Loss Shaping Constraints for Long-Term Time Series Forecasting

Ignacio Hounie [*1] Javier Porras-Valenzuela [*1] Alejandro Ribeiro [1]

Abstract

Several applications in time series forecasting require predicting multiple steps ahead. Despite the vast amount of literature in the topic, both classical and recent deep learning based approaches have mostly focused on minimising performance *averaged* over the predicted window. We observe that this can lead to disparate distributions of errors across forecasting steps, especially for recent transformer architectures trained on popular forecasting benchmarks. That is, optimising performance on average can lead to undesirably large errors at specific time-steps. In this work, we present a Constrained Learning approach for long-term time series forecasting that aims to find the best model in terms of average performance that respects a user-defined upper bound on the loss at each time-step. We call our approach *loss shaping constraints* because it imposes constraints on the loss at each time step, and leverage recent duality results to show that despite its non-convexity, the resulting problem has a bounded duality gap. We propose a practical Primal-Dual algorithm to tackle it, and demonstrate that the proposed approach exhibits competitive average performance in time series forecasting benchmarks, while shaping the distribution of errors across the predicted window.

1. Introduction

Predicting multiple future values of time series data, also known as multi-step forecasting (Bontempi et al., 2013), has a myriad of applications such as forecasting weather (Wang et al., 2016), electricity demand (Yi et al., 2022), prices (Chen et al., 2018), and passenger demand (Bai et al., 2019). Several approaches to generating predictions for the next window have been proposed, including direct (Chevillon, 2007; Sorjamaa et al., 2007), autoregressive or recursive (Hill et al., 1996; Tiao & Tsay, 1994; Hamzaebi et al., 2009) and MIMO techniques (Kline, 2004; Bontempi, 2008; Bontempi & Ben Taieb, 2011; Kitaev et al., 2019; Zhou et al., 2021; Wu et al., 2021; Liu et al., 2021; Zhou et al., 2022b; Nie et al., 2022; Zeng et al., 2023; Garza & Mergenthaler-Canseco, 2023; Das et al., 2023b; Liu et al., 2024; 2022). Moreover, a plethora of learning parametrizations exist, ranging from linear models (Box & Jenkins, 1976; Zeng et al., 2023) to recent transformer (Kitaev et al., 2019; Zhou et al., 2021; Wu et al., 2021; Liu et al., 2021; 2024; 2022) and custom (Zhou et al., 2022a) architectures.

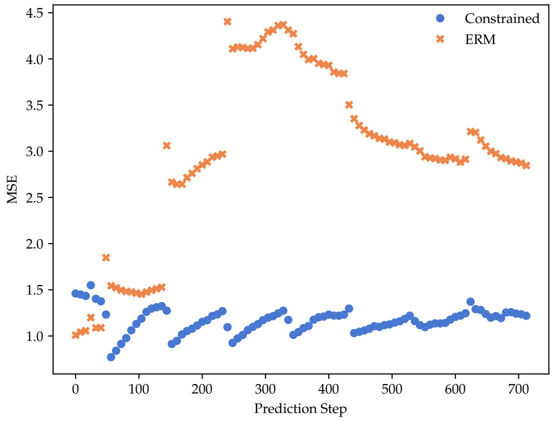

Figure 1. Test Mean Squared Error (MSE) computed at individual time steps on the forecasting window for Autoformer (Wu et al., 2021) on exchange rate data using ERM and our approach.

Regardless of the model and parametrization, most approaches optimize a performance, risk or model fit functional (usually MSE), *averaged* over the predicted window. Therefore, the distribution of errors across the window – without any additional assumptions – can vary depending on the model and data generating process. In practice, this can lead to uneven performance across the different steps of the window.

Recent works using transformer based architectures focus on aggregate metrics, see for example (Kitaev et al., 2019; Wu et al., 2021; Nie et al., 2022), while addressing errors at different time steps has recieved little attention (Cheng et al., 2023).

[*]Equal contribution [1]University of Pennsylvania. Correspondence to: Ignacio Hounie <ihounie@seas.upenn.edu>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

However, having no control on the error incurred at each time step in the predicted window can be detrimental in scenarios that can be affected by this variability. For example, analysing average behaviour is insufficient to assess financial risks in econometrics (Chavleishvili & Manganelli, 2019) or to ensure stability when using the predictor in a Model Predictive Control framework (Terzi et al., 2021).

Therefore, our first contribution is addressing this problem through constrained learning.

(C1) We formulate multi-step series forecasting as a constrained learning problem, which aims to find the best model in terms of *average performance* while imposing a user-defined upper bound on the loss at each time-step.

Since this constrains the distribution of the loss per time step across the window, we call this approach *loss shaping constraints*. Due to the challenges in finding appropriate constraints, we leverage a recent approach (Hounie et al., 2023) to re-interpret the specification as a soft constraint and find a minimal norm relaxation while jointly solving the learning task. We leverage recent duality results (Chamon & Ribeiro, 2020) to show that, despite being non-convex, the resulting problem has a bounded duality gap. This allows us to provide approximation guarantees at the step level.

We then propose an alternating Primal-Dual algorithm to tackle the loss shaping constrained problem. Our final contribution is the experimental evaluation of this algorithms in a practical setting:

(C2) We evaluate our algorithms for different constraints using state-of-the-art transformer architectures (Kitaev et al., 2019; Zhou et al., 2021; Wu et al., 2021) in popular forecasting benchmarks.

Our empirical results showcase the ability to alter the shape of the loss by introducing constraints, and how this can lead to better performance in terms of standard deviation across the predictive window.

1.1. Related work

There is little prior work in time series prediction on promoting or imposing a certain distribution of errors *across* the predicted window, i.e., at specific timesteps. Works that propose reweighting the errors at different steps of the window mostly aim to improve average performance by leveraging the structure of residuals (Chevillon, 2007; Guo et al., 1999; Hansen, 2010) and rely on properties of the data generating process and predictive model class. On the other hand, works that address the empirical distribution of forecasting errors and robustness in non-parametric models (Spiliotis et al., 2019; Taleb, 2009) do not analyse the distribution of errors *across* multiple steps. Similarly, Multiple output Support Vector Regression for multi-step forecasting (Bao et al., 2014), which also tackles a constrained problem, has only addressed aggregate errors across the whole window.

Although our method is related to other loss shaping approaches (Park et al., 2023; Wang et al., 2024; Cheng et al., 2023) in the sense that it alters the weighting given to each time step on the loss, both the motivation and the problems that these approaches address differ from our work. (Park et al., 2023) aim to downweight the impact of large errors as long as they have low probability, regardless of their position in the predictive window. Our approach aims to control average errors at each step and can thus go in the opposite direction of sacrificing performance in order to reduce undesirably large errors at specific steps. (Wang et al., 2024) gives each time-step a fixed weighting that is inversely proportional to their position in the window, thus aiming to discount the impact of far-future errors and give a higher weighting to closer steps. However, imposing a fixed weighting does not take into account the difficulty of fitting an error at a certain time-step, which will depend also on the model and the data. That is, imposing a certain weighting across the window *need not* result in the desired distribution of errors.

Lastly, recent works in generative time series models have also sought to impose constraints using penalty based methods (Coletta et al., 2023), but the nature of the constraints and proposed approach also differs from ours.

2. Multi-Step Time Series Forecasting

Let $\mathbf{x}_t \in \mathcal{X} \subseteq \mathbb{R}^{d_x}$ denote the in feature vector and $y_t \in \mathcal{Y} \subseteq \mathbb{R}^{d_y}$ its associated output or measurement at time step t. The goal in multi-step time series forecasting is to predict T_p future values of the output, i.e., $y_p := y_{[t+1:t+T_p]}$ given a window of length T_c of input features, i.e., $\mathbf{x}_c := \mathbf{x}_{[t-T_c:t]}$. The quality of the forecasts can be evaluated at each time-step using a non-negative loss function or metric $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$, e.g., the squared error or absolute difference.

The most common approach in supervised multi-step forecasting is to learn a predictor that minimizes the expected loss *averaged over the predicted window*:

$$\min_{\boldsymbol{\theta}\in\Theta} \mathbb{E}_{(\mathbf{x}_c, y_p)\sim\mathfrak{D}} \left[\frac{1}{T_p} \sum_{i=1}^{T_p} \ell_i(f_{\boldsymbol{\theta}}(\mathbf{x}_c), y_p) \right], \qquad \text{(ERM)}$$

where $\ell_i := \ell([f_{\boldsymbol{\theta}}(\mathbf{x}_c)]_i, [y_p]_i)$ denotes the loss evaluated at the i-th time step, $\mathfrak{D}$ a probability distribution over data pairs[1] $(\mathbf{x}_c, y_p)$ and $f_{\boldsymbol{\theta}} : \mathcal{X}^{T_c} \to \mathcal{Y}^{T_p}$ is a predictor associ-

[1] Our approach is distribution agnostic, i.e. we impose no additional structure or assumptions on $\mathfrak{D}$.

ated with parameters $\boldsymbol{\theta} \in \Theta \subset \mathbb{R}^k$, for example, a transformer based neural network architecture.

However, this choice of objective does not account for the *structure* or distribution of the errors across different time steps, which can lead to disparate behaviour across the predicted window as depicted in Figure 1. In particular, we observe empirically that state-of-the-art transformer architectures (Kitaev et al., 2019; Zhou et al., 2021; Wu et al., 2021; Liu et al., 2021; Zhou et al., 2022a) can yield highly varying loss dynamics, including non-monotonic, flat and highly non-linear landscapes, as presented in Appendix D

In order to control or promote a desirable loss pattern, a weighted average across time steps could be employed. However, since the realised losses will depend not only on the data distribution but also on the model class and learning algorithm. As a result, such penalization coefficients would have to be tuned in order to achieve a desired loss pattern, which in contrast can be naturally expressed as a requirement, as presented next.

