lemma 3. *Suppose that f is parametrized with globally bounded mean and variance. Then f satisfies the assumptions of Lemma 2.*

In particular, the most commonly used distributions including Gaussian, generalized Student's t ($\nu > 2$), or negative binomial are all allowed if the mean and scaling parameters are bounded.

Multi-horizon predictors. In the case when f is a random function from $\mathbb{R}^T$ to $\mathbb{R}^\tau$ as in the multi-horizon forecasting setup, we can apply Theorem 1 and Lemmas 2, 3 to each component function of f. This directly implies that f, under suitable assumption, is $O(\eta)$–η robust with respect to the l_2 adversarial perturbation in the sense of Definition 1, which we formally restate in the following as Corollary 4. Therefore, we take the (component-wisely) smoothed version g_σ of the baseline predictor f as our predictor with robustness guarantees, which is presented in Algorithm 1. Note that we implement g_σ by directly sampling its output paths, which is done by sampling independently over randomness in $\boldsymbol{z}$ and f.

Corollary 4. *Let f be a probabilistic multi-horizon forecaster, denoted $f(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_\tau(\boldsymbol{x}))$. Suppose that $\varphi : \mathbb{R}_+ \to \mathbb{R}_+$ satisfies $\Pr\left[|f_j(\boldsymbol{x})| \geqslant r\right] \leqslant \varphi(r)$ for any $\boldsymbol{x} \in \mathbb{R}^T$, $r > 0$ and $j = 1, \ldots, \tau$, and* (14) *holds. Then there exists a constant $C > 0$ depending only on φ and σ such that for any $\eta > 0$, the smoothed forecaster g_σ defined by applying the smoothing* (12) *to each component $f_j(\boldsymbol{x})$ is $C\eta$–η robust at all $\boldsymbol{x}$ with respect to $T_{\mathcal{X}}$ as in* (6) *with $\|\cdot\| = \|\cdot\|_2$, $T_{\mathcal{Y}} = \mathrm{Id}$ and $d(\boldsymbol{x}; T_{\mathcal{X}}) = \|\boldsymbol{\delta}^\star(\boldsymbol{x})\|$.*

Algorithm 2: Future smoothing for probabilistic autoregressive forecasters

Input: Probabilistic autoregressive forecaster f, Input series $\boldsymbol{x} = (x_1, \dots, x_T)$, τ, n, σ^2, (Optionally) Noisy observations $(y_1, \dots, y_k) = (\tilde{x}_{T+1}, \dots, \tilde{x}_{T+k})$

Output: Sample forecasts $\hat{x}_{T+i}^{(j)}$ $(i = k+1, \dots, k+\tau,\ j = 1, \dots, n)$

for $i = 1, \dots, k+\tau$ **do**
  if $i \leqslant k$ **then**
    $y_{i-1} \leftarrow \tilde{x}_{T+i-1}$
  else
    $y_{i-1} \leftarrow \hat{x}_{T+i-1}$
    for $j = 1, \dots, n$ **do**
      $\zeta_1, \dots, \zeta_{i-1} \leftarrow \mathcal{N}(0, \sigma^2)$ i.i.d.
      $\hat{x}_{T+i}^{(j)} \leftarrow f\left(\boldsymbol{x}; y_1 + \zeta_1, \dots, y_{i-1} + \zeta_{i-1}\right)$
    end
    $\hat{x}_{T+i} \leftarrow \frac{1}{n} \sum_{j=1}^{n} \hat{x}_{T+i}^{(j)}$
  end
end

Possible generalization of the theorems. We point out that the domain of f could be replaced with some representation space $\mathcal{W}$ without altering the essence of the theorems. If $\Psi : \mathcal{X} \to \mathcal{W}$ is an invertible representation map such that $d_{\mathcal{X}}(\boldsymbol{x}; T_{\mathcal{X}}) = \|\Psi(T_{\mathcal{X}}(\boldsymbol{x})) - \Psi(\boldsymbol{x})\|_2$, then one can simply apply Theorem 1 and Corollary 4 to $f \circ \Psi^{-1}$, which is a random function from $\mathcal{W}$ to $\mathbb{R}$. Thus, albeit the theorems seem to only provide guarantees for local l_2 perturbations, one can extend the results to fundamentally distinct types of perturbations by choosing an appropriate $\mathcal{W}$, e.g., the frequency domain.

4.2 Noisy Autoregressive Inference for Robustness Under Time Shift

In this section, we propose a strategy, Algorithm 2, for achieving robustness against the time-shift setup of Section 3.1.2. As randomized smoothing buffers the effect of input perturbation by averaging over noised inputs, we exploit random noises to buffer the effect of appending the noisy observation $\tilde{x}_{T+1}$. Note that because the uncertainty is now within future times, we perform smoothing over future time indices.

An autoregressive forecaster f can be generally described in the form $f(\boldsymbol{x}) = (Y_1, Y_2, \dots)$, where $\boldsymbol{x} \in \mathbb{R}^T$ is an input series and

$$Y_h = f^{(h)}\left(\boldsymbol{x}; Y_1, \dots, Y_{h-1}\right) \tag{15}$$

for some random functions $f^{(h)}$ from $\mathbb{R}^{T+h-1}$ to $\mathbb{R}$, for $h = 1, 2, \dots$. For fixed $\sigma > 0$ and $\boldsymbol{x} \in \mathbb{R}^T$, consider the smoothed version $g_\sigma^{(h)}$ of $f^{(h)}$, defined as

$$g_\sigma^{(h)}(\boldsymbol{x}; y_1, \dots, y_{h-1}) = \mathop{\mathbb{E}}_{\boldsymbol{\zeta} \sim \mathcal{N}(0, \sigma^2 I_{h-1})} \left[f^{(h)}(\boldsymbol{x}; y_1 + \zeta_1, \cdots, y_{h-1} + \zeta_{h-1}) \right]$$

where we noise only the variables $y_1, \dots, y_{h-1}$ but not $\boldsymbol{x}$. Now we define g_σ, the smoothed version of f, by $g_\sigma(\boldsymbol{x}) = (Y_1, Y_2, \dots)$, where

$$Y_h = g_\sigma^{(h)}\left(\boldsymbol{x}; y_1, \dots, y_{h-1}\right) \tag{16}$$

for $h = 1, 2, \dots$, and $y_j \in \mathbb{R}$ $(j = 1, \dots, h-1)$ denotes either 1) the mean forecast from g_σ at that time point (i.e. expected value of Y_j) if the true value is unobserved, or 2) the given ground-truth value if a new observation at that time has arrived. Algorithm 2 presents the procedural implementation of g_σ based on sampling. The theoretical justification for performing smoothing in this way is Corollary 5, which states the robustness property of the smoothed forecaster g_σ with respect to the notions provided in Sections 3.1.2 and 3.2, which is done by applying the smoothing framework developed in Section 4.1 under a slightly different setting.

Corollary 5. *Let f be an autoregressive forecaster defined as* (15) *and suppose that $\varphi : \mathbb{R}_+ \to \mathbb{R}_+$ satisfies* (14) *and*

$$\Pr\left[\left| f^{(h)}(\boldsymbol{x}; y_1, \dots, y_{h-1}) \right| \geqslant r \right] \leqslant \varphi(r)$$

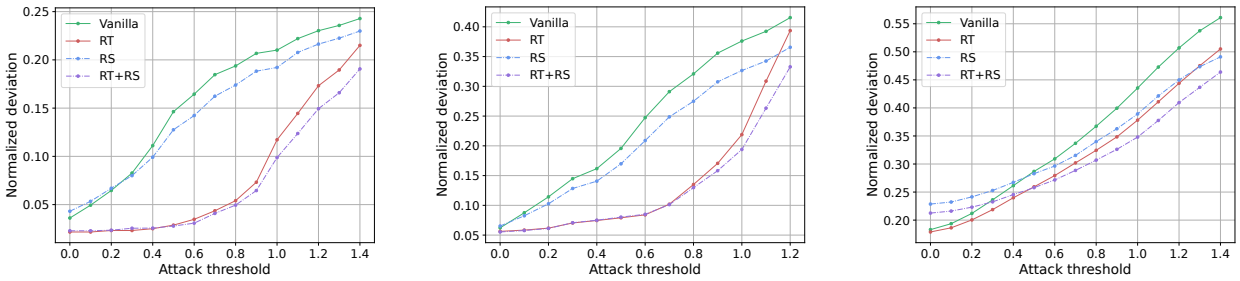
holds for each $h = 1, 2, \dots$. Then, for all $\eta > 0$, the forecaster g_σ defined as (16) *is $O(\eta)$–η robust at all $\boldsymbol{x} \in \mathbb{R}^T$ with respect to $T_{\mathcal{X}}, T_{\mathcal{Y}}$ of Section 3.1.2 and*

$$d(\boldsymbol{x}; T_{\mathcal{X}}) = \left\| \tilde{\boldsymbol{x}}_{T+1:T+k} - \hat{\boldsymbol{x}}_{T+1:T+k} \right\|_2,$$

where $\tilde{\boldsymbol{x}}_{T+1:T+k}$ are the values appended by $T_{\mathcal{X}}$ and $\hat{\boldsymbol{x}}_{T+1:T+k}$ are the k-step mean predictions from the initial forecasts $g_\sigma(\boldsymbol{x})$.

4.3 Training the Baseline Model via Random Data Augmentation

Having a baseline model f with solid prediction performance over noised inputs is the key for achieving success in randomized smoothing, which is expected to improve the theoretical certificate (13) as well as practical performance (Cohen et al., 2019). This encourages us to additionally apply *randomized training*, which augments the training dataset with random noises to adapt the baseline forecaster to random perturbations. Note that randomized training and smoothing are different schemes; in particular, the former requires training from scratch while the latter does not. In the end, we take advantage of both techniques to achieve effective robust forecasters.



(a) Exchange Rate, $\tau = 30$, $\{H\} = \{\tau\}$ (b) M4-Daily, $\tau = 14$, $\{H\} = \{\tau\}$ (c) Traffic, $\tau = 24$, $\{H\} = \{\tau\}$

Figure 2: ND_H from DeepAR models on different datasets. Randomized smoothing (RS) of baseline models uses $\sigma = 0.5$. Randomized training (RT) uses $\sigma_{\mathrm{tr}} = 0.1$.

5 EXPERIMENTS

In this section, we demonstrate the effects of our frameworks for the DeepAR (Salinas et al., 2020) implementation within GluonTS (Alexandrov et al., 2020) on real datasets, including the M4-Daily, Exchange Rate, Traffic and UCI Electricity datasets preprocessed as in Salinas et al. (2019). We use DeepAR as it is the standard model with properties of being sampling-based and autoregressive, which Algorithms 1 and 2 respectively require.