3. Loss Shaping Constraints

In order to control the shape of the loss over the time window, we require the loss on timestep i to be smaller than some quantity $\boldsymbol{\epsilon}_i$. This leads to the constrained statistical learning problem:

$$
\begin{aligned}
P^\star(\boldsymbol{\epsilon}) = &\min_{\boldsymbol{\theta}\in\Theta} \; \mathbb{E}_{(\mathbf{x}_c, y_p)\sim\mathfrak{D}} \left[\frac{1}{T_p} \sum_{i=1}^{T_p} \ell_i(f_{\boldsymbol{\theta}}(\mathbf{x}_c), y_p) \right] \\
&\text{s. to}: \quad \mathbb{E}_{(\mathbf{x}_c, y_p)\sim\mathfrak{D}} \left[\ell_i(f_{\boldsymbol{\theta}}(\mathbf{x}_c), y_p) \right] \leq \boldsymbol{\epsilon}_i \\
&\qquad i = 1, \ldots, T_p. \qquad\qquad \text{(P-LS)}
\end{aligned}
$$

An advantage of (P-LS) is that it is interpretable in the sense that constraints – unlike penalty coefficients – make explicit the requirement that they represent. That is, the constraint is expressed of terms of the *expected* loss at individual timesteps, and thus prior knowledge about and the underlying data distribution and model class can be exploited.

In Section 5, we focus on simply imposing a constant upper bound for all time steps, which we set using the performance of the (unconstrained) ERM solution. By upper bounding the error, we prevent errors from being undesirably large irrespectively of their position in the window, effectively limiting the spread of the errors across the whole window. Since we still minimize the mean error across the window, this approach is not as conservative as minimax formulations (Liu & Taniguchi, 2021), which only focus on the worst error.

It is worth pointing out that many other constraint choices are possible. For instance, it is often reasonable to assume errors increase monotonically along the prediction window. In this case, $\boldsymbol{\epsilon}_i$ could take increasing values based on prior knowledge of the learning problem. We provide further discussions about such configurations in Appendix B.1.

Nonetheless, which loss patterns are desirable and attainable will depend ultimately on the model, data and task at hand. In the next section we explore how to *automatically* adapt constraints during training, so that the problem is more robust to their mis-specification.

3.1. Adapting Constraints: Resilient Constrained Learning

In the unconstrained risk minimization problem (ERM) an optimal function $\theta^\star$ always exists. This is not the case of the constrained learning problem in (P-LS) in which there may be no parameters in Θ that satisfy all the requirements if the target losses ϵ_i are too restrictive.

In practice, landing on satisfiable loss shaping requirements may necessitate the relaxation of some constraints –i.e., the values of $\boldsymbol{\epsilon}_i$ in (P-LS)–. This is challenging because assessing the impact of tightening or relaxing a particular constraint can have intricate dependencies with the model class, the unknown data distribution, and learning algorithm, which are hard to determine a priori.

Thus, the point at issue is coming up with reasonable constraints that achieve a desirable trade-off between controlling the distribution of the error across the predictive window and attaining good average performance. That is, for larger constraint levels the average performance improves, although the constraints will have less impact on the optimal function $f_{\theta^\star}$.

In order to do so, we introduce a non negative perturbation $\boldsymbol{\zeta}_t \in \mathbb{R}_+$ associated to each time step, and consider relaxing the $i - th$ constraint in the original problem by $\boldsymbol{\zeta}_i$. Explicitly, we impose $\mathbb{E}_{(\mathbf{x}_c, y_p)\sim\mathfrak{D}} \left[\ell_i(f_{\boldsymbol{\theta}}(\mathbf{x}_c), y_p) \right] \leq \boldsymbol{\epsilon}_i + \boldsymbol{\zeta}_i$. We also introduce a differentiable, convex, non-decreasing cost $h : \mathbb{R}_+^{T_p} \to \mathbb{R}_+$, that penalizes deviating from the original specification, e.g., the squared L_2 norm $h(\boldsymbol{\zeta}) \propto \|\boldsymbol{\zeta}\|_2^2$.Thus, we seek a relaxation $\boldsymbol{\zeta}^\star$ that achieves a desirable trade-off by equating marginal decrease in the objective and the marginal increase in relaxation cost. Explicitly,

$$
-\partial h(\boldsymbol{\zeta}^\star) \in \partial P^\star(\boldsymbol{\epsilon} + \boldsymbol{\zeta}^\star), \qquad (1)
$$

where $\partial P^\star$ denotes the subdifferential of $P^\star$.

This allows us to re-interpret the initial constraint levels $\boldsymbol{\epsilon}$ as a soft constraint, and learn a relaxation that makes the problem easier to solve. As shown by (Hounie et al., 2023) this relaxation can be found while jointly solving the learning task by solving the problem

$$\min_{\boldsymbol{\zeta}\in\mathbb{R}^{T_p}} P^{\star}(\boldsymbol{\epsilon}+\boldsymbol{\zeta})+h(\boldsymbol{\zeta})\ . \tag{R-LS}$$

Further discussion on equivalent formulations that provide a straightforward way to compute the resilient relaxation are discussed in Appendix B.2. In the next section, we introduce practical primal-dual algorithms that enable approximating the constrained (P-LS) and resilient (R-LS) problems based on samples and finite (possibly non-convex) parametrizations such as transformer architectures.

4. Empirical Dual Resilient and Constrained Learning

A challenge in solving (P-LS) and (R-LS) is that in general, (i) no closed form projection into the feasible set or proximal operator exists, and (ii) it involves an unknown data distribution $\mathfrak{D}$. In what follows, we describe our approach for tackling the Resilient problem, noting that Primal-Dual algorithms tackling the original constrained problem can be similarly derived by simply excluding the slack variables.

To undertake R-LS, (i) we replace expectations by sample means over a data set $\{(\boldsymbol{x}_c^n, y_p^n)\ :\ n=1,\cdots,N\}$, as typically done in (unconstrained) statistical learning, and (ii) resort to its Lagrangian dual.

These modifications lead to the Empirical Dual problem

$$\hat{D}^{\star}=\max_{\boldsymbol{\lambda}\geq 0}\quad \min_{\boldsymbol{\theta}\in\Theta,\ \boldsymbol{\zeta}\in\mathbb{R}_{+}^{T_p}} \hat{L}(\boldsymbol{\theta},\boldsymbol{\lambda},\boldsymbol{\epsilon},\boldsymbol{\zeta}), \tag{ED-LS}$$

where $\hat{L}$ is the empirical Lagrangian of (R-LS), defined as

$$\hat{L}(\boldsymbol{\theta},\boldsymbol{\lambda},\boldsymbol{\epsilon},\boldsymbol{\zeta}) := \frac{1}{2}h(\boldsymbol{\zeta})+$$

$$\frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{T_p}\left(\boldsymbol{\lambda}_i+\frac{1}{T_p}\right)\left[\ell_i([f_{\boldsymbol{\theta}}(\mathbf{x}_c^n)],y_p^n)\right]-\boldsymbol{\lambda}_i(\boldsymbol{\epsilon}_i+\boldsymbol{\zeta}_i),$$

and $\boldsymbol{\lambda}_i$ is the dual variable associated to the constraint on the i-th time step loss ℓ_i.

The Empirical Dual problem (ED-LS) is an approximation of the Dual problem associated to R-LS based on training samples. The dual problem itself can be interpreted as finding the tightest lower bound on the primal. Although the estimation of expectations using sample means and non-convexity of the hypothesis class can introduce a duality gap, i.e., $\hat{D}^{\star} < P^{\star}$, under certain conditions this gap can be bounded.

Unlike unconstrained statistical learning bounds, these approximation bounds depend not only in the sample complexity of the model class and loss but also on the optimal dual variables or slacks associated to the constraint problem. These reflect how challenging it is to meet the constraints. This aspect is crucial, as applying overly stringent loss shaping constraints in a given learning setting may be detrimental in terms of approximation and lead to sub-optimal test performance. We include a summary of these results as well as a discussion on their implications in this setting in Appendix A, and refer to (Chamon & Ribeiro, 2020; Hounie et al., 2023) for further details.

The advantage of tackling the the empirical dual problem $\hat{D}^{\star}$ is that it can be solved using saddle point methods presented in the next section.

4.1. Algorithm

In order to solve problem (ED-LS), we resort to dual ascent methods, which can be shown to converge even if the inner minimization problem is non-convex (Chamon et al., 2022). The saddle point problem (ED-LS) can then be undertaken by alternating the minimization with respect to $\boldsymbol{\theta}$ and $\boldsymbol{\zeta}$ with the maximization with respect to $\boldsymbol{\lambda}$ (K. J. Arrow & Uzawa, 1960), which leads to the Primal-Dual constrained learning procedure in Algorithm 1.

Although a bounded empirical duality gap *does not* guarantee that the primal variables obtained after running Algorithm 1 are near optimal or approximately feasible *in general*, recent constrained learning literature provides sub-optimality and near-feasibility bounds for primal iterates (Elenter et al., 2024) as well as abundant empirical evidence (Robey et al., 2021; Gallego-Posada et al., 2022; Elenter et al., 2022) that good solutions can still be obtained.

Algorithm 1 **P**rimal **D**ual **L**oss **S**haping.

```
Input: Dataset $\{x_i, y_i\}_{i=1,\cdots,N}$, primal learning rate $\eta_p$, dual learning rate $\eta_d$, perturbation learning rate $\eta_\epsilon$, number of epochs $T_e$, number of batches $T_b$, initial constraint tightness $\epsilon_{\alpha_0}$.
Initialize: $\boldsymbol{\theta}, \lambda_1, \ldots, \lambda_{T_{\text{pred}}} \leftarrow \mathbf{0}$
for epoch $= 1, \ldots, T_e$ do
  for batch $= 1, \ldots, T_b$ do
      Update primal variables
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta_p \nabla_\theta \hat{L}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\epsilon}, \boldsymbol{\zeta})$
      Evaluate constraints.
    $\mathbf{s}_i \leftarrow \left(\frac{1}{N_b}\sum_{n=1}^{N_b} \ell_i([f_{\boldsymbol{\theta}}(\mathbf{x}^n_{\mathrm{c}(t)})], y^n_{\mathrm{p}(t)})\right) - (\epsilon_i + \boldsymbol{\zeta}_i)$
      Update slacks.
    $\boldsymbol{\zeta} \leftarrow \left[\boldsymbol{\zeta} - \eta_\zeta\left(\nabla h(\boldsymbol{\zeta}) - \boldsymbol{\lambda}\right)\right]_+$
      Update dual variables.
    $\boldsymbol{\lambda} \leftarrow [\lambda + \eta_d \mathbf{s}]_+$
  end for
end for
Return: $\theta$, $\boldsymbol{\lambda}$, $\boldsymbol{\zeta}$.
```

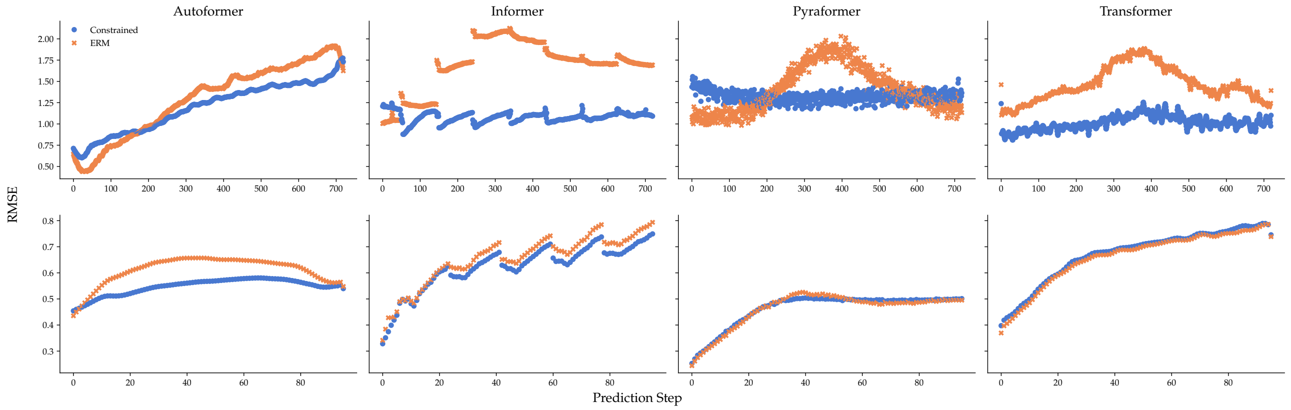

Figure 2. Test MSE at each prediction step for two datasets across different models, datasets and predictive windows. The top row shows results for the Weather dataset with a predictive window length of 96 steps, and the second row corresponds to the Exchange Rate dataset with a predictive window length of 720 steps. Each column corresponds to a different architecture, and each curve represents a different training algorithm, we include both the ERM baseline and our method using a constant constraint across the window for all models.

5. Experiments

We conduct extensive evaluations comparing constrained and resilient constrained learning against the customary unconstrained training pipeline across eight model architectures and nine popular datasets. For each dataset, we train models with four different predictive window lengths to evaluate the impact of constraints as the forecasting horizons extended. In total, this amounts to 288 different experiment settings.

Explicitly, the datasets are: Electricity Consumption Load (ECL), Weather, Exchange Rate (Lai et al., 2018), Traffic, Electricity Transformer Temperature (ETT) (two hourly datasets and two every 15 minutes) (Zhou et al., 2021), and Influenza Like Illness (ILI). Following the literature, we train with predictive window lengths of 96, 192, 320, and 720 for each dataset, except for Illness, which is trained to predict with lengths of 24, 36, 48, and 60. For a more detailed explanation of the datasets' contents and source, refer to the Appendix C.1.

We also include a wide variety of time series prediction models comprising seven transformer-based architectures and one non-transformer architecture. Namely, the transformer models are: Reformer (Kitaev et al., 2019), Autoformer (Wu et al., 2021), Informer (Zhou et al., 2021), Pyraformer (Liu et al., 2021), iTransformer (Liu et al., 2024), Nonstationary Transformer (Liu et al., 2022) and a vanilla transformer architecture (Vaswani et al., 2017). The non-transformer model is FiLM (Zhou et al., 2022a).

We follow the same setup, including preprocessing, hyperparameters and implementation, as described in (Kitaev et al., 2019; Wu et al., 2021; Liu et al., 2021; Zhou et al., 2021; Liu et al., 2022; Zhou et al., 2022a). For our method, we only perform a grid search over six values of the constraint level ϵ. The constrained and resilient loss shaping results presented in this section correspond to the best performing constraint level. We provide an additional ablation analysis on one dataset in Appendix D.5.

Data is split into train, validation, and test chronologically with a ratio of 7:1:2. For each data split we extract every pair of consecutive context and prediction windows of length T_c and T_p. That is, we use rolling (overlapping) windows for both training and testing. [2] Additional experiment details can be found on Appendix C.

5.1. Loss Shaping

In this section, we demonstrate that our approach effectively reduces performance fluctuations across the window while maintaining comparable average performance. We compute the mean squared errors (MSE) at each predicted timestep and report both the average MSE and the standard deviation across the window, referred to as Window STD.

To ease interpretation and facilitate comparisons across different experimental settings, we normalize errors using the unconstrained ERM baseline. For complete result tables with unnormalized MSE values and additional example cases, refer to Appendix D.

We do not to use constraint violation as a metric because it can be uninformative when the loss landscape is consistently infeasible due to a large generalization gap and constraint levels are constant. In such cases, constraint violation is as informative as MSE, even when the error distribution over

[2] We also fix a known bug (reported in (Das et al., 2023a)) from (Wu et al., 2021) and followup works, where the last T_p samples where discarded.

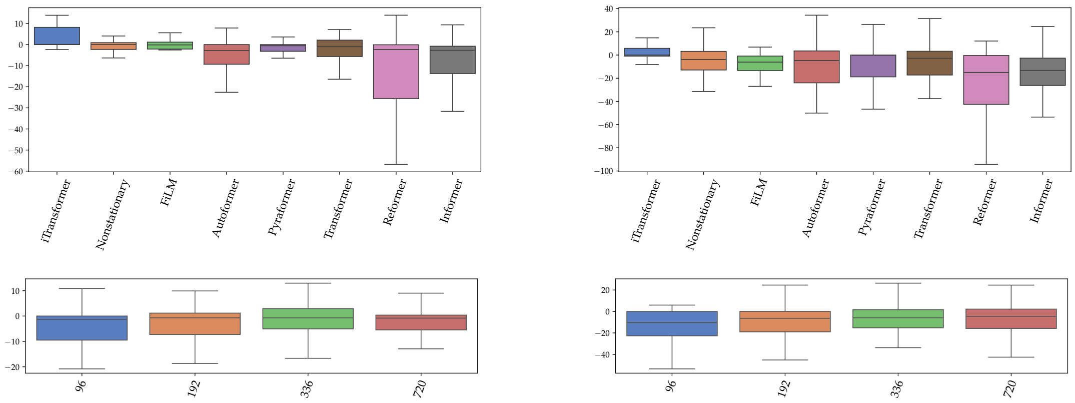

Figure 3. Box plots of percentual changes across all experiments. The left column contains plots of MSE across experiments, and the right column is Window STD. We segment experiments by model and prediction length. The x axes of each box plot are sorted by the mean ERM MSE (better models and easier datasets to the left). The full table with the results for each ERM, constrained and resilient setting can be found in Appendix D.

time changes.

Due to the large number of experimental setups, we summarize quantitative results in Figure 4, which illustrates the relative differences in MSE and Window STD between ERM and constrained runs. It shows that loss shaping constrained models typically achieve reductions in Window STD while preserving or even improving average MSE.

We then analyze how the performance of our method varies across setups, by grouping results by prediction length, model architecture and dataset. First, it varies significantly across models, as shown in the second row of Figure 3. We conjecture that this variability can be attributed to the inductive biases of certain architectures, which makes it harder to impose the desired loss landscape. Second, the relative effect of our method on both MSE and Window STD remains largely unchanged as the prediction window increases, as shown in the third row of box plots in Figure 3. Since both average MSE and Window STD increase with prediction length, a constant relative effect means that the absolute change *increases* with prediction length. Appendix D.1 shows several concrete examples of this phenomenon.

To qualitatively assess the impact of our method on the error distribution, we include plots of per time-step losses comparing models trained with ERM and our method in Figure 2. In these settings, our approach effectively affects the distribution of losses across the prediction windows. For example, in the Weather dataset (first row), Autoformer and Pyraformer are trading off overall MSE for a flatter error across the window. In other cases, like the Exchange Rate dataset for the Informer and Transformer models, in addition to showing flatter landscapes, the constrained models perform better than their ERM counterparts overall.

While we observe that in many cases imposing constraints is beneficial, we also distinguish two failure modes where we are unable to change test loss as desired. The first is when constraints are not feasible at the end of training, as seen in the first row of Figure 5. This is the motivation behind the resilient approach, presented in the next section. A second failure mode is due to the inherent generalization gap, which results in feasible constraints during training, but no effective loss shaping in test data, such as in the second row of Figure 5. We defer to Appendix D.1 for more in-depth comparisons of ERM and constrained runs.

Another advantage of our primal-dual approach is that multipliers indicate the difficulty of satisfying each constraint. Therefore, we can examine the impact of constraints on loss landscapes on a run by analyzing the loss distributions and multipliers in Figure 6. In this experiment, the training loss increases approximately monotonically over the prediction window. The final solution is infeasible, as the MSE constraints are violated beginning at step 341. Consequently, the optimal multipliers are also high, reflecting the difficulty of satisfying the constraints in the latter part of the window. Despite the growth of the multipliers, the model still fails to satisfy the constraints. This infeasibility during training results in ineffective loss shaping during testing.