We specifically focus on how point forecasts from the model change under input transformations of Sections 3.1.1 and 3.1.2, and quantitatively assess them using the normalized deviation (ND)

$$\mathrm{ND}_H = \frac{\sum_{k=1}^{N} \sum_{h \in H} |\hat{x}_{k,T+h} - x^{\mathrm{ref}}_{k,T+h}|}{\sum_{k=1}^{N} \sum_{h \in H} |x^{\mathrm{ref}}_{k,T+h}|}. \tag{17}$$

Here $H \subset \{1, \dots, \tau\}$ is the set of prediction indices of interest, N is the size of the test dataset, $\hat{x}_{k,T+h}$ is the model's prediction for $(T+h)$-th time step for the kth (possibly transformed) test series, and $x^{\mathrm{ref}}_{k,T+h}$ is the corresponding reference value (which may either be the ground-truth value or the model output before transformation, depending on the setup).

5.1 Prediction Performance Under Additive Adversarial Attacks

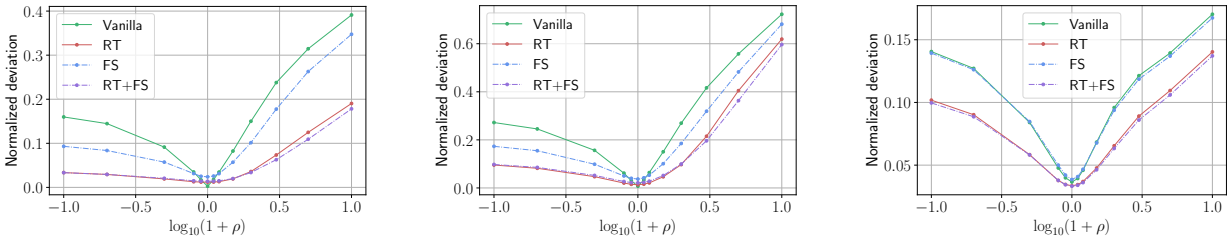
In this section, we consider the setup of Section 3.1.1, where the input transformation corresponds to the adversarial attack of Dang-Nhu et al. (2020).

Experimental setup. Following Dang-Nhu et al. (2020), we choose the relative l_2 norm

$$\|\boldsymbol{\delta}\|_{\boldsymbol{x}} := \left(\sum_{i=1}^{t} (\delta_i / x_i)^2 \right)^{1/2}$$

as the measure of perturbation magnitude. Given the set $H \subset \{1, \dots, \tau\}$ of attack indices and attack threshold $\eta > 0$, we solve the problem (4) with different choices of $\mathbf{t}_{\mathrm{adv}}$ and λ, and among the resulting approximate solutions $\boldsymbol{\delta}$ meeting the norm constraint $\|\boldsymbol{\delta}\|_{\boldsymbol{x}} \leqslant \eta$, we measure the largest point forecast error in terms of normalized deviation (17). Here we take the ground-truth future values as reference values, i.e., $x^{\mathrm{ref}}_{k,T+h} = x_{k,T+h}$. As the attack is performed under norm constraint with respect to $\|\cdot\|_{\boldsymbol{x}}$, we accordingly use *relative* noises in both randomized smoothing and randomized training. That is, if the given input is $\boldsymbol{x} = (x_1, \dots, x_T)$ and the randomizing variance is σ^2, then we use $(x_1(1+\zeta_1), \dots, x_t(1+\zeta_t))$ as noised inputs, where $\zeta_1, \dots, \zeta_t$ are i.i.d. samples from $\mathcal{N}(0, \sigma^2)$. Note that in this case, our theoretical results can be applied with locally scaled version of f with respect to each input $\boldsymbol{x}$. To distinguish the two randomizing procedures, we respectively denote by σ^2 and σ^2_{tr} the variance values of noises used in smoothing and data augmentation for training.

Results and discussion. Figure 2 respectively compares the performance of vanilla DeepAR models (trained without data augmentation; solid green line) with their corresponding smoothed versions (labeled RS; dashed blue line), and the random-trained models (labeled RT; solid red line) with their smoothed versions (labeled RS+RT; dashed purple line). We observe three important points. First, for the majority of cases, randomized smoothing provides statistically significant improvement to vanilla models' performance under attacks with moderate to high threshold values η (see Table 2 in the appendix), at the cost of possibly slightly worsening the performance for small η. Second, RT models tend to strongly outperform the corresponding vanilla models, uniformly over the levels of attack threshold η. Third, further smoothing the RT model (RS+RT) can improve upon RT and in these cases, RS+RT achieves the best empirical performance



(a) Exchange Rate (b) M4-daily (c) Traffic

Figure 3: $\mathrm{ND}_{\mathrm{rel}}$ from DeepAR models on different datasets. Randomized training (RT) uses $\sigma_{\mathrm{tr}} = 0.1$. Future smoothing (FS) uses $\sigma = 1.0$ for the first two datasets, and $\sigma = 0.1$ for the Traffic dataset.

against attacks with high values of η. Table 2, provided in the appendix, displays full experiment results for all datasets and distinct attack indices.

We emphasize that only the smoothed models are the ones that come with robustness certificates, and in particular, we find that RS+RT is a promising methodology, supported from both theoretical and empirical sides. Additionally, randomized smoothing is readily applicable to any pre-trained baseline model as a post-processing step without the cost of model retraining, and still offers a potential direction for improving the model's robustness. We believe that these points constitute important practical values of the smoothing technique in general.

Randomized training and prediction performance. As an aside, we unexpectedly find that RT tends to improve the usual prediction performance of forecasting models (see Table 1 in the appendix). That is, randomizing the training data may positively impact a model's generalization in the time series domain, which we believe, is an interesting phenomenon in its own right. In the appendix, we provide further discussion on this point, connecting the observation to prior works on generalization and training with noise.

5.2 Forecast Consistency Under Time Shift with Noisy Observation

In this section, we examine the setup of Section 3.1.2 with $k = 1$, where we append an adversarial observation to input series.

Experimental setup. Given a series $\boldsymbol{x} \in \mathbb{R}^T$, we append $\tilde{x}_{T+1}$ as in (8) with various values of ρ within the range $-1 \leqslant \log_{10}(1+\rho) \leqslant 1$. As a metric, we compute the ND (17) with $x^{\mathrm{ref}}_{k,T+h} = \hat{x}_{k,T+h}$ as reference values, for $h \in H = \{2, \dots, \tau\}$. This measures the relative discrepancy between the forecasts $(\hat{x}_{T+1}, \dots, \hat{x}_{T+\tau})$ and $(\hat{x}'_{T+2}, \dots, \hat{x}'_{T+\tau+1})$, respectively based on $\boldsymbol{x}$ and $(\boldsymbol{x}; \tilde{x}_{T+1})$, at common indices. The lower this value is, the more consistent the forecasts are, before and after the arrival of adversarial observation $\tilde{x}_{T+1}$. We scale each input series $\boldsymbol{x}$ before applying noises; i.e., $\boldsymbol{x}$ is replaced by $\boldsymbol{x}/S_{\boldsymbol{x}}$ for some $S_{\boldsymbol{x}} > 0$ computed by the model to process each series within a consistent scale.

Results and discussion. Figure 3 compares the metrics from vanilla DeepAR model, its smoothed version using Algorithm 2 (labeled FS), the random-trained (RT) model as in the previous section, and its smoothed version (RT+FS), on each dataset. The horizontal axes represent the adversarial parameter in a logarithmic scale $\log_{10}(1+\rho)$, and the vertical axes represent the relative ND. The vanilla models already have a desirable behavior around $\rho = 0$, but as $|\rho|$ grows, their forecast consistency is progressively broken. On the other hand, RT models tend to be more resilient compared to vanilla models. FS provides statistically significant improvement in forecast consistency to vanilla and RT models for large values of ρ in many cases (see Table 3 in the appendix). We elicit a message similar to that of Section 5.1; RT+FS is theoretically well-supported, and often achieves superior empirical performance as well. We provide Table 3 containing all experiment results in the appendix.

6 CONCLUSION

In this paper, we study robustness in the context of probabilistic time series forecasting. We provide a framework of robust forecasting based on randomized smoothing with theoretical certificates, and display empirical effectiveness of the randomizing framework against two distinct types of input perturbations.

The formal treatment of robustness for probabilistic forecasting models is still only at its beginning. We anticipate that the topic of robust probabilistic forecasting allows for multiple interesting and promising directions of future work, including establishing tighter theoretical guarantees or more extensive empirical study with broader classes of transformations.

Acknowledgements

TY and EKR were supported by the National Research Foundation of Korea (NRF) Grant funded by the Korean Government (MSIP) [No. 2017R1A5A1015626] and by the Samsung Science and Technology Foundation (Project Number SSTF-BA2101-02). The authors would like to thank Nghia Hoang, Hilaf Hasson, Danielle Robinson, and Anoop Deoras in Amazon research for their fruitful feedback.

References

Gabriel Agamennoni, Juan I Nieto, and Eduardo M Nebot. An outlier-robust Kalman filter. *IEEE International Conference on Robotics and Automation*, 2011.

Alexander Alexandrov, Konstantinos Benidis, Michael Bohlke-Schneider, Valentin Flunkert, Jan Gasthaus, Tim Januschowski, Danielle C Maddix, Syama Sundar Rangapuram, David Salinas, Jasper Schulz, et al. GluonTS: Probabilistic and neural time series modeling in Python. *Journal of Machine Learning Research*, 21(116):1–6, 2020.

Oren Anava, Elad Hazan, and Assaf Zeevi. Online time series prediction with missing data. *International Conference on Machine Learning*, 2015.

Sercan O Arik, Chun-Liang Li, Jinsung Yoon, Rajarishi Sinha, Arkady Epshteyn, Long T Le, Vikas Menon, Shashank Singh, Leyou Zhang, Nate Yoder, et al. Interpretable sequence learning for COVID-19 forecasting. *Neural Information Processing Systems*, 2020.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *International Conference on Machine Learning*, 2018.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. *Neural Information Processing Systems*, 2015.

Konstantinos Benidis, Syama Sundar Rangapuram, Valentin Flunkert, Bernie Wang, Danielle Maddix, Caner Turkmen, Jan Gasthaus, Michael Bohlke-Schneider, David Salinas, Lorenzo Stella, et al. Neural forecasting: Introduction and literature overview. *arXiv:2004.10240*, 2020.

Chris M Bishop. Training with noise is equivalent to Tikhonov regularization. *Neural Computation*, 7(1): 108–116, 1995.