5.2. Resilience

Resilient constrained learning can effectively overcome feasibility issues in training –like those mentioned in the previous section– by relaxing the most difficult constraints. As shown in Figure 7, when comparing hard constraints with resilient constraints using the same setting as the previous section, the resilient approach not only leads to smaller

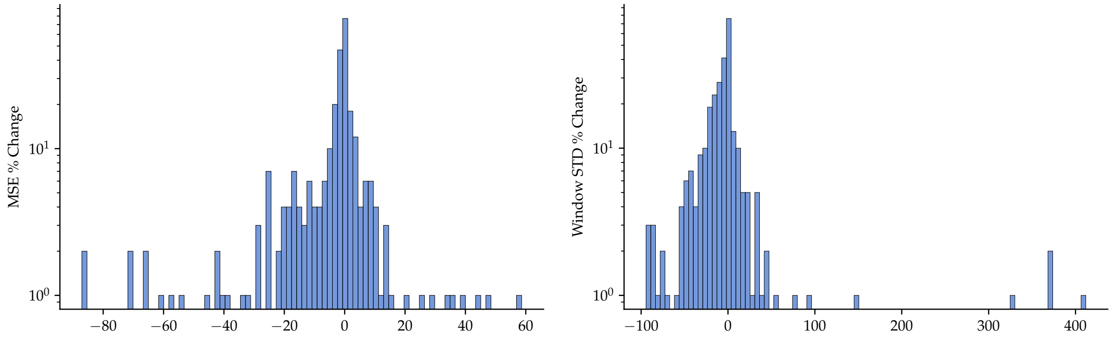

Figure 4. Distribution of percentual change in test MSE and Window STD across experiment settings when comparing a run of ERM with a constrained run. On average, STD decreases by 3.47% and MSE by 4.47% respectively.

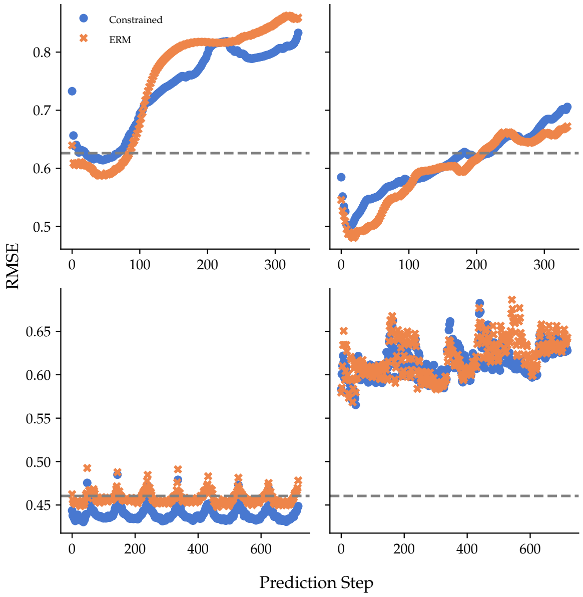

Figure 5. Two instances of failure cases. The first row is the training and testing errors of Autoformer on Weather data with window length of 336. The second row is Informer on ECL data with window length of 720. The gray lines are the values of ϵ used during training.

multipliers by relaxing some constraints, but also yields a solution that is more feasible overall.

In addition, we find that the solutions achieved by the resilient approach can also have better generalization when compared to hard constraints, as seen in Figure 8. The resilient model reduces the peak around step 150. This illustrates how resilient learning effectively mitigates the failure mode mentioned in the previous section.

As a result of relaxing constraints and solving feasibility issues during training, resilient models are effective in reducing Window STD in a wide variety of settings. This is supported by our empirical results in Tables 6 and 7. For instance, resilient FiLM achieves the lowest Window STD when compared to constrained and ERM-trained models. Models such as Autoformer, Pyraformer and iTransformer also achieve its lowest Window STD when using Resilient constraints.

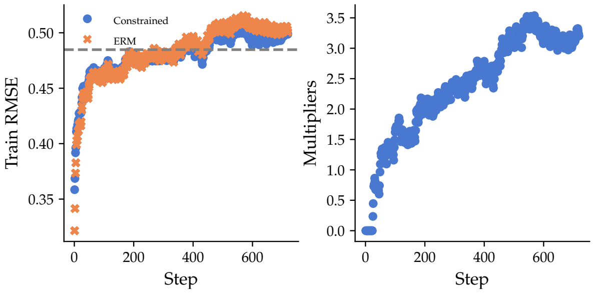

Figure 6. Training MSE and multipliers of a constrained model, with the ERM curve for reference. The setting is an Autoformer model for the Exchange Rate dataset, with a predictive window length of 720 steps. The gray line is the constraint, with a value of $\sqrt{\epsilon} = 0.484$.

Furthermore, we find that some settings also achieve its lowest MSE when using resilient constraints. Such is the case of iTransformer on datasets ETTh2,ETTm1 and ETTm2, where the resilient model has the best overall error in almost all predictive window lengths, as shown in the full MSE results, Tables 4 and 5 in Appendix D.

5.3. Monotonically Increasing Constraints

The results of the previous sections all use ϵ_i set to a constant value derived from the statistics of training or validation ERM errors. As discussed in Section 3, an alternative is to have the constraints take monotonically increasing values, for instance, from a linear or exponential fit. Having increasingly looser constraints is only logical in the common scenario where errors are expected to grow as we predict

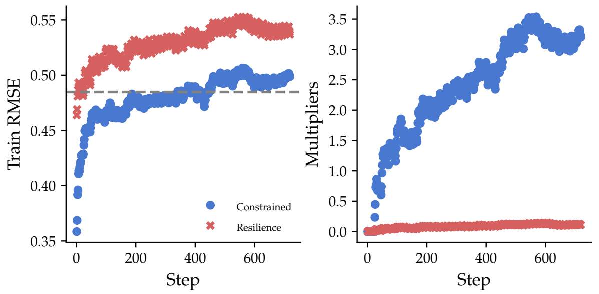

Figure 7. Training MSE and multipliers of constrained and resilient models. Resilience finds a feasible solution by relaxing the harder constraints. The setting is the same as in Figure 6.

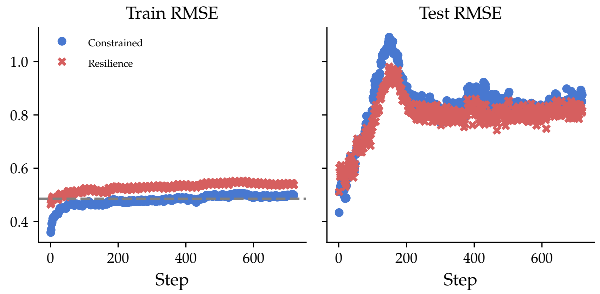

Figure 8. Train and test MSE of constrained and resilient models. By finding a feasible solution in training, generalization improves. The setting is the same as in Figure 6.

farther into the future. Note that we call these constraints exponential because they were set from an exponential fit, but they are linear in the optimization variable.

An empirical evaluation on the ECL dataset with three different models shows this is a viable constraint design. Table 1 contains the MSE for each of the three models: in two cases, exponential achieves the lowest MSE across all windows. With respect to Window STD, Table 2 shows that exponential does not always find the most stable shape. However, this is expected, because the monotonically increasing constraint will not necessarily enforce a reduction in variance across the window.

6. Conclusion

This paper introduced a time series forecasting constrained learning framework that aims to find the best model in terms of average performance while imposing a user-defined upper bound on the loss at each time-step. Given that we observed that the distribution of loss landscapes varies considerably, we explored a resilient constrained learning approach to dynamically adapt individual constraints during training. We analyzed the properties of this problem by leveraging recent duality results and developed practical algorithms to tackle it. We empirically corroborated that our approach can effectively alter the loss distribution across the forecast window.

Although we have focused on transformers for long-term time series forecasting, motivated by the empirical finding that their loss varies considerably, the loss shaping constrained learning framework can be extended to other settings. This includes a plethora of other models, datasets and tasks where losses are distributed across different time steps or grids. Furthermore, in this work, we focused on constraints characterized by a constant value across all timesteps, but other types of constraints, such as exponentially increasing stepwise constraints that we briefly presented, could also be explored in the future.

		Nonstationary Transformer			Pyraformer			Reformer		
		ERM	Constant	Exp	ERM	Constant	Exp	ERM	Constant	Exp
ECL	96	0.178	0.182	**0.170**	0.287	0.278	**0.274**	0.299	**0.297**	0.298
	192	0.186	0.191	**0.180**	0.294	0.292	**0.289**	0.332	0.326	**0.324**
	336	0.204	0.204	**0.196**	0.313	0.302	**0.295**	**0.359**	0.365	0.364
	720	0.233	0.229	**0.228**	0.310	0.298	**0.296**	**0.316**	0.316	0.318

Table 1. Test MSE of ERM, constant-constrained and exponentially-constrained settings in the ECL dataset, for three model architectures

		Nonstationary Transformer			Pyraformer			Reformer		
		ERM	Constant	Exp	ERM	Constant	Exp	ERM	Constant	Exp
ECL	96	0.017	**0.014**	0.016	0.019	**0.015**	**0.015**	**0.011**	0.011	0.012
	192	0.026	**0.019**	0.020	0.022	**0.017**	0.022	0.020	0.020	**0.018**
	336	0.031	**0.024**	0.025	0.034	**0.023**	0.028	**0.022**	0.023	0.023
	720	0.040	**0.037**	0.038	**0.015**	0.016	0.016	**0.007**	**0.007**	**0.007**

Table 2. Window STD of ERM, constant-constrained and exponentially-constrained settings in the ECL dataset, for three model architectures.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Bai, L., Yao, L., Kanhere, S., Wang, X., Sheng, Q., et al. Stg2seq: Spatial-temporal graph to sequence model for multi-step passenger demand forecasting. *arXiv preprint arXiv:1905.10069*, 2019.

Bao, Y., Xiong, T., and Hu, Z. Multi-step-ahead time series prediction using multiple-output support vector regression. *Neurocomputing*, 129:482–493, 2014.

Bontempi, G. Long term time series prediction with multi-input multi-output local learning. 2008. URL https://api.semanticscholar.org/CorpusID:137250.

Bontempi, G. and Ben Taieb, S. Conditionally dependent strategies for multiple-step-ahead prediction in local learning. *International Journal of Forecasting*, 27:689–699, 07 2011. doi: 10.1016/j.ijforecast.2010.09.004.

Bontempi, G., Ben Taieb, S., and Le Borgne, Y.-A. Machine learning strategies for time series forecasting. *Business Intelligence: Second European Summer School, eBISS 2012, Brussels, Belgium, July 15-21, 2012, Tutorial Lectures 2*, pp. 62–77, 2013.

Box, G. E. and Jenkins, G. M. Time series analysis. forecasting and control. *Holden-Day Series in Time Series Analysis*, 1976.

Boyd, S., Boyd, S. P., and Vandenberghe, L. *Convex optimization*. Cambridge university press, 2004.

Chamon, L. and Ribeiro, A. Probably approximately correct constrained learning. *Advances in Neural Information Processing Systems*, 33:16722–16735, 2020.

Chamon, L. F., Paternain, S., Calvo-Fullana, M., and Ribeiro, A. Constrained learning with non-convex losses. *IEEE Transactions on Information Theory*, 69(3):1739–1760, 2022.

Chavleishvili, S. and Manganelli, S. Forecasting and stress testing with quantile vector autoregression. *Available at SSRN 3489065*, 2019.

Chen, Y., Zhang, C., He, K., and Zheng, A. Multi-step-ahead crude oil price forecasting using a hybrid grey wave model. *Physica A: Statistical Mechanics and its Applications*, 501:98–110, 2018.

Cheng, J., Huang, K., and Zheng, Z. Fitting imbalanced uncertainties in multi-output time series forecasting. *ACM Transactions on Knowledge Discovery from Data*, 17(7): 1–23, 2023.

Chevillon, G. Direct multi-step estimation and forecasting. *Journal of Economic Surveys*, 21(4):746–785, 2007.

Coletta, A., Gopalakrishan, S., Borrajo, D., and Vyetrenko, S. On the constrained time-series generation problem. *arXiv preprint arXiv:2307.01717*, 2023.

Das, A., Kong, W., Leach, A., Sen, R., and Yu, R. Long-term forecasting with tide: Time-series dense encoder. *arXiv preprint arXiv:2304.08424*, 2023a.

Das, A., Kong, W., Sen, R., and Zhou, Y. A decoder-only foundation model for time-series forecasting, 2023b.

Elenter, J., Naderializadeh, N., and Ribeiro, A. A lagrangian duality approach to active learning. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 37575–37589. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/f475bdd151d8b5fa01215aeda925e75c-Paper-Conference.pdf.

Elenter, J., Chamon, L., and Ribeiro, A. Near-optimal solutions of constrained learning problems. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=fDaLmkdSKU.

Gallego-Posada, J., Ramirez, J., Erraqabi, A., Bengio, Y., and Lacoste-Julien, S. Controlled sparsity via constrained optimization or: How i learned to stop tuning penalties and love constraints. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 1253–1266. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/089b592cccfafdca8e0178e85b609f19-Paper-Conference.pdf.

Garza, A. and Mergenthaler-Canseco, M. Timegpt-1. *arXiv preprint arXiv:2310.03589*, 2023.

Guo, M., Bai, Z., and An, H. Z. Multi-step prediction for nonlinear autoregressive models based on empirical distributions. *Statistica Sinica*, 9(2):559–570, 1999. ISSN 10170405, 19968507. URL http://www.jstor.org/stable/24306598.

Hamzaebi, C., Akay, D., and Kutay, F. Comparison of direct and iterative artificial neural network forecast approaches in multi-periodic time series forecasting. *Expert systems with applications*, 36(2):3839–3844, 2009.

Hansen, B. E. Multi-step forecast model selection. In *20th Annual Meetings of the Midwest Econometrics Group*, 2010.

Hill, T., O'Connor, M., and Remus, W. Neural network models for time series forecasts. *Management science*, 42(7):1082–1092, 1996.

Hornik, K., Stinchcombe, M., and White, H. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Hounie, I., Ribeiro, A., and Chamon, L. F. O. Resilient constrained learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=h0RVoZuUl6.

K. J. Arrow, L. H. and Uzawa, H. Studies in linear and non-linear programming, by k. j. arrow, l. hurwicz and h. uzawa. stanford university press, 1958. 229 pages. *Canadian Mathematical Bulletin*, 3(3):196–198, 1960. doi: 10.1017/S0008439500025522.

Kitaev, N., Kaiser, L., and Levskaya, A. Reformer: The efficient transformer. In *International Conference on Learning Representations*, 2019.

Kline, D. M. Methods for multi-step time series forecasting neural networks. In *Neural networks in business forecasting*, pp. 226–250. IGI Global, 2004.

Lai, G., Chang, W.-C., Yang, Y., and Liu, H. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pp. 95–104, 2018.

Liu, S., Yu, H., Liao, C., Li, J., Lin, W., Liu, A. X., and Dustdar, S. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International conference on learning representations*, 2021.

Liu, Y. and Taniguchi, M. Minimax estimation for time series models. *METRON*, 79(3):353–359, 2021. doi: 10.1007/s40300-021-00217-6. URL https://doi.org/10.1007/s40300-021-00217-6.

Liu, Y., Wu, H., Wang, J., and Long, M. Non-stationary transformers: Exploring the stationarity in time series forecasting. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 9881–9893. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/4054556fcaa934b0bf76da52cf4f92cb-Paper-Conference.pdf.

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

Mohri, M., Rostamizadeh, A., and Talwalkar, A. *Foundations of machine learning*. MIT press, 2018.

Nie, Y., Nguyen, N. H., Sinthong, P., and Kalagnanam, J. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.

Park, J., Lee, J., Cho, Y., Shin, W., Kim, D., Choo, J., and Choi, E. Deep imbalanced time-series forecasting vianbsp;local discrepancy density. In *Machine Learning and Knowledge Discovery in Databases: Research Track: European Conference, ECML PKDD 2023, Turin, Italy, September 18–22, 2023, Proceedings, Part V*, pp. 139–155, Berlin, Heidelberg, 2023. Springer-Verlag. ISBN 978-3-031-43423-5. doi: 10.1007/978-3-031-43424-2_9. URL https://doi.org/10.1007/978-3-031-43424-2_9.

Robey, A., Chamon, L., Pappas, G. J., Hassani, H., and Ribeiro, A. Adversarial robustness with semi-infinite constrained learning. *Advances in Neural Information Processing Systems*, 34:6198–6215, 2021.

Sorjamaa, A., Hao, J., Reyhani, N., Ji, Y., and Lendasse, A. Methodology for long-term prediction of time series. *Neurocomputing*, 70(16-18):2861–2869, 2007.

Spiliotis, E., Nikolopoulos, K., and Assimakopoulos, V. Tales from tails: On the empirical distributions of forecasting errors and their implication to risk. *International Journal of Forecasting*, 35(2):687–698, 2019. ISSN 0169-2070. doi: https://doi.org/10.1016/j.ijforecast.2018.10.004. URL https://www.sciencedirect.com/science/article/pii/S0169207018301547.

Taleb, N. N. Errors, robustness, and the fourth quadrant. *International Journal of Forecasting*, 25(4):744–759, 2009. ISSN 0169-2070. doi: https://doi.org/10.1016/j.ijforecast.2009.05.027. URL https://www.sciencedirect.com/science/article/pii/S016920700900096X. Special section: Decision making and planning under low levels of predictability.

Terzi, E., Bonassi, F., Farina, M., and Scattolini, R. Learning model predictive control with long short-term memory networks. *International Journal of Robust and Nonlinear*

Control, 31(18):8877–8896, 2021. doi: https://doi.org/10.1002/rnc.5519. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/rnc.5519.

Tiao, G. C. and Tsay, R. S. Some advances in non-linear and adaptive modelling in time-series. *Journal of forecasting*, 13(2):109–131, 1994.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Wang, J., Song, Y., Liu, F., and Hou, R. Analysis and application of forecasting models in wind power integration: A review of multi-step-ahead wind speed forecasting models. *Renewable and Sustainable Energy Reviews*, 60:960–981, 2016.

Wang, X., Zhou, T., Wen, Q., Gao, J., Ding, B., and Jin, R. CARD: Channel aligned robust blend transformer for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=MJksrOhurE.

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

Yi, Z., Liu, X. C., Wei, R., Chen, X., and Dai, J. Electric vehicle charging demand forecasting using deep learning model. *Journal of Intelligent Transportation Systems*, 26(6):690–703, 2022.

Yun, C., Bhojanapalli, S., Rawat, A. S., Reddi, S. J., and Kumar, S. Are transformers universal approximators of sequence-to-sequence functions? *arXiv preprint arXiv:1912.10077*, 2019.

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

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., and Zhang, W. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11106–11115, 2021.

Zhou, T., MA, Z., wang, x., Wen, Q., Sun, L., Yao, T., Yin, W., and Jin, R. FiLM: Frequency improved legendre memory model for long-term time series forecasting. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 12677–12690. Curran Associates, Inc., 2022a.

Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., and Jin, R. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 27268–27286. PMLR, 17–23 Jul 2022b. URL https://proceedings.mlr.press/v162/zhou22g.html.

A. Approximation Guarantees

The functional space induced by the parametrization, which we will denote $\mathcal{F}_\theta = \{f_\theta : \boldsymbol{\theta} \in \Theta\}$ can be non-convex, as in the case of transformer architectures. However, it has been shown (Chamon & Ribeiro, 2020; Hounie et al., 2023) that as long as the distance to its convex hull $\overline{\mathcal{F}}_\theta = \operatorname{conv}(\mathcal{F}_\theta)$ is bounded, then we can leverage the strong duality of the convex variational program defined over $\overline{\mathcal{F}}_\theta$ together with uniform convergence bounds to provide approximation guarantees. That is, the values of (R-LS) and (ED-LS) are close. This holds both for the constrained and resilient formulation, although as discussed next, the resilient problem requires milder assumptions.