Guy Blanc, Neha Gupta, Gregory Valiant, and Paul Valiant. Implicit regularization for deep neural networks driven by an Ornstein–Uhlenbeck like process. *Conference on Learning Theory*, 2020.

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. *SIGNLL Conference on Computational Natural Language Learning*, 2016.

Guobin Chang. Robust Kalman filtering based on Mahalanobis distance as outlier judging criterion. *Journal of Geodesy*, 88(4):391–401, 2014.

Yitian Chen, Yanfei Kang, Yixiong Chen, and Zizhuo Wang. Probabilistic forecasting with temporal convolutional neural network. *Neurocomputing*, 399: 491–501, 2020.

Ping-yeh Chiang, Michael J Curry, Ahmed Abdelkader, Aounon Kumar, John Dickerson, and Tom Goldstein. Detection as regression: Certified object detection by median smoothing. *Neural Information Processing Systems*, 2020.

Tomáš Cipra. Robust exponential smoothing. *Journal of Forecasting*, 11(1):57–69, 1992.

Tomáš Cipra and R Romera. Kalman filter with outliers and missing observations. *Test*, 6(2):379–395, 1997.

Tomáš Cipra, José Trujillo, and Asunción Robio. Holt-winters method with missing observations. *Management Science*, 41(1):174–178, 1995.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. *International Conference on Machine Learning*, 2019.

J.T. Connor, R.D. Martin, and L.E. Atlas. Recurrent neural networks and robust time series prediction. *IEEE Transactions on Neural Networks*, 5(2):240–254, 1994.

Alex Damian, Tengyu Ma, and Jason Lee. Label noise SGD provably prefers flat global minimizers. *Neural Information Processing Systems*, 2021.

Raphaël Dang-Nhu, Gagandeep Singh, Pavol Bielik, and Martin Vechev. Adversarial attacks on probabilistic autoregressive forecasting models. *International Conference on Machine Learning*, 2020.

Emmanuel de Bézenac, Syama Sundar Rangapuram, Konstantinos Benidis, Michael Bohlke-Schneider, Richard Kurle, Lorenzo Stella, Hilaf Hasson, Patrick Gallinari, and Tim Januschowski. Normalizing Kalman filters for multivariate time series analysis. *Neural Information Processing Systems*, 2020.

Krishnamurthy Dvijotham, Sven Gowal, Robert Stanforth, Relja Arandjelovic, Brendan O'Donoghue, Jonathan Uesato, and Pushmeet Kohli. Training verified learners with learned verifiers. *arXiv:1805.10265*, 2018.

Carson Eisenach, Yagna Patel, and Dhruv Madeka. MQTransformer: Multi-Horizon forecasts with

context dependent and feedback-aware attention. *arXiv:2009.14799*, 2022.

Chenyou Fan, Yuze Zhang, Yi Pan, Xiaoyue Li, Chi Zhang, Rong Yuan, Di Wu, Wensheng Wang, Jian Pei, and Heng Huang. Multi-horizon time series forecasting with temporal attention learning. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Adversarial attacks on deep neural networks for time series classification. *International Joint Conference on Neural Networks*, pages 1–8, 2019.

Roman Garnett, Michael A Osborne, and Stephen J Roberts. Sequential Bayesian prediction in the presence of changepoints. *International Conference on Machine Learning*, 2009.

Sarah Gelper, Roland Fried, and Christophe Croux. Robust forecasting with exponential and Holt–Winters smoothing. *Journal of Forecasting*, 29(3):285–300, 2010.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *International Conference on Learning Representations*, 2015.

Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet Kohli. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv:1810.12715*, 2018.

Tian Guo, Zhao Xu, Xin Yao, Haifeng Chen, Karl Aberer, and Koichi Funaya. Robust online time series prediction with recurrent neural networks. *IEEE International Conference on Data Science and Advanced Analytics*, 2016.

Hansika Hewamalage, Christoph Bergmeir, and Kasun Bandara. Recurrent neural networks for time series forecasting: Current status and future directions. *International Journal of Forecasting*, 37(1):388–427, 2021.

Lasse Holmstrom and Petri Koistinen. Using additive noise in back-propagation training. *IEEE Transactions on Neural Networks*, 3(1):24–38, 1992.

MJC Hu and Halbert E Root. An adaptive data processing system for weather forecasting. *Journal of Applied Meteorology and Climatology*, 3(5):513–523, 1964.

Xiaoyong Jin, Youngsuk Park, Danielle C. Maddix, Hao Wang, and Yuyang Wang. Domain adaptation for time series forecasting via attention sharing. *arXiv:2102.06828*, 2022.

Fazle Karim, Somshubra Majumdar, and Houshang Darabi. Adversarial attacks on time series. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(10):3309–3320, 2021.

Jongho Kim, Youngsuk Park, John D Fox, Stephen P Boyd, and William Dally. Optimal operation of a plug-in hybrid vehicle with battery thermal and degradation model. *American Control Conference*, 2020.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *International Conference on Learning Representations*, 2017.

Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. *IEEE Symposium on Security and Privacy*, 2019.

Alexander Levine, Sahil Singla, and Soheil Feizi. Certifiably robust interpretation in deep learning. *arXiv:1905.12105*, 2019.

Bai Li, Changyou Chen, Wenlin Wang, and Lawrence Carin. Certified adversarial robustness with additive noise. *Neural Information Processing Systems*, 2019a.

Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. *Neural Information Processing Systems*, 2019b.

Bryan Lim, Sercan Ö. Arık, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4):1748–1764, 2021.

Yucheng Lu, Youngsuk Park, Lifan Chen, Yuyang Wang, Christopher De Sa, and Dean Foster. Variance reduced training with stratified sampling for forecasting models. *International Conference on Machine Learning*, 2021.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *International Conference on Learning Representations*, 2018.

Kiyotoshi Matsuoka. Noise injection into inputs in back-propagation learning. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(3):436–440, 1992.

Mohammad Norouzi, Samy Bengio, Navdeep Jaitly, Mike Schuster, Yonghui Wu, Dale Schuurmans, et al. Reward augmented maximum likelihood for neural structured prediction. *Neural Information Processing Systems*, 2016.

Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-BEATS: Neural basis expan-

sion analysis for interpretable time series forecasting. *arXiv:1905.10437*, 2019.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. *ACM Asia Conference on Computer and Communications Security*, 2017.

Youngsuk Park, Kanak Mahadik, Ryan A Rossi, Gang Wu, and Handong Zhao. Linear quadratic regulator for resource-efficient cloud services. *ACM Symposium on Cloud Computing*, 2019.

Youngsuk Park, Ryan Rossi, Zheng Wen, Gang Wu, and Handong Zhao. Structured policy iteration for linear quadratic regulator. *International Conference on Machine Learning*, 2020.

Youngsuk Park, Danielle Maddix, François-Xavier Aubet, Kelvin Kan, Jan Gasthaus, and Yuyang Wang. Learning quantile functions without quantile crossing for distribution-free time series forecasting. *arXiv:2111.06581*, 2021.

Fotios Petropoulos, Daniele Apiletti, Vassilios Assimakopoulos, Mohamed Zied Babai, Devon K. Barrow, Souhaib Ben Taieb, Christoph Bergmeir, Ricardo J. Bessa, Jakub Bijak, John E. Boylan, Jethro Browell, Claudio Carnevale, Jennifer L. Castle, Pasquale Cirillo, Michael P. Clements, Clara Cordeiro, Fernando Luiz Cyrino Oliveira, Shari De Baets, Alexander Dokumentov, Joanne Ellison, Piotr Fiszeder, Philip Hans Franses, David T. Frazier, Michael Gilliland, M. Sinan Gönül, Paul Goodwin, Luigi Grossi, Yael Grushka-Cockayne, Mariangela Guidolin, Massimo Guidolin, Ulrich Gunter, Xiaojia Guo, Renato Guseo, Nigel Harvey, David F. Hendry, Ross Hollyman, Tim Januschowski, Jooyoung Jeon, Victor Richmond R. Jose, Yanfei Kang, Anne B. Koehler, Stephan Kolassa, Nikolaos Kourentzes, Sonia Leva, Feng Li, Konstantia Litsiou, Spyros Makridakis, Gael M. Martin, Andrew B. Martinez, Sheik Meeran, Theodore Modis, Konstantinos Nikolopoulos, Dilek Önkal, Alessia Paccagnini, Anastasios Panagiotelis, Ioannis Panapakidis, Jose M. Pavía, Manuela Pedio, Diego J. Pedregal, Pierre Pinson, Patrícia Ramos, David E. Rapach, J. James Reade, Bahman Rostami-Tabar, Michał Rubaszek, Georgios Sermpinis, Han Lin Shang, Evangelos Spiliotis, Aris A. Syntetos, Priyanga Dilini Talagala, Thiyanga S. Talagala, Len Tashman, Dimitrios Thomakos, Thordis Thorarinsdottir, Ezio Todini, Juan Ramón Trapero Arenas, Xiaoqian Wang, Robert L. Winkler, Alisa Yusupova, and Florian Ziel. Forecasting: theory and practice. *International Journal of Forecasting*, 2022.

Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. *International Conference on Learning Representations*, 2018a.

Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Semidefinite relaxations for certifying robustness to adversarial examples. *Neural Information Processing Systems*, 2018b.

Syama Sundar Rangapuram, Matthias W Seeger, Jan Gasthaus, Lorenzo Stella, Yuyang Wang, and Tim Januschowski. Deep state space models for time series forecasting. *Neural Information Processing Systems*, 2018.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. *International Conference on Learning Representations*, 2016.

Goce Ristanoski, Wei Liu, and James Bailey. A time-dependent enhanced support vector machine for time series regression. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2013.

David Salinas, Michael Bohlke-Schneider, Laurent Callot, Roberto Medico, and Jan Gasthaus. High-dimensional multivariate forecasting with low-rank Gaussian copula processes. *Neural Information Processing Systems*, 2019.

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.

Hadi Salman, Greg Yang, Jerry Li, Pengchuan Zhang, Huan Zhang, Ilya Razenshteyn, and Sebastien Bubeck. Provably robust deep learning via adversarially trained smoothed classifiers. *Neural Information Processing Systems*, 2019.

Matteo Sangiorgio and Fabio Dercole. Robustness of LSTM neural networks for multi-step forecasting of chaotic time series. *Chaos, Solitons & Fractals*, 139: 110045, 2020.

Florian Schmidt. Generalization in generation: A closer look at exposure bias. *EMNLP–IJCNLP Workshop on Neural Generation and Translation*, 2019.

Rajat Sen, Hsiang-Fu Yu, and Inderjit Dhillon. Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting. *Neural Information Processing Systems*, 2019.

Jocelyn Sietsma and Robert JF Dow. Creating artificial neural networks that generalize. *Neural Networks*, 4 (1):67–79, 1991.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks.

International Conference on Learning Representations, 2013.

Jo-Anne Ting, Evangelos Theodorou, and Stefan Schaal. A Kalman filter for robust outlier detection. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007.

Yuyang Wang, Alex Smola, Danielle Maddix, Jan Gasthaus, Dean Foster, and Tim Januschowski. Deep factors for forecasting. *International Conference on Machine Learning*, 2019.

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

Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. *International Conference on Machine Learning*, 2018.

Eric Wong, Frank R Schmidt, Jan Hendrik Metzen, and J Zico Kolter. Scaling provable adversarial defenses. *Neural Information Processing Systems*, 2018.

G Peter Zhang. A neural network ensemble method with jittered training data for time series forecasting. *Information Sciences*, 177(23):5329–5346, 2007.

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 Conference on Artificial Intelligence*, 2021.

Supplementary Material: Robust Probabilistic Time Series Forecasting

A PROOF OF THEORETICAL RESULTS

A.1 Randomized Smoothing of Deterministic Bounded Functions

We first discuss some preliminary results on randomized smoothing of deterministic functions, which will be useful for the proofs in the subsequent sections. Roughly speaking, smoothing *any* bounded function results in a function with smoothness (Lipschitz continuity) property. In the classification setup, Cohen et al. (2019) provided a tight result bounding the certified radius, which is a lower bound on the norm of adversarial perturbation needed to incur incorrect classification. Here we state its generalized version, proved in Salman et al. (2019); Levine et al. (2019).

Lemma 6 (Salman et al. (2019); Levine et al. (2019))**.** *Given a function $h : \mathbb{R}^d \to [0, 1]$, define*

$$\hat{h}(\boldsymbol{x}) := (h * \mathcal{N}(0, I_d))(\boldsymbol{x}) = \mathop{\mathbb{E}}_{\boldsymbol{Z} \sim \mathcal{N}(0, I_d)}[h(\boldsymbol{x} + \boldsymbol{Z})].$$

Then $\|\nabla \hat{h}(\boldsymbol{x})\| \leqslant \phi(\Phi^{-1}(\hat{h}(\boldsymbol{x})))$, which implies that the mapping $\boldsymbol{x} \mapsto \Phi^{-1}(\hat{h}(\boldsymbol{x}))$ is 1-Lipschitz continuous.

To repeat the nomenclature, ϕ, Φ are respectively the pdf and cdf of the standard normal distribution. Note that if we denote $p_\sigma(\boldsymbol{z}) = (2\pi\sigma^2)^{-d/2} \exp\left(-\|\boldsymbol{z}\|^2/2\sigma^2\right)$ the pdf for the multivariate Gaussian distribution $\mathcal{N}(0, \sigma^2 I_d)$ and $p(\boldsymbol{z}) = p_1(\boldsymbol{z})$, then using the change of variables $\boldsymbol{z}' = \boldsymbol{x} + \boldsymbol{z}$, we have

$$\begin{aligned}
\nabla \hat{h}(\boldsymbol{x}) &= \nabla_{\boldsymbol{x}} \left(\int_{\boldsymbol{z} \in \mathbb{R}^d} h(\boldsymbol{x} + \boldsymbol{z}) p(\boldsymbol{z}) \, d\boldsymbol{z} \right) \\
&= \nabla_{\boldsymbol{x}} \left(\int_{\boldsymbol{z}' \in \mathbb{R}^d} h(\boldsymbol{z}') p(\boldsymbol{z}' - \boldsymbol{x}) \, d\boldsymbol{z}' \right) \\
&= \int_{\boldsymbol{z}' \in \mathbb{R}^d} h(\boldsymbol{z}') (\boldsymbol{z}' - \boldsymbol{x}) p(\boldsymbol{z}' - \boldsymbol{x}) \, d\boldsymbol{z}' \\
&= \int_{\boldsymbol{z} \in \mathbb{R}^d} h(\boldsymbol{x} + \boldsymbol{z}) \boldsymbol{z} p(\boldsymbol{z}) \, d\boldsymbol{z}. \qquad (18)
\end{aligned}$$

The proof from Salman et al. (2019) uses the argument that if $\boldsymbol{z} \mapsto \psi(\boldsymbol{z})$ is a function such that $0 \leqslant \psi \leqslant 1$ and

$$\int_{\boldsymbol{z} \in \mathbb{R}^d} \psi(\boldsymbol{z}) p(\boldsymbol{z}) \, d\boldsymbol{z} = s,$$

then for any unit vector $\boldsymbol{u}$, the following inequality holds:

$$\int_{\boldsymbol{z} \in \mathbb{R}^d} \psi(\boldsymbol{z}) (\boldsymbol{u} \cdot \boldsymbol{z}) p(\boldsymbol{z}) \, d\boldsymbol{z} \leqslant \int_{\boldsymbol{z} \in \mathbb{R}^d} \mathbf{1}_{\{\boldsymbol{z} \in \mathbb{R}^d \,|\, \boldsymbol{u} \cdot \boldsymbol{z} \geqslant -\Phi^{-1}(s)\}}(\boldsymbol{z}) (\boldsymbol{u} \cdot \boldsymbol{z}) p(\boldsymbol{z}) \, d\boldsymbol{z} = \phi(\Phi^{-1}(s)). \qquad (19)$$

Indeed, given the constrained budget on $\mathbb{E}_{\boldsymbol{Z} \sim \mathcal{N}(0, I_d)}[\psi(\boldsymbol{Z})]$, one will maximize $\mathbb{E}_{\boldsymbol{Z} \sim \mathcal{N}(0, I_d)}[(\boldsymbol{u} \cdot \boldsymbol{Z}) \psi(\boldsymbol{Z})]$ only by concentrating the mass $\psi(\boldsymbol{z})$ in the region with larger values of $\boldsymbol{u} \cdot \boldsymbol{z}$, i.e., on the set of the form $\{\boldsymbol{z} \in \mathbb{R}^d \,|\, \boldsymbol{u} \cdot \boldsymbol{z} \geqslant c\}$ for some $c \in \mathbb{R}$. Assuming $\boldsymbol{u} = (1, 0, \ldots, 0) \in \mathbb{R}^d$ without loss of generality, we get

$$\int_{\boldsymbol{z} \in \mathbb{R}^d} \mathbf{1}_{\{\boldsymbol{z} \in \mathbb{R}^d \,|\, \boldsymbol{u} \cdot \boldsymbol{z} \geqslant c\}}(\boldsymbol{z}) p(\boldsymbol{z}) \, d\boldsymbol{z} = \int_c^\infty \phi(z_1) \, dz_1 = s \iff c = -\Phi^{-1}(s).$$

A.2 Proof of Theorem 1

Note that in (13), we abused the notation and written the W_1 distance in terms of cumulative distribution functions, as one can directly quantify the W_1 distance in terms of cdf's: if F, G are respectively the cdf of a real-valued random variable and μ, ν are the corresponding distributions, then

$$W_1(\mu,\nu) = \int_0^1 |F^{-1}(p) - G^{-1}(p)|\, dp = \int_{-\infty}^{\infty} |F(r) - G(r)|\, dr. \tag{20}$$

In the case of smoothed scalar predictors g_σ, respectively evaluated at $\boldsymbol{x}$ and $\boldsymbol{x}+\boldsymbol{\delta}$, we have

$$\begin{aligned}
W_1(G_{\boldsymbol{x},\sigma}, G_{\boldsymbol{x}+\boldsymbol{\delta},\sigma}) &= \int_{-\infty}^{\infty} |G_{\boldsymbol{x},\sigma}(r) - G_{\boldsymbol{x}+\boldsymbol{\delta},\sigma}(r)|\ dr \\
&= \int_{-\infty}^{\infty} \left| \int_{\boldsymbol{z}\in\mathbb{R}^T} F_{\boldsymbol{x}+\boldsymbol{z}}(r) p_\sigma(\boldsymbol{z})\, d\boldsymbol{z} - \int_{\boldsymbol{z}\in\mathbb{R}^T} F_{\boldsymbol{x}+\boldsymbol{\delta}+\boldsymbol{z}}(r) p_\sigma(\boldsymbol{z})\, d\boldsymbol{z} \right| dr \\
&= \int_{-\infty}^{\infty} \left| \int_{\boldsymbol{z}\in\mathbb{R}^T} F_{\boldsymbol{z}}(r)(p_\sigma(\boldsymbol{z}-\boldsymbol{x}) - p_\sigma(\boldsymbol{z}-\boldsymbol{x}-\boldsymbol{\delta}))\, d\boldsymbol{z} \right| dr.
\end{aligned}$$

Now note that

$$\begin{aligned}
\int_{\boldsymbol{z}\in\mathbb{R}^T} F_{\boldsymbol{z}}(r)(p_\sigma(\boldsymbol{z}-\boldsymbol{x}) - p_\sigma(\boldsymbol{z}-\boldsymbol{x}-\boldsymbol{\delta}))\, d\boldsymbol{z} &= \int_{\boldsymbol{z}\in\mathbb{R}^T} F_{\boldsymbol{z}}(r) \int_0^1 \nabla p_\sigma(\boldsymbol{z}-\boldsymbol{x}-t\boldsymbol{\delta}) \cdot (-\boldsymbol{\delta})\, dt\, d\boldsymbol{z} \\
&= \int_0^1 \int_{\boldsymbol{z}\in\mathbb{R}^T} F_{\boldsymbol{z}}(r) \left(\boldsymbol{\delta} \cdot \frac{\boldsymbol{z}-\boldsymbol{x}-t\boldsymbol{\delta}}{\sigma^2} \right) p_\sigma(\boldsymbol{z}-\boldsymbol{x}-t\boldsymbol{\delta})\, d\boldsymbol{z}\, dt \\
&= \frac{1}{\sigma} \int_0^1 \int_{\boldsymbol{z}'\in\mathbb{R}^T} F_{\boldsymbol{x}+t\boldsymbol{\delta}+\sigma\boldsymbol{z}'}(r)\, (\boldsymbol{\delta}\cdot\boldsymbol{z}')\, p(\boldsymbol{z}')\, d\boldsymbol{z}'\, dt
\end{aligned}$$

where in the last line we make the change of variables $\boldsymbol{z}' = \frac{\boldsymbol{z}-\boldsymbol{x}-t\boldsymbol{\delta}}{\sigma}$. Because the mapping $\boldsymbol{z}' \mapsto F_{\boldsymbol{x}+t\boldsymbol{\delta}+\sigma\boldsymbol{z}'}(r)$ is a function $\mathbb{R}^T \to [0,1]$, by (19) we have