Assumption A.1. There exist a function $\phi \in \mathcal{F}$ such that all constraints are met with margin $c > 0$, i.e., $\mathbb{E}_{\mathcal{D}_i}\big[\ell_i(\phi(\mathbf{x}), y)\big] \leq c$, for all $i = 1, \ldots, m$.

Assumption A.1.$'$ There exist a finite relaxation $\boldsymbol{\zeta} \preceq \infty$ and a function $\phi \in \mathcal{F}$ such that all constraints are met with margin $c > 0$, i.e., $\mathbb{E}_{\mathcal{D}_i}\big[\ell_i(\phi(\mathbf{x}), y)\big] \leq \boldsymbol{\zeta} - c$, for all $i = 1, \ldots, m$.

Assumption A.2. The loss functions ℓ_i, $i = 0 \ldots m$, are convex and M-Lipschitz continuous.

Assumption A.3. For every $\phi \in \mathcal{F}$, there exists $\boldsymbol{\theta}^\dagger \in \Theta$ such that $\mathbb{E}_{\mathcal{D}_i}\big[|\phi(\mathbf{x}) - f_{\boldsymbol{\theta}^\dagger}(\mathbf{x})|\big] \leq \nu$, for all $i = 0, \ldots, m$.

Assumption A.4. There exists $\xi(N, \delta) \geq 0$ such that for all $i = 0, \ldots, m$ and all $\boldsymbol{\theta} \in \Theta$,

$$\left|\mathbb{E}_{\mathcal{D}_i}\left[\ell_i(f_\theta(\mathbf{x}_c), y_p)\right] - \frac{1}{N}\sum_{n=1}^{N} \ell_i\big(f_\theta(\mathbf{x}_c^n), y_p^n\big)\right| \leq \xi(N, \delta)$$

with probability $1 - \delta$ over draws of $\{(\mathbf{x}_i^n, y_i^n)\}$.

Note that the constraint qualification in Assumption A.1 (known as Slater's condition (Boyd et al., 2004)), which is required for the constrained problem, can be relaxed to the milder qualification given in Assumption A.1.$'$ for the resilient problem.

Assumption A.2 holds for commonly used objectives in Time-Series forecasting, including Mean Squared Error, Mean Absolute Error and Huber-loss, among others.

Assumption A.3 will hold if the parametrized function space is rich in the sense that the distance to its convex hull is bounded. Since neural networks and transformers have universal approximation properties (Hornik et al., 1989; Yun et al., 2019), we posit that given a parametrization with enough capacity, this assumption holds.

Assumption A.4 – known as uniform convergence – is customary in statistical learning theory. This includes generalization bounds based on VC dimension, Rademacher complexity, or algorithmic stability, among others (Mohri et al., 2018). Unlike generalization bounds for the unconstrained problem (ERM), this assumption bounds the loss at each step, and not the aggregated loss throughout the window.

We now state the main theorems that bound the approximation errors.

Theorem 1 (Hounie et al., 2023) Let $\boldsymbol{\zeta}^\star$ be an optimal relaxation for problem (R-LS). Under Ass. A.1$'$–A.4, it holds with probability of $1 - (3m + 2)\delta$ that

$$\left|P^\star - D^\star\right| \leq h(\boldsymbol{\zeta}^\star + 1 \cdot M\nu) - h(\boldsymbol{\zeta}^\star) + M\nu + (1+\Delta)\xi(N, \delta).$$

Theorem 2 (Chamon & Ribeiro, 2020) Let $\boldsymbol{\lambda}^\star$ be an optimal relaxation for the constrained problem (P-LS) defined over $\overline{\mathcal{F}}_\theta$ with constraints $\boldsymbol{\epsilon}_i - M\nu$. Under Ass. A.1–A.4, it holds with probability of $1 - (3m + 2)\delta$ that

$$\left|P^\star - D^\star\right| \leq (1 + \boldsymbol{\lambda}^\star)M\nu + (1 + \Delta)\xi(N, \delta).$$

Note that unlike in unconstrained statistical learning, optimal dual variables $\boldsymbol{\lambda}^\star$ or relaxations $\boldsymbol{\zeta}^\star$ play an important role in these approximation bounds. This is because, intuitively, they represent the difficulty of satisfying constraints.

B. Monotonic Loss constraints

A less restrictive constraint that accounts for error compounding is to only require errors to be monotonically increasing. That is, that there exists some $\boldsymbol{\epsilon}$ such that $\boldsymbol{\epsilon}_{t+1} \geq \boldsymbol{\epsilon}_t$ for all t. Instead of searching for such an $\boldsymbol{\epsilon}$, we can directly impose monotonicity, which requires only a small modification in (P-LS), as discussed in Section B.1.

B.1. Formulation

We can require the loss on each timestep i to be smaller than the loss step at the following $i + 1$. This leads to the constrained statistical learning problem:

$$\begin{aligned} \min_{\boldsymbol{\theta} \in \Theta}\ & \mathbb{E}_{(\mathbf{x}_c, y_p) \sim \mathfrak{D}}\left[\frac{1}{T_p}\sum_{i=1}^{T_p} \ell_i(f_{\boldsymbol{\theta}}(\mathbf{x}_c), y_p)\right] \\ \text{s. to: } & \mathbb{E}_{(\mathbf{x}_c, y_p) \sim \mathfrak{D}}\left[\ell_i(f_{\boldsymbol{\theta}}(\mathbf{x}_c), y_p)\right] \\ & \quad \leq \mathbb{E}_{(\mathbf{x}_c, y_p) \sim \mathfrak{D}}\left[\ell_{i+1}(f_{\boldsymbol{\theta}}(\mathbf{x}_c), y_p)\right] \\ & \qquad i = 1, \ldots, T_{\text{pred}-1}. \end{aligned} \tag{P-M}$$

Because of linearity of the expectation, the constraint can be re-written as a single expectation of a (non-convex) functional, explicitly,

$$\begin{aligned} \mathbb{E}_{(\mathbf{x}_c, y_p) \sim \mathfrak{D}}\ & [\ell_i(f_{\boldsymbol{\theta}}(\mathbf{x}_c), y_p) \\ & -\ell_{i+1}(f_{\boldsymbol{\theta}}(\mathbf{x}_c), y_p)] \leq 0. \end{aligned}$$

Despite the lack of convexity, the bounds presented in Appendix A still hold under additional assumptions on the distribution of the data and model class, as shown in Theorem 1 and Proposition B.1 (Chamon et al., 2022). We include these assumptions here for completeness.

Assumption B.1. The functions $y \mapsto \ell_i(\phi(\cdot), y) f_i(\cdot \mid y), i = 1, \ldots, m$, are uniformly continuous in the total variation topology for each $\phi \in \mathcal{H}$, where $f_i(\boldsymbol{x} \mid y)$ denotes the density of the conditional random variable induced by $\mathfrak{D}_i$. Explicitly, for each $\phi \in \overline{\mathcal{H}}$ and every $\epsilon > 0$ there exists $\delta_{\phi,i} > 0$ such that for all $|y - \tilde{y}| \leq \delta_{\phi,i}$ it holds that

$$\sup_{\mathcal{Z} \in \mathcal{B}} \int_{\mathcal{Z}} |\ell_i(\phi(\boldsymbol{x}), y) f_i(\boldsymbol{x} \mid y) - \ell_i(\phi(\boldsymbol{x}), \tilde{y}) f_i(\boldsymbol{x} \mid \tilde{y})| \, d\boldsymbol{x} \leq \epsilon$$

Assumption B.2. The conditional distribution $\boldsymbol{x} \mid y$ induced by $\mathfrak{D}$ are non-atomic.

Note that the epigraph formulation of (P-M) resembles that of the original los schaping constrained problem (P-LS), with the difference that $\boldsymbol{\epsilon}$ is an optimization variable, now constrained to be monotonically increasing:

$$\begin{aligned} \min_{\boldsymbol{\theta} \in \Theta} \; & \mathbb{E}_{(\mathbf{x}_c, y_p) \sim \mathfrak{D}} \left[\frac{1}{T_p} \sum_{i=1}^{T_p} \ell_i(f_{\boldsymbol{\theta}}(\mathbf{x}_c), y_p) \right] \\ \text{s. to :} \quad & \mathbb{E}_{(\mathbf{x}_c, y_p) \sim \mathfrak{D}} \left[\ell_i(f_{\boldsymbol{\theta}}(\mathbf{x}_c), y_p) \right] \leq \boldsymbol{\epsilon}_i \\ & \boldsymbol{\epsilon}_i \leq \boldsymbol{\epsilon}_{i+1} \\ & i = 1, \ldots, T_p - 1. \end{aligned} \tag{P-M-epi}$$

B.2. Equivalent Resilient formulations

The resilient relaxation, as defined in Equation 1 can be found by solving the *Resilient* Constrained Learning problem R-LS. Note that evaluating $P^\star(\boldsymbol{\epsilon} + \boldsymbol{\zeta})$ amounts to solving the a constrained learning problem. As shown by (Hounie et al., 2023) the relaxation and model can be found jointly by solving:

$$\begin{aligned} \min_{\boldsymbol{\theta} \in \Theta} \; & \mathbb{E}_{(\mathbf{x}_c, y_p) \sim \mathfrak{D}} \left[\frac{1}{T_p} \sum_{i=1}^{T_p} \ell_i(f_{\boldsymbol{\theta}}(\mathbf{x}_c), y_p) \right] + h(\boldsymbol{\zeta}) \\ \text{s. to :} \; & \mathbb{E}_{(\mathbf{x}_c, y_p) \sim \mathfrak{D}} \left[\ell_i(f_{\boldsymbol{\theta}}(\mathbf{x}_c), y_p) \right] \leq \boldsymbol{\epsilon}_i + \boldsymbol{\zeta}_i \\ & i = 1, \ldots, T_p. \end{aligned} \tag{R-LS}$$