$$\begin{aligned}
\left| \int_{\boldsymbol{z}\in\mathbb{R}^T} F_{\boldsymbol{z}}(r)(p_\sigma(\boldsymbol{z}-\boldsymbol{x}) - p_\sigma(\boldsymbol{z}-\boldsymbol{x}-\boldsymbol{\delta}))\, d\boldsymbol{z} \right| &\leqslant \frac{1}{\sigma} \int_0^1 \left| \int_{\boldsymbol{z}'\in\mathbb{R}^T} F_{\boldsymbol{x}+t\boldsymbol{\delta}+\sigma\boldsymbol{z}'}(r)\, (\boldsymbol{\delta}\cdot\boldsymbol{z}')\, p(\boldsymbol{z}')\, d\boldsymbol{z}' \right| dt \\
&\leqslant \frac{1}{\sigma} \int_0^1 \|\boldsymbol{\delta}\| \cdot \left\| \int_{\boldsymbol{z}'\in\mathbb{R}^T} F_{\boldsymbol{x}+t\boldsymbol{\delta}+\sigma\boldsymbol{z}'}(r) \boldsymbol{z}' p(\boldsymbol{z}')\, d\boldsymbol{z}' \right\| dt \\
&\leqslant \frac{1}{\sigma} \int_0^1 \|\boldsymbol{\delta}\| \cdot \phi\left(\Phi^{-1}\left(\int_{\boldsymbol{z}'\in\mathbb{R}^T} F_{\boldsymbol{x}+t\boldsymbol{\delta}+\sigma\boldsymbol{z}'}(r)\, p(\boldsymbol{z}')\, d\boldsymbol{z}' \right) \right) dt \\
&= \frac{1}{\sigma} \int_0^1 \|\boldsymbol{\delta}\| \cdot \phi(\Phi^{-1}(G_{\boldsymbol{x}+t\boldsymbol{\delta},\sigma}(r)))\, dt.
\end{aligned} \tag{21}$$

Therefore, provided that $\int_{-\infty}^{\infty} \phi(\Phi^{-1}(G_{\boldsymbol{x}',\sigma}(r)))\, dr < C$ for $\boldsymbol{x}'$ near $\boldsymbol{x}$ for some $C > 0$, one can apply the dominated convergence theorem to obtain

$$\begin{aligned}
\limsup_{\|\boldsymbol{\delta}\|\to 0} \frac{W_1(G_{\boldsymbol{x},\sigma}, G_{\boldsymbol{x}+\boldsymbol{\delta},\sigma})}{\|\boldsymbol{\delta}\|} &\leqslant \limsup_{\|\boldsymbol{\delta}\|\to 0} \int_{-\infty}^{\infty} \frac{1}{\sigma} \int_0^1 \phi(\Phi^{-1}(G_{\boldsymbol{x}+t\boldsymbol{\delta},\sigma}(r)))\, dt\, dr \\
&= \int_{-\infty}^{\infty} \frac{1}{\sigma} \int_0^1 \lim_{\|\boldsymbol{\delta}\|\to 0} \phi(\Phi^{-1}(G_{\boldsymbol{x}+t\boldsymbol{\delta},\sigma}(r)))\, dt\, dr \\
&= \frac{1}{\sigma} \int_{-\infty}^{\infty} \phi\left(\Phi^{-1}(G_{\boldsymbol{x},\sigma}(r))\right)\, dr \\
&= \mathrm{Ro}(\boldsymbol{x};\sigma),
\end{aligned}$$

where we applied the fact that $\lim_{\|\boldsymbol{\delta}\|\to 0} G_{\boldsymbol{x}+t\boldsymbol{\delta},\sigma}(r) = G_{\boldsymbol{x}}(r)$ by Lemma 6.

A.3 Proof of Lemma 2

For $r < 0$, we proceed similarly as we bounded (21), but bound the integrand in a different way:

$$\begin{aligned}
&\left| \int_{\boldsymbol{z}\in\mathbb{R}^T} \Pr\left[f_j(\boldsymbol{z}) \leqslant r\right] \left(p_\sigma(\boldsymbol{z}-\boldsymbol{x}) - p_\sigma(\boldsymbol{z}-\boldsymbol{x}-\boldsymbol{\delta})\right) d\boldsymbol{z} \right| \\
&\quad \leqslant \frac{1}{\sigma} \int_0^1 \left| \int_{\boldsymbol{z}\in\mathbb{R}^T} \Pr\left[f_j(\boldsymbol{x}+t\delta+\sigma\boldsymbol{z}) \leqslant r\right] (\boldsymbol{\delta}\cdot\boldsymbol{z})\, p(\boldsymbol{z})\, d\boldsymbol{z} \right| dt \\
&\quad \leqslant \frac{1}{\sigma} \int_0^1 \|\boldsymbol{\delta}\| \int_{\boldsymbol{z}\in\mathbb{R}^T} \Pr\left[f_j(\boldsymbol{x}+t\delta+\sigma\boldsymbol{z}) \leqslant r\right] \|\boldsymbol{z}\|\, p(\boldsymbol{z})\, d\boldsymbol{z}\, dt
\end{aligned}$$

for $j = 1, \ldots, \tau$. Note that

$$\int_{\boldsymbol{z}\in\mathbb{R}^T} p_\sigma(\boldsymbol{z}-\boldsymbol{x}) - p_\sigma(\boldsymbol{z}-\boldsymbol{x}-\boldsymbol{\delta})\, d\boldsymbol{z} = 0,$$

so for $r > 0$, we can write

$$\begin{aligned}
&\left| \int_{\boldsymbol{z}\in\mathbb{R}^T} \Pr\left[f_j(\boldsymbol{z}) \leqslant r\right] \left(p_\sigma(\boldsymbol{z}-\boldsymbol{x}) - p_\sigma(\boldsymbol{z}-\boldsymbol{x}-\boldsymbol{\delta})\right) d\boldsymbol{z} \right| \\
&\quad = \left| \int_{\boldsymbol{z}\in\mathbb{R}^T} (\Pr\left[f_j(\boldsymbol{z}) \leqslant r\right] - 1)(p_\sigma(\boldsymbol{z}-\boldsymbol{x}) - p_\sigma(\boldsymbol{z}-\boldsymbol{x}-\boldsymbol{\delta}))\, d\boldsymbol{z} \right| \\
&\quad = \left| \int_{\boldsymbol{z}\in\mathbb{R}^T} \Pr\left[f_j(\boldsymbol{z}) > r\right] \left(p_\sigma(\boldsymbol{z}-\boldsymbol{x}) - p_\sigma(\boldsymbol{z}-\boldsymbol{x}-\boldsymbol{\delta})\right) d\boldsymbol{z} \right| \\
&\quad \leqslant \frac{1}{\sigma} \int_0^1 \|\boldsymbol{\delta}\| \int_{\boldsymbol{z}\in\mathbb{R}^T} \Pr\left[f_j(\boldsymbol{x}+t\delta+\sigma\boldsymbol{z}) > r\right] \|\boldsymbol{z}\|\, p(\boldsymbol{z})\, d\boldsymbol{z}\, dt.
\end{aligned}$$

Therefore, if we denote $F_{j,\boldsymbol{x}}(r) = \Pr\left[f_j(\boldsymbol{x}) \leqslant r\right]$ and $G_{j,\boldsymbol{x},\sigma}(r) = \int_{\boldsymbol{z}\in\mathbb{R}^T} F_{j,\boldsymbol{x}+\boldsymbol{z}}(r) p_\sigma(\boldsymbol{z})\, d\boldsymbol{z}$,

$$\begin{aligned}
W_1(G_{j,\boldsymbol{x},\sigma}, G_{j,\boldsymbol{x}+\boldsymbol{\delta},\sigma}) &= \int_{-\infty}^{\infty} \left| \int_{\boldsymbol{z}\in\mathbb{R}^T} F_{j,\boldsymbol{z}}(r)(p_\sigma(\boldsymbol{z}-\boldsymbol{x}) - p_\sigma(\boldsymbol{z}-\boldsymbol{x}-\boldsymbol{\delta}))\, d\boldsymbol{z} \right| dr \\
&= \int_{-\infty}^{0} \left| \int_{\boldsymbol{z}\in\mathbb{R}^T} F_{j,\boldsymbol{z}}(r)(p_\sigma(\boldsymbol{z}-\boldsymbol{x}) - p_\sigma(\boldsymbol{z}-\boldsymbol{x}-\boldsymbol{\delta}))\, d\boldsymbol{z} \right| dr \\
&\quad + \int_{0}^{\infty} \left| \int_{\boldsymbol{z}\in\mathbb{R}^T} F_{j,\boldsymbol{z}}(r)(p_\sigma(\boldsymbol{z}-\boldsymbol{x}) - p_\sigma(\boldsymbol{z}-\boldsymbol{x}-\boldsymbol{\delta}))\, d\boldsymbol{z} \right| dr \\
&= \frac{\|\boldsymbol{\delta}\|}{\sigma} \int_0^\infty \int_0^1 \int_{\boldsymbol{z}\in\mathbb{R}^T} (\Pr\left[f_j(\boldsymbol{x}+t\boldsymbol{\delta}+\sigma\boldsymbol{z}) \leqslant -r\right] + \Pr\left[f_j(\boldsymbol{x}+t\boldsymbol{\delta}+\sigma\boldsymbol{z}) > r\right]) \|\boldsymbol{z}\|\, p(\boldsymbol{z})\, d\boldsymbol{z}\, dt\, dr \\
&\leqslant \frac{\|\boldsymbol{\delta}\|}{\sigma} \left(\int_0^\infty \varphi(r)\, dr \right) \left(\int_{\boldsymbol{z}\in\mathbb{R}^T} \|\boldsymbol{z}\| p(\boldsymbol{z})\, d\boldsymbol{z} \right).
\end{aligned}$$

This shows that the smoothed forecaster $g(\boldsymbol{x}) = (g_1(\boldsymbol{x}), \ldots, g_\tau(\boldsymbol{x}))$ is $C\eta - \eta$ robust in the given sense, where

$$C = \frac{1}{\sigma} \left(\int_0^\infty \varphi(r)\, dr \right) \left(\int_{\boldsymbol{z}\in\mathbb{R}^T} \|\boldsymbol{z}\| p(\boldsymbol{z})\, d\boldsymbol{z} \right) < \infty.$$

Remark. The constant C may be chosen more tightly; for example, applying the bound in the above proof for $r > R$ with some large R, and using the original bound of Theorem 1 for $r \leqslant R$ would result in

$$C = \frac{1}{\sigma} \left(\int_{-R}^{R} \phi(\Phi^{-1}(G_{\boldsymbol{x},\sigma}(r)))\, dr + \int_R^\infty \varphi(r)\, dr \int_{\boldsymbol{z}\in\mathbb{R}^T} \|\boldsymbol{z}\| p(\boldsymbol{z})\, d\boldsymbol{z} \right),$$

which can be smaller.

A.4 Proof of Lemma 3

Suppose that there is $M_1, M_2 > 0$ such that $|\mathbb{E}[f(\boldsymbol{x})]| \leqslant M_1$ and $\mathrm{Var}[f(\boldsymbol{x})] \leqslant M_2$ holds for all $\boldsymbol{x} \in \mathbb{R}^T$. Then for any $\boldsymbol{x} \in \mathbb{R}^T$ and $r > 0$,

$$r^2 \mathrm{Pr}[|f(\boldsymbol{x})| \geqslant r] \leqslant \mathbb{E}[f(\boldsymbol{x})^2] = \mathrm{Var}[f(\boldsymbol{x})] + |\mathbb{E}[f(\boldsymbol{x})]|^2 \leqslant M_1^2 + M_2.$$

Thus one can simply take $\varphi(r) = 1$ for $r \in [0,1]$ and $\varphi(r) = \frac{M_1^2 + M_2}{r^2}$ for $r > 1$.

A.5 Proof of Corollary 5

Denote $g_\sigma(\boldsymbol{x}) = (Y_1, Y_2, \dots)$ and $g_\sigma(\boldsymbol{x}; \tilde{\boldsymbol{x}}_{T+1:T+k}) = (Y'_{k+1}, Y'_{k+2}, \dots)$. Denote the probability measures corresponding to the random variables Y_h, Y'_h respectively by μ_h, μ'_h. When $\boldsymbol{x} \in \mathbb{R}^T$ is fixed, we can view $f^{(h)}(\boldsymbol{x}; y_1, \dots, y_{h-1})$ as a function of $(y_1, \dots, y_{h-1}) \in \mathbb{R}^h$, for each $h = 1, 2, \dots$. Because we have

$$\Pr\left[\left|f^{(h)}(\boldsymbol{x}; y_1, \dots, y_{h-1})\right| \geqslant r\right] \leqslant \varphi(r),$$

we can apply Corollary 4 with $f^{(h)}(\boldsymbol{x}; \cdot) : \mathbb{R}^{h-1} \to \mathbb{R}$, which implies that there exists $C_h > 0$ such that

$$g_\sigma^{(h)}(\boldsymbol{x}; y_1, \dots, y_{h-1}) = \mathop{\mathbb{E}}_{\boldsymbol{\zeta} \sim \mathcal{N}(0, \sigma^2 I_{h-1})} [f(\boldsymbol{x}_{1:T}; y_1 + \zeta_1, \cdots, y_{h-1} + \zeta_{h-1})]$$

satisfies

$$W_1\left(G^{(h)}_{\boldsymbol{y}', \sigma}, G^{(h)}_{\boldsymbol{y}, \sigma}\right) \leqslant C_h \left\|\boldsymbol{y}' - \boldsymbol{y}\right\|_2$$

for any $\boldsymbol{y} = (y_1, \dots, y_{h-1})$ and $\boldsymbol{y}' = (y'_1, \dots, y'_{h-1})$, where $G^{(k)}_{\boldsymbol{y}, \sigma}$ is the cdf for the random variable $g_\sigma^{(k)}(\boldsymbol{y})$ and W_1 distance between the cdfs denotes the W_1 distance between the corresponding probability measures by abuse of notation. Applying the above bound in the case $h = k+1$, $\boldsymbol{y} = \hat{\boldsymbol{x}}_{T+1:T+k}$ and $\boldsymbol{y}' = \tilde{\boldsymbol{x}}_{T+1:T+k}$ gives

$$W_1(\mu'_{k+1}, \mu_{k+1}) = W_1\left(G^{(k)}_{\tilde{\boldsymbol{x}}_{T+1:T+k}, \sigma}, G^{(k)}_{\hat{\boldsymbol{x}}_{T+1:T+k}, \sigma}\right) \leqslant C_k \left\|\tilde{\boldsymbol{x}}_{T+1:T+k} - \hat{\boldsymbol{x}}_{T+1:T+k}\right\|_2$$

because $Y_{k+1} = g_\sigma^{(k)}(\boldsymbol{x}; \hat{\boldsymbol{x}}_{T+1:T+k})$ and $Y'_{k+1} = g_\sigma^{(k)}(\boldsymbol{x}; \tilde{\boldsymbol{x}}_{T+1:T+k})$. In particular, this bounds the difference between the mean point forecasts:

$$\begin{aligned}
\left|\overline{Y}'_{k+1} - \overline{Y}_{k+1}\right| = \left|\hat{x}'_{T+k+1} - \hat{x}_{T+k+1}\right| &= \left|\mathbb{E}\left[g_\sigma^{(k)}(\boldsymbol{x}; \tilde{\boldsymbol{x}}_{T+1:T+k})\right] - \mathbb{E}[g_\sigma^{(k)}(\boldsymbol{x}; \hat{\boldsymbol{x}}_{T+1:T+k})\right]\right| \\
&\leqslant \sup_{\chi : 1\text{-Lipschitz}} \mathbb{E}\left[\chi\left(g_\sigma^{(k)}(\boldsymbol{x}; \tilde{\boldsymbol{x}}_{T+1:T+k})\right)\right] - \mathbb{E}\left[\chi\left(g_\sigma^{(k)}(\boldsymbol{x}; \hat{\boldsymbol{x}}_{T+1:T+k})\right)\right] \\
&= W_1\left(G^{(h)}_{\boldsymbol{y}', \sigma}, G^{(h)}_{\boldsymbol{y}, \sigma}\right) \\
&\leqslant C_k \left\|\tilde{\boldsymbol{x}}_{T+1:T+k} - \hat{\boldsymbol{x}}_{T+1:T+k}\right\|_2.
\end{aligned}$$

Then we obtain

$$\left\|\left(\tilde{\boldsymbol{x}}_{T+1:T+k}; \hat{x}'_{T+k+1}\right) - \left(\hat{\boldsymbol{x}}_{T+1:T+k}; \hat{x}_{T+k+1}\right)\right\|_2 \leqslant (1 + C_k) \left\|\tilde{\boldsymbol{x}}_{T+1:T+k} - \hat{\boldsymbol{x}}_{T+1:T+k}\right\|_2,$$

which again implies

$$\begin{aligned}
W_1(\mu'_{k+2}, \mu_{k+2}) &= W_1\left(G^{(k)}_{(\tilde{\boldsymbol{x}}_{T+1:T+k}; \hat{\boldsymbol{x}}'_{T+k+1}), \sigma}, G^{(k)}_{(\hat{\boldsymbol{x}}_{T+1:T+k}; \hat{\boldsymbol{x}}_{T+k+1}), \sigma}\right) \\
&\leqslant C_{k+1} \left\|\left(\tilde{\boldsymbol{x}}_{T+1:T+k}; \hat{x}'_{T+k+1}\right) - \left(\hat{\boldsymbol{x}}_{T+1:T+k}; \hat{x}_{T+k+1}\right)\right\|_2 \\
&\leqslant C_{k+1}(1 + C_k) \left\|\tilde{\boldsymbol{x}}_{T+1:T+k} - \hat{\boldsymbol{x}}_{T+1:T+k}\right\|_2.
\end{aligned}$$

Repeating the same argument, we see that

$$T_{\mathcal{Y}}(g_\sigma(\boldsymbol{x})) = T_{\mathcal{Y}}(Y_1, Y_2, \dots) = (Y_{k+1}, Y_{k+2}, \dots) \approx (Y'_{k+1}, Y'_{k+2}, \dots) = g_\sigma\left(\boldsymbol{x}; \tilde{\boldsymbol{x}}_{T+1:T+k}\right) = g_\sigma\left(T_{\mathcal{X}}(\boldsymbol{x})\right)$$

in the sense that

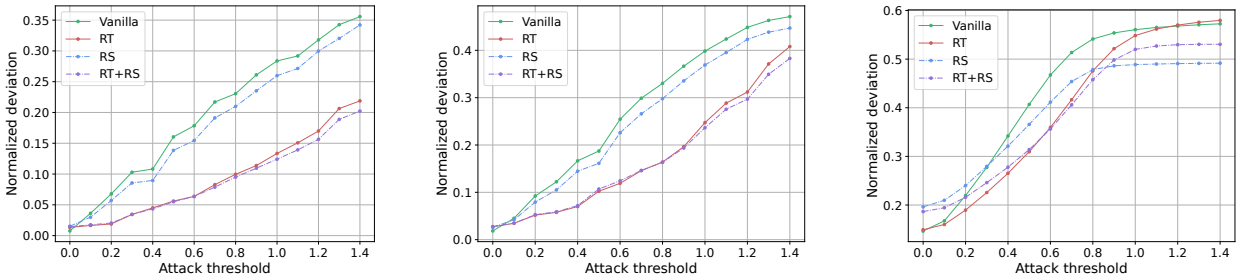
$$W_1(\mu'_{k+j}, \mu_{k+j}) = O\left(\left\|\tilde{\boldsymbol{x}}_{T+1:T+k} - \hat{\boldsymbol{x}}_{T+1:T+k}\right\|_2\right) = O\left(d_{\mathcal{X}}(\boldsymbol{x}; T_{\mathcal{X}})\right)$$

holds for each $j = 1, 2, \dots$, which completes the proof.

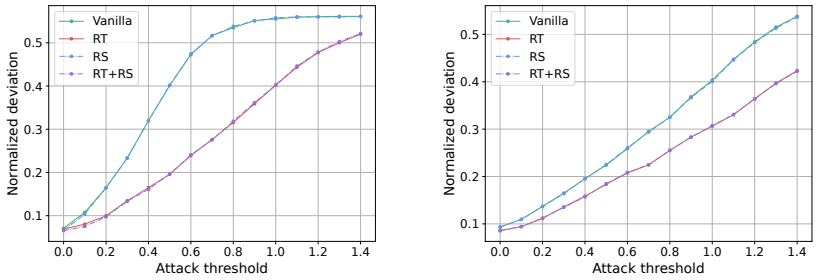
B EXPERIMENTAL DETAILS

We use the time series forecasting library GluonTS (Alexandrov et al., 2020) to configure and train the vanilla and random-trained DeepAR models, and to perform randomized smoothing using them. For the experiments in Sections 5.1 and 5.2, we use the standard prediction lengths $\tau = 24$ for the Electricity and Traffic datasets, $\tau = 30$ for the Exchange Rate dataset, and $\tau = 14$ for the M4-Daily dataset. The context lengths are set to 4τ for all cases, and all the other model hyperparameters are set to default values within the GluonTS implementation. The training of all baseline models (with or without data augmentation with random noises) uses batch size 128 and is run for 50 epochs. We use 100 sample paths from each baseline and smoothed model to perform adversarial attack and generate forecasts. The code for the experiments is available at `https://github.com/tetrzim/robust-probabilistic-forecasting`.