The Lagrangian associated with this problem is

$$\begin{aligned} L(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\epsilon}, \boldsymbol{\zeta}) :=& h(\boldsymbol{\zeta}) \\ & + \sum_{i=1}^{T_p} \left(\boldsymbol{\lambda}_i + \frac{1}{T_p} \right) \\ & \mathbb{E}_{(\mathbf{x}_c, y_p) \sim \mathfrak{D}} \left[\ell_i \left(f_{\boldsymbol{\theta}}(\mathbf{x}_c), y_p \right) \right] \\ & - \boldsymbol{\lambda}_i (\epsilon_i + \boldsymbol{\zeta}_i). \end{aligned}$$

Since it is separable in $\boldsymbol{\theta}$ and $\boldsymbol{\zeta}$, we can re-write the joint minimization as

$$\begin{aligned} \min_{\boldsymbol{\zeta} \in \mathbb{R}^{T_p}, \boldsymbol{\theta} \in \Theta} L(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\epsilon}, \boldsymbol{\zeta}) =& \min_{\boldsymbol{\zeta} \in \mathbb{R}^{T_p}} L_{\boldsymbol{\zeta}}(\boldsymbol{\lambda}, \boldsymbol{\zeta}) \\ & + \min_{\boldsymbol{\theta} \in \Theta} L_{\boldsymbol{\theta}}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\epsilon}), \end{aligned} \tag{2}$$

where

$$L_{\boldsymbol{\zeta}}(\boldsymbol{\lambda}, \boldsymbol{\zeta}) = h(\boldsymbol{\zeta}) - \boldsymbol{\lambda}^T \boldsymbol{\zeta},$$

and

$$\begin{aligned} L_{\boldsymbol{\theta}}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\epsilon}) =& \sum_{i=1}^{T_p} \left(\boldsymbol{\lambda}_i + \frac{1}{T_p} \right) \mathbb{E}_{(\mathbf{x}_c, y_p) \sim \mathfrak{D}} \left[\ell_i \left(f_{\boldsymbol{\theta}}(\mathbf{x}_c), y_p \right) \right] \\ & - \boldsymbol{\lambda}^T \boldsymbol{\epsilon}. \end{aligned}$$

Since the minimization over $\boldsymbol{\zeta}$ in Equation 2 yields $-h^*(\boldsymbol{\lambda})$, where $-h^*$ is the convex conjugate of h, we can re-write the Lagrangian minimization as

$$\min_{\boldsymbol{\zeta} \in \mathbb{R}^{T_p}, \boldsymbol{\theta} \in \Theta} L(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\epsilon}, \boldsymbol{\zeta}) = \min_{\boldsymbol{\theta} \in \Theta} L_{\boldsymbol{\theta}}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\epsilon}) - h^*(\boldsymbol{\lambda}).$$

Then the dual problem associated to R-LS is equivalent to

$$D^\star = \max_{\boldsymbol{\lambda} \geq 0} \; \min_{\boldsymbol{\theta} \in \Theta} L_{\boldsymbol{\theta}}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\epsilon}) - h^*(\boldsymbol{\lambda}),$$

which corresponds to the original dual problem with a regularization on dual variables given by $-h^*(\boldsymbol{\lambda})$.

Although this equivalent formulation was not used directly in this work, which instead solves for $\boldsymbol{\zeta}$ using gradient descent as described in the next section, it helps to interpret how the resilient approach makes the problem easier to solve by regularizing the dual function.

B.3. Algorithm development

Algorithm 1 aims to find a saddle-point of the empirical Lagrangian

$$\begin{aligned} & \hat{L}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\epsilon}, \boldsymbol{\zeta}) := h(\boldsymbol{\zeta}) + \\ & \frac{1}{N} \sum_{n=1}^{N} \sum_{i=1}^{T_p} \left(\boldsymbol{\lambda}_i + \frac{1}{T_p} \right) \left[\ell_i([f_{\boldsymbol{\theta}}(\mathbf{x}_c^n)], y_p^n) \right] - \boldsymbol{\lambda}_i (\boldsymbol{\epsilon}_i + \boldsymbol{\zeta}_i), \end{aligned}$$

The updates in Algorithm 1 use gradient descent for $\boldsymbol{\zeta}$ and stochastic sub-gradient ascent for $\boldsymbol{\lambda}$. However, nothing precludes our method from using other optimization algorithms.

To obtain the gradients of the Lagrangian with respect to the primal variables, we exploit the fact that the minimization of the Lagrangian can be separated into two parts, each depending on only one primal variable, as described in the previous section.

The gradient with respect to $\boldsymbol{\zeta}$ is

$$d\boldsymbol{\zeta} = \frac{\partial L(\boldsymbol{\lambda}, \boldsymbol{\zeta})}{\partial \boldsymbol{\zeta}} = \nabla h(\boldsymbol{\zeta}) - \boldsymbol{\lambda}.$$

For dual variables, stochastic supergradient ascent updates the dual variable $\boldsymbol{\lambda}$ using a batch of B samples

$$d\boldsymbol{\lambda}_i = \frac{1}{B}\sum_{n=1}^{B} \ell_i(f_{\boldsymbol{\theta}}(x_n), y_n) - \boldsymbol{\zeta}_i \ \text{ for } i = 1 \ldots T_p.$$

Lastly, finding the parameters $\boldsymbol{\theta}$ that minimize the Lagrangian amounts to solving an empirical risk minimization problem with a time weighted loss, with weights given by the current multipliers $\boldsymbol{\lambda}$.

C. Experiment Setup

Early Stopping While the common practice in transformers for forecasting is to train with early stopping, we disable it for the constrained approach and train for a full 10 epochs, due to the slower convergence. Note that in the results presented in this work we keep early stopping for ERM.

Tuning of dual learning rate and initialization parameters. Preliminary exploration yielded consistently superior results with dual learning rate set to 0.01 and duals initialized to 1.0. All experiments reported in the paper were performed with this parameterization.

Choice of $\boldsymbol{\epsilon}$. To choose an appropriate upper bound constraint for the stepwise losses, we perform a grid search of six values for every setting of dataset, model and prediction window, optimizing for validation MSE. The values for the search are the 25, 50, and 75th percentile of the training and validation errors of each model trained with ERM. The values reported in Tables 4, 5, 6 and 7 are for the optimal values of this grid search.

Alternative choices of epsilon. During preliminary experiments, we explored

C.1. Datasets

We use commmon multivariate, long-term forecasting benchmarks from the transformer time series literature. Here we provide a brief summary of each dataset. For more details about the data, refer to their respective sources cited in the main body.

Weather: Contains local climatological data for around 1,600 U.S. locations from 2010 to 2013. It includes 11 climate features, and the target is “Wet Bulb Temp”. The data is available at `https://www.ncei.noaa.gov/data/local-climatological-data/`.

ETT(Electricity Transformer Temperature): Records of transformer oil temperature and six power load features at 1-hour and 15-minute intervals across two years from two separated counties in China. The data was collected by (Zhou et al., 2021).

ECL (Electricity Consuming Load): Includes hourly electricity consumption of 321 clients over two years, with the target being the ’MT 320’ consumption value. The data is available at `https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014`.

Exchange Rate: Contains the daily exchange rates of Australia, British, Canada, Switzerland, China, Japan, New Zealand and Singapore from 1990 to 2016. The data was collected by (Lai et al., 2018).

ILI (National Illness): Contains weekly influenza-like illness (ILI) patients’ data, showing the ratio of ILI patients to the total number of patients from 2002 to 2021. The data is available at `https://gis.cdc.gov/grasp/fluview/fluportaldashboard.html`.

Traffic: Contains hourly road occupancy rates measured by sensors in San Francisco Bay area freeways. The data is available at `https://pems.dot.ca.gov/`.

D. Extended Experiment Results

The unnormalized results used to compute the relative metrics presented in Section 5 are presented in Tables 4 and 5, containing the MSE of ERM, constrained and resilient runs, and Tables 6 and 7 contain Window STDs. While the overall effect can be better appreciated across all settings in Figures 4 and 3, the detailed tables also showcase how the loss shaping effect is observed across a wide variety of dataset and model combinations.

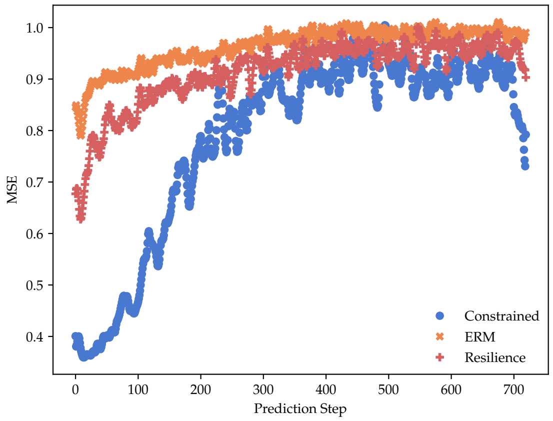

Figure 9. The anomaly of Figure 4. Test MSE of ERM, constrained and resilient runs.

	Features	Length	Target Mean	Target STD	Target Min	Target Max
ETTh1	6	17420	13.32	8.57	-4.08	46.01
ETTh2	6	17420	26.61	11.89	-2.65	58.88
ETTm1	6	69680	13.32	8.56	-4.22	46.01
ETTm2	6	69680	26.61	11.89	-2.65	58.88
Electricity	320	26304	3335.88	552.75	0.00	6035.00
Exchange Rate	7	7588	0.65	0.12	0.39	0.88
ILI	6	966	651497.46	349018.89	64699.00	1640587.00
Traffic	861	17544	0.03	0.02	0.00	0.22
Weather	20	52696	417.80	321.57	-9999.00	524.20

Table 3. Summary statistics for the benchmark datasets

D.1. Loss Shape Visualization

In this section, we take a closer look at some examples of loss distributions. As was shown in Section 5.1, the loss shaping effect is consistent as the predictive window increases. This is illustrated in Figure 10, where we plot the loss landscapes of three architectures tested on different datasets with increasing predictive window lengths. In all settings, a significant loss shaping effect is observed throughout the window, with the exception of the longest Pyraformer window (720), where the change is less pronounced.