C ADDITIONAL EXPERIMENT RESULTS



(a) Exchange Rate, $\tau = 30$, $\{H\} = \{1\}$ (b) M4-Daily, $\tau = 14$, $\{H\} = \{1\}$ (c) Traffic, $\tau = 24$, $\{H\} = \{1\}$



(d) Electricity, $\tau = 24$, $\{H\} = \{1\}$ (e) Electricity, $\tau = 24$, $\{H\} = \{\tau\}$

Figure 4: ND_H from DeepAR models on different datasets under adversarial attacks with respect to relative l_2 norm. Smoothing of baseline models uses $\sigma = 0.1$ for the Electricity dataset and $\sigma = 0.5$ for other datasets. Randomized training uses $\sigma_{\mathrm{tr}} = 0.1$.

Table 1: Mean and standard deviation of ND on clean test set for *all* prediction indices over 10 runs.

	Vanilla	Random-trained ($\sigma_{\mathrm{tr}} = 0.1$)
Exchange Rate	0.024±0.008	**0.018**±0.001
Traffic	0.131±0.006	**0.127**±0.003
Electricity	0.075±0.010	**0.067**±0.005

Figure 4, together with Figure 2 and Table 2, shows the prediction accuracy (measured in terms of ND on the attacked indices) of DeepAR models under the adversarial attack of Dang-Nhu et al. (2020) with respect to relative l_2 norms. Table 3, together with Figure 3, shows the relative ND on overlapping indices under supplement of a noisy observation.

Table 1 compares ND on the test set without adversarial attack, measured over all prediction indices (that is, ND_H (17) with $H = \{1, \ldots, \tau\}$). Each row indicates that random-trained (RT) models have attained better accuracy compared to the corresponding vanilla model. Because the training of DeepAR involves windowing over multiple time intervals, and we apply noise to every observation available in the training data, randomized training noises both input and output (label) values. It is well-known that randomization of training inputs is positively related to generalization (Sietsma and Dow, 1991; Matsuoka, 1992; Holmstrom and Koistinen, 1992; Bishop, 1995), as observed by Zhang (2007) in the context of time series forecasting. On the other hand, some recent results (Blanc et al., 2020; Damian et al., 2021) provided theoretical study on the implicit bias of optimization algorithms with label noise towards well-generalizing minima. However, we are not aware of prior works that particularly studied the effect of training with label noising on forecasting accuracy.

Table 2: Mean and standard deviation of ND_H from DeepAR models on different datasets under adversarial attacks, measured over 10 independent runs. The * symbols for RS models indicate statistically significant improvement against the corresponding baselines according to the Wilcoxon signed-rank test.

(a) Exchange Rate

H	η	Vanilla	RS ($\sigma = 0.5$)	RT ($\sigma_{\mathrm{tr}} = 0.1$)	RT + RS ($\sigma_{\mathrm{tr}} = 0.1, \sigma = 0.5$)
$\{1\}$	0	$\mathbf{0.008} \pm 0.002$	0.015 ± 0.003	0.013 ± 0.002	0.014 ± 0.003
	0.2	0.068 ± 0.013	$0.057 \pm 0.010^*$	$\mathbf{0.019} \pm 0.002$	$\mathbf{0.020} \pm 0.002$
	0.4	0.108 ± 0.021	$0.090 \pm 0.014^*$	0.045 ± 0.005	$\mathbf{0.043} \pm 0.006$
	0.6	0.178 ± 0.026	$0.154 \pm 0.019^*$	$\mathbf{0.064} \pm 0.012$	$\mathbf{0.063} \pm 0.008$
	0.8	0.231 ± 0.031	$0.210 \pm 0.031^*$	0.100 ± 0.014	$\mathbf{0.095} \pm 0.011$
	1.0	0.284 ± 0.038	$0.260 \pm 0.038^*$	0.133 ± 0.037	$\mathbf{0.124} \pm 0.028^*$
	1.2	0.318 ± 0.048	$0.300 \pm 0.046^*$	0.170 ± 0.044	$\mathbf{0.156} \pm 0.035^*$
	1.4	0.356 ± 0.052	$0.342 \pm 0.054^*$	0.219 ± 0.029	$\mathbf{0.202} \pm 0.024^*$
$\{\tau\}$	0	0.036 ± 0.014	0.043 ± 0.017	$\mathbf{0.022} \pm 0.002$	$\mathbf{0.023} \pm 0.005$
	0.2	0.065 ± 0.016	0.067 ± 0.017	$\mathbf{0.023} \pm 0.002$	$\mathbf{0.023} \pm 0.007$
	0.4	0.111 ± 0.035	$0.099 \pm 0.029^*$	$\mathbf{0.025} \pm 0.002$	$\mathbf{0.026} \pm 0.003$
	0.6	0.164 ± 0.054	$0.142 \pm 0.044^*$	0.035 ± 0.021	$\mathbf{0.031} \pm 0.006$
	0.8	0.194 ± 0.057	$0.174 \pm 0.050^*$	0.054 ± 0.033	$\mathbf{0.049} \pm 0.020$
	1.0	0.210 ± 0.060	$0.192 \pm 0.057^*$	0.117 ± 0.058	$\mathbf{0.099} \pm 0.042^*$
	1.2	0.230 ± 0.059	$0.216 \pm 0.057^*$	0.173 ± 0.066	$\mathbf{0.149} \pm 0.052^*$
	1.4	0.243 ± 0.062	$0.230 \pm 0.060^*$	0.215 ± 0.038	$\mathbf{0.191} \pm 0.034^*$

(b) M4-Daily

H	η	Vanilla	RS ($\sigma = 0.5$)	RT ($\sigma_{\mathrm{tr}} = 0.1$)	RT + RS ($\sigma_{\mathrm{tr}} = 0.1, \sigma = 0.5$)
$\{1\}$	0	$\mathbf{0.018} \pm 0.002$	0.026 ± 0.001	0.027 ± 0.001	0.028 ± 0.002
	0.2	0.092 ± 0.004	$0.079 \pm 0.003^*$	$\mathbf{0.052} \pm 0.007$	$\mathbf{0.053} \pm 0.007$
	0.4	0.167 ± 0.021	$0.145 \pm 0.018^*$	$\mathbf{0.070} \pm 0.012$	0.072 ± 0.011
	0.6	0.254 ± 0.024	$0.226 \pm 0.021^*$	$\mathbf{0.119} \pm 0.011$	0.124 ± 0.011
	0.8	0.330 ± 0.034	$0.298 \pm 0.032^*$	$\mathbf{0.164} \pm 0.028$	$\mathbf{0.164} \pm 0.025$
	1.0	0.398 ± 0.041	$0.369 \pm 0.038^*$	0.247 ± 0.037	$\mathbf{0.236} \pm 0.030^*$
	1.2	0.449 ± 0.027	$0.423 \pm 0.027^*$	0.312 ± 0.041	$\mathbf{0.297} \pm 0.034^*$
	1.4	0.471 ± 0.017	$0.447 \pm 0.015^*$	0.408 ± 0.034	$\mathbf{0.383} \pm 0.033^*$
$\{\tau\}$	0	0.062 ± 0.011	0.065 ± 0.009	$\mathbf{0.056} \pm 0.003$	$\mathbf{0.055} \pm 0.005$
	0.2	0.114 ± 0.019	$0.103 \pm 0.016^*$	$\mathbf{0.062} \pm 0.003$	$\mathbf{0.061} \pm 0.006$
	0.4	0.162 ± 0.020	$0.141 \pm 0.017^*$	$\mathbf{0.074} \pm 0.006$	$\mathbf{0.075} \pm 0.010$
	0.6	0.247 ± 0.027	$0.209 \pm 0.026^*$	$\mathbf{0.084} \pm 0.007$	$\mathbf{0.085} \pm 0.011$
	0.8	0.321 ± 0.044	$0.275 \pm 0.041^*$	0.135 ± 0.058	$\mathbf{0.130} \pm 0.052$
	1.0	0.376 ± 0.052	$0.327 \pm 0.049^*$	0.219 ± 0.096	$\mathbf{0.194} \pm 0.075^*$
	1.2	0.415 ± 0.056	$0.366 \pm 0.053^*$	0.394 ± 0.085	$\mathbf{0.333} \pm 0.072^*$
	1.4	0.442 ± 0.045	$\mathbf{0.392} \pm 0.042^*$	0.493 ± 0.025	$0.415 \pm 0.040^*$

Table 2: *(Continued)* Mean and standard deviation of ND_H from DeepAR models on different datasets under adversarial attacks, measured over 10 independent runs. The * symbols for RS models indicate statistically significant improvement against the corresponding baselines according to the Wilcoxon signed-rank test.

(c) Traffic

H	η	Vanilla	RS ($\sigma = 0.5$)	RT ($\sigma_{\mathrm{tr}} = 0.1$)	RT + RS ($\sigma_{\mathrm{tr}} = 0.1, \sigma = 0.5$)
$\{1\}$	0	$\mathbf{0.146} \pm 0.002$	0.196 ± 0.004	0.149 ± 0.003	0.186 ± 0.007
	0.2	0.220 ± 0.003	0.240 ± 0.005	$\mathbf{0.190} \pm 0.003$	0.216 ± 0.006
	0.4	0.342 ± 0.005	$0.321 \pm 0.004^*$	$\mathbf{0.265} \pm 0.005$	0.278 ± 0.006
	0.6	0.467 ± 0.007	$0.412 \pm 0.005^*$	$\mathbf{0.359} \pm 0.007$	$\mathbf{0.356} \pm 0.013$
	0.8	0.541 ± 0.003	$0.478 \pm 0.004^*$	0.476 ± 0.014	$\mathbf{0.458} \pm 0.024^*$
	1.0	0.561 ± 0.002	$\mathbf{0.489} \pm 0.006^*$	0.548 ± 0.006	$0.520 \pm 0.013^*$
	1.2	0.568 ± 0.002	$\mathbf{0.491} \pm 0.007^*$	0.570 ± 0.003	$0.530 \pm 0.011^*$
	1.4	0.573 ± 0.003	$\mathbf{0.492} \pm 0.007^*$	0.580 ± 0.003	$0.531 \pm 0.011^*$
$\{\tau\}$	0	0.183 ± 0.007	0.229 ± 0.012	$\mathbf{0.179} \pm 0.005$	0.213 ± 0.014
	0.2	0.212 ± 0.008	0.241 ± 0.011	$\mathbf{0.200} \pm 0.005$	0.223 ± 0.014
	0.4	0.261 ± 0.009	0.267 ± 0.011	$\mathbf{0.240} \pm 0.006$	0.245 ± 0.014
	0.6	0.309 ± 0.011	0.296 ± 0.011	0.279 ± 0.007	$\mathbf{0.272 \pm 0.015}$
	0.8	0.367 ± 0.011	$0.340 \pm 0.013^*$	0.324 ± 0.008	$\mathbf{0.307} \pm 0.014^*$
	1.0	0.435 ± 0.014	$0.389 \pm 0.014^*$	0.378 ± 0.011	$\mathbf{0.348} \pm 0.015^*$
	1.2	0.507 ± 0.015	$0.450 \pm 0.015^*$	0.444 ± 0.013	$\mathbf{0.409} \pm 0.016^*$
	1.4	0.561 ± 0.014	$0.491 \pm 0.015^*$	0.505 ± 0.015	$\mathbf{0.464} \pm 0.017^*$

(d) Electricity

H	η	Vanilla	RS ($\sigma = 0.1$)	RT ($\sigma_{\mathrm{tr}} = 0.1$)	RT + RS ($\sigma_{\mathrm{tr}} = 0.1, \sigma = 0.1$)
$\{1\}$	0	0.071 ± 0.003	$\mathbf{0.066} \pm 0.003^*$	0.069 ± 0.009	$\mathbf{0.065} \pm 0.008^*$
	0.2	0.165 ± 0.005	0.164 ± 0.004	0.100 ± 0.009	$\mathbf{0.097} \pm 0.008$
	0.4	0.320 ± 0.012	0.320 ± 0.009	0.165 ± 0.011	$\mathbf{0.161} \pm 0.009^*$
	0.6	0.472 ± 0.013	0.474 ± 0.011	$\mathbf{0.239} \pm 0.011$	$\mathbf{0.241} \pm 0.011$
	0.8	0.535 ± 0.009	0.538 ± 0.011	$\mathbf{0.316} \pm 0.019$	$\mathbf{0.318} \pm 0.017$
	1.0	0.555 ± 0.013	0.558 ± 0.009	$\mathbf{0.402} \pm 0.019$	$\mathbf{0.403} \pm 0.019$
	1.2	0.560 ± 0.011	0.559 ± 0.011	$\mathbf{0.477} \pm 0.017$	$\mathbf{0.478} \pm 0.017$
	1.4	0.559 ± 0.014	0.561 ± 0.011	$\mathbf{0.520} \pm 0.015$	$\mathbf{0.521} \pm 0.015$
$\{\tau\}$	0	0.093 ± 0.018	0.093 ± 0.016	$\mathbf{0.086} \pm 0.013$	$\mathbf{0.086} \pm 0.014$
	0.2	0.137 ± 0.021	0.136 ± 0.021	$\mathbf{0.112} \pm 0.014$	$\mathbf{0.112} \pm 0.014$
	0.4	0.196 ± 0.023	0.195 ± 0.024	$\mathbf{0.158} \pm 0.015$	$\mathbf{0.157} \pm 0.015$
	0.6	0.259 ± 0.029	0.260 ± 0.028	$\mathbf{0.208} \pm 0.018$	$\mathbf{0.208} \pm 0.017$
	0.8	0.325 ± 0.035	0.325 ± 0.033	$\mathbf{0.255} \pm 0.017$	$\mathbf{0.255} \pm 0.014$
	1.0	0.403 ± 0.043	0.401 ± 0.041	$\mathbf{0.306} \pm 0.019$	$\mathbf{0.307} \pm 0.018$
	1.2	0.484 ± 0.049	0.483 ± 0.048	$\mathbf{0.364} \pm 0.018$	$\mathbf{0.364} \pm 0.019$
	1.4	0.538 ± 0.046	0.537 ± 0.048	$\mathbf{0.423} \pm 0.019$	$\mathbf{0.422} \pm 0.021$

Table 3: Mean and standard deviation of relative ND from DeepAR models on different datasets under supplement of noisy observation with adversarial parameter ρ, measured over 10 independent runs. The * symbols for FS models indicate statistically significant improvement against the corresponding baselines according to the Wilcoxon signed-rank test.

(a) Exchange Rate

ρ	Vanilla	FS ($\sigma = 1.0$)	RT ($\sigma_{\text{tr}} = 0.1$)	RT + FS ($\sigma_{\text{tr}} = 0.1$, $\sigma = 1.0$)
-0.9	0.160 ± 0.003	$0.093 \pm 0.005^*$	$\mathbf{0.034} \pm 0.003$	$\mathbf{0.033} \pm 0.005$
-0.5	0.091 ± 0.001	$0.057 \pm 0.002^*$	$\mathbf{0.019} \pm 0.001$	0.021 ± 0.002
0	$\mathbf{0.003} \pm 0.000$	0.024 ± 0.000	0.012 ± 0.000	0.014 ± 0.000
0.5	0.082 ± 0.002	$0.057 \pm 0.001^*$	$\mathbf{0.019} \pm 0.002$	$\mathbf{0.020} \pm 0.001$
1.0	0.150 ± 0.007	$0.101 \pm 0.005^*$	0.036 ± 0.007	$\mathbf{0.034} \pm 0.005^*$
2.0	0.238 ± 0.019	$0.178 \pm 0.012^*$	0.073 ± 0.019	$\mathbf{0.063} \pm 0.012^*$
4.0	0.315 ± 0.030	$0.263 \pm 0.019^*$	0.125 ± 0.030	$\mathbf{0.109} \pm 0.019^*$
9.0	0.391 ± 0.034	$0.347 \pm 0.024^*$	0.190 ± 0.034	$\mathbf{0.178} \pm 0.024^*$

(b) M4-Daily

ρ	Vanilla	FS ($\sigma = 1.0$)	RT ($\sigma_{\text{tr}} = 0.1$)	RT + FS ($\sigma_{\text{tr}} = 0.1$, $\sigma = 1.0$)
-0.9	0.272 ± 0.012	$0.173 \pm 0.014^*$	$\mathbf{0.095} \pm 0.012$	0.098 ± 0.014
-0.5	0.157 ± 0.002	$0.099 \pm 0.007^*$	$\mathbf{0.047} \pm 0.002$	0.052 ± 0.007
0	$\mathbf{0.007} \pm 0.000$	0.037 ± 0.002	0.013 ± 0.000	0.021 ± 0.002
0.5	0.151 ± 0.002	$0.101 \pm 0.007^*$	$\mathbf{0.046} \pm 0.002$	0.052 ± 0.007
1.0	0.269 ± 0.008	$0.185 \pm 0.015^*$	$\mathbf{0.097} \pm 0.008$	0.099 ± 0.015
2.0	0.416 ± 0.035	$0.319 \pm 0.034^*$	0.215 ± 0.035	$\mathbf{0.196} \pm 0.034^*$
4.0	0.558 ± 0.073	$0.483 \pm 0.066^*$	0.405 ± 0.073	$\mathbf{0.363} \pm 0.066^*$
9.0	0.723 ± 0.090	$0.681 \pm 0.093^*$	0.619 ± 0.090	$\mathbf{0.597} \pm 0.093^*$

Table 3: *(Continued)* Mean and standard deviation of relative ND from DeepAR models on different datasets under supplement of noisy observation with adversarial parameter ρ, measured over 10 independent runs. The * symbols for FS models indicate statistically significant improvement against the corresponding baselines according to the Wilcoxon signed-rank test.

(c) Traffic

ρ	Vanilla	FS ($\sigma = 0.1$)	RT ($\sigma_{\text{tr}} = 0.1$)	RT + FS ($\sigma_{\text{tr}} = 0.1$, $\sigma = 0.1$)
-0.9	0.141 ± 0.008	0.139 ± 0.008	0.102 ± 0.008	$\mathbf{0.100} \pm 0.008^{*}$
-0.5	0.084 ± 0.003	0.085 ± 0.003	$\mathbf{0.058} \pm 0.003$	$\mathbf{0.058} \pm 0.003$
0	0.037 ± 0.001	0.039 ± 0.001	$\mathbf{0.034} \pm 0.001$	$\mathbf{0.033} \pm 0.001^{*}$
0.5	0.068 ± 0.002	$0.068 \pm 0.002^{*}$	0.048 ± 0.002	$\mathbf{0.046} \pm 0.002^{*}$
1.0	0.096 ± 0.002	$0.094 \pm 0.003^{*}$	0.065 ± 0.002	$\mathbf{0.063} \pm 0.003^{*}$
2.0	0.121 ± 0.004	$0.119 \pm 0.004^{*}$	0.089 ± 0.004	$\mathbf{0.086} \pm 0.004^{*}$
4.0	0.139 ± 0.007	$0.137 \pm 0.007^{*}$	0.109 ± 0.007	$\mathbf{0.106} \pm 0.007^{*}$
9.0	0.170 ± 0.010	$0.167 \pm 0.009^{*}$	0.140 ± 0.010	$\mathbf{0.137} \pm 0.009^{*}$

(d) Electricity

ρ	Vanilla	FS ($\sigma = 0.5$)	RT ($\sigma_{\text{tr}} = 0.1$)	RT + FS ($\sigma_{\text{tr}} = 0.1$, $\sigma = 0.5$)
-0.9	0.065 ± 0.001	0.078 ± 0.002	$\mathbf{0.034} \pm 0.001$	0.039 ± 0.002
-0.5	0.042 ± 0.001	0.061 ± 0.002	$\mathbf{0.025} \pm 0.001$	0.032 ± 0.002
0	$\mathbf{0.017} \pm 0.001$	0.044 ± 0.002	0.019 ± 0.001	0.025 ± 0.002
0.5	0.047 ± 0.001	0.058 ± 0.002	$\mathbf{0.025} \pm 0.001$	0.031 ± 0.002
1.0	0.088 ± 0.002	0.087 ± 0.003	$\mathbf{0.037} \pm 0.002$	0.041 ± 0.003
2.0	0.170 ± 0.004	$0.147 \pm 0.005^{*}$	$\mathbf{0.066} \pm 0.004$	0.069 ± 0.005
4.0	0.295 ± 0.018	$0.246 \pm 0.016^{*}$	0.136 ± 0.018	$\mathbf{0.134} \pm 0.016^{*}$
9.0	0.435 ± 0.055	$0.379 \pm 0.047^{*}$	0.291 ± 0.055	$\mathbf{0.276} \pm 0.047^{*}$