The particular loss shape remains idiosyncratic to each dataset, model, and predictive window length combination. In settings (a) and (b) of Figure 10, a similar distribution pattern can be observed as the predictive window grows longer, while, in setting (c), the shape generated by the constrained model slightly varies with the window length.

Finally, we also observe that, more often than not, loss landscapes are not linear, let alone constant. Moreover, in rare cases, such as Reformer on Exchange Rate with prediction window of 192 steps (Figure 12), the stepwise loss surprisingly trends *downwards*. This motivates further experimentation on different kinds of constraints that are better suited for a particular learning problem.

D.2. Resilient Loss Shape Plots

We have presented extensive evidence that the resilient loss shaping effect is also consistent across different settings and predictive window lengths, and often times finds better shapes than non-resilient constrained learning. In Figure 11, we present three more loss distributions to illustrate this. In all window lengths of the three cases, except for Nonstationary Transformer (336), resilient learning results in flatter loss shapes or lower MSEs. Case (a) is worth noting: by finding the optimal relaxation of constraints, resilient is able to trade off a higher error in the first timesteps for a more stable error throughout the rest of the window. This case is also notable because the most pronounced loss shaping occurred in the setting with longest predictive window, which is in line with our analysis of effect size by window length in Section 5.1.

D.3. Correlation Between Training and Test Errors.

We compute the Spearman correlation between mean stepwise errors computed on the training and test sets for models trained with constant constraint levels. Compared to the values reported for ERM in Table 8, we observe weaker correlations for our method. This can indicate that, while imposing a constraint on the training loss, our approach increasingly *overfits* to training data.

D.4. Correlations Between Prediction Lengths

In order to quantify the degree of similarity between error patterns across various prediction lengths, we compute Pearson correlation coefficients between step-wise errors, as shown in Figure 13. We achieve this by first re-sampling shorter prediction windows using linear interpolation to match the number of steps. Notably, some setups present high correlations for several prediction lengths, which we interpret as the error distributions being similar.

D.5. Ablation Analysis for Constant Constraints

By examining the loss landscape across varying values of ϵ used on the Exchange Rate dataset, we find that our constrained method is not overly sensitive to the choice of hyperparameters. In Table 9, we can observe each model's relative changes in MSE. Pyraformer, Reformer, Informer and Nonstationary Transformer exhibit a biphasic behavior: the loss shape changes drastically only after the ϵ is set to a parameter much higher than ERM's average training loss, which results in a trivial constraint. In the cases of FiLM and iTransformer, the shape remains relatively consistent across all values, while Autoformer exhibits larger variations. The Resilient approach is less sensitive to the constraint level, in some cases allowing for better generalization with the same value of epsilon compared to the constrained run.

Figure 14 illustrates the effect of the ϵ parameter on various models on the Exchange Rate dataset. We choose ϵ to be

		Autoformer			FiLM			Informer			Nonstationary Transformer		
		ERM	Ours	Ours+R	ERM	Ours	Ours+R	ERM	Ours	Ours+R	ERM	Ours	Ours+R
Etth1	96	0.679	0.644	**0.606**	0.468	**0.459**	0.514	0.890	**0.800**	0.812	0.623	0.623	**0.585**
	192	0.869	**0.838**	1.085	0.524	**0.514**	0.561	1.179	1.040	**0.987**	0.647	0.647	**0.611**
	336	0.853	0.896	**0.653**	0.578	**0.567**	0.612	1.237	**1.232**	1.333	0.853	**0.819**	0.842
	720	1.107	**1.052**	1.274	0.751	**0.732**	0.769	1.278	**1.277**	1.285	0.967	0.974	**0.899**
Etth2	96	0.314	0.261	**0.219**	0.189	**0.188**	0.200	0.433	**0.349**	0.349	0.239	**0.238**	0.254
	192	0.340	**0.319**	0.507	0.232	**0.231**	0.243	0.648	0.488	**0.478**	0.294	**0.289**	0.294
	336	0.399	**0.334**	0.355	0.266	**0.264**	0.275	0.675	0.462	**0.441**	**0.296**	0.296	0.296
	720	0.436	0.470	**0.339**	0.343	**0.340**	0.351	0.794	0.665	**0.653**	0.316	**0.312**	0.322
Ettm1	96	0.605	**0.588**	0.601	**0.363**	0.369	0.374	0.646	**0.571**	0.585	**0.535**	**0.535**	0.634
	192	0.828	**0.641**	0.760	**0.431**	0.436	0.442	0.768	**0.577**	0.585	0.592	**0.581**	0.649
	336	0.685	**0.658**	0.749	**0.502**	0.505	0.512	0.857	**0.842**	0.848	**0.661**	0.666	0.694
	720	0.746	**0.726**	0.734	**0.577**	0.579	0.590	0.929	0.922	**0.900**	**0.748**	**0.748**	0.818
Ettm2	96	0.168	0.165	**0.148**	**0.122**	0.123	0.125	0.190	0.141	**0.137**	0.139	**0.135**	0.141
	192	0.191	0.172	**0.169**	**0.150**	0.152	0.153	0.223	**0.188**	0.200	0.178	**0.175**	0.187
	336	0.214	**0.197**	0.224	**0.183**	0.185	0.186	0.289	**0.282**	0.309	0.246	**0.230**	0.251
	720	0.271	**0.260**	0.336	**0.242**	0.243	0.244	0.572	**0.533**	0.558	0.328	0.328	**0.306**
ECL	96	0.204	**0.200**	0.220	0.239	**0.198**	0.217	0.325	**0.319**	0.342	0.178	0.182	**0.175**
	192	**0.215**	0.226	0.258	0.266	**0.200**	0.221	0.352	**0.341**	0.360	**0.186**	0.191	0.190
	336	0.334	**0.249**	0.287	0.306	**0.219**	0.239	0.354	**0.348**	0.371	**0.204**	0.204	0.206
	720	**0.252**	0.270	0.410	0.351	**0.281**	0.294	0.385	**0.377**	0.438	0.233	**0.229**	0.230
Exchange	96	**0.155**	0.179	0.163	0.162	**0.094**	0.100	0.880	**0.872**	0.910	0.160	0.152	**0.139**
	192	2.052	**0.266**	0.267	0.256	**0.182**	0.187	1.113	**1.110**	1.131	**0.268**	0.290	0.273
	336	3.442	0.446	**0.442**	0.542	**0.319**	0.323	1.633	1.088	**1.080**	**0.412**	0.573	0.488
	720	1.852	1.526	**0.995**	1.153	**0.820**	0.822	2.961	1.170	**1.166**	1.342	**1.120**	1.123
Illness	24	**3.180**	3.398	3.454	3.441	**3.436**	3.452	5.213	**5.174**	5.264	2.476	**2.275**	2.305
	36	3.216	3.213	**3.182**	3.929	3.920	**3.918**	5.785	5.792	**5.427**	**2.257**	2.413	2.462
	48	3.013	**2.738**	2.746	3.365	**3.360**	3.367	5.500	**5.496**	5.686	2.703	**2.546**	2.605
	60	3.148	3.106	**2.875**	3.783	**3.780**	3.782	5.772	5.783	**5.703**	2.634	**2.587**	2.606
Traffic	96	0.633	**0.615**	0.691	**0.411**	0.652	0.695	0.726	**0.716**	0.886	0.634	**0.604**	0.625
	192	0.613	**0.602**	0.679	**0.408**	0.605	0.651	**0.736**	0.738	0.776	**0.624**	0.631	0.649
	336	0.612	**0.606**	0.622	**0.426**	0.616	0.667	0.830	**0.821**	0.891	**0.625**	0.646	0.653
	720	**0.675**	0.677	0.785	**0.526**	0.711	0.775	1.016	**0.885**	1.016	0.660	**0.657**	0.699
Weather	96	0.375	**0.300**	0.314	**0.196**	0.198	0.199	0.435	**0.395**	0.441	**0.203**	0.204	0.206
	192	**0.303**	0.311	0.322	**0.234**	0.240	0.241	0.515	**0.447**	0.491	**0.271**	0.283	0.286
	336	**0.359**	0.374	0.402	**0.278**	0.290	0.291	**0.600**	0.656	0.600	0.383	**0.356**	0.375
	720	0.443	**0.442**	0.447	**0.342**	0.361	0.362	1.100	**1.051**	1.198	0.454	0.466	**0.438**

Table 4. MSE of ERM, constrained (Ours) and resilient runs (Ours+R) for each model and dataset, part 1/2, for each prediction length.

		Pyraformer			Reformer			Transformer			iTransformer		
		ERM	Ours	Ours+R	ERM	Ours	Ours+R	ERM	Ours	Ours+R	ERM	Ours	Ours+R
Etth1	96	**0.569**	**0.569**	0.572	0.789	**0.691**	0.727	0.866	0.770	**0.739**	0.459	0.459	**0.440**
	192	**0.618**	**0.618**	0.636	**0.818**	0.820	0.867	0.988	**0.780**	0.865	0.539	0.539	**0.535**
	336	0.756	0.750	**0.748**	**0.833**	0.941	1.006	**0.941**	0.968	1.108	**0.594**	0.601	0.602
	720	0.994	0.971	**0.960**	**0.911**	1.038	1.164	0.989	**0.954**	1.113	**0.747**	**0.747**	0.755
Etth2	96	0.277	0.277	**0.251**	0.562	0.309	**0.307**	0.274	**0.231**	0.233	0.203	0.203	**0.190**
	192	0.583	0.588	**0.548**	1.246	**0.359**	0.374	**0.316**	0.327	0.320	0.242	0.247	**0.234**
	336	0.597	0.595	**0.541**	1.327	**0.442**	0.453	**0.378**	0.392	0.383	0.271	0.279	**0.271**
	720	0.660	0.660	**0.524**	2.062	0.711	**0.615**	**0.381**	0.490	0.496	0.323	0.327	**0.321**
Ettm1	96	0.542	0.542	**0.495**	0.697	**0.517**	0.518	0.569	**0.565**	0.579	0.414	0.414	**0.383**
	192	0.607	0.607	**0.522**	0.830	0.618	**0.613**	**0.637**	0.650	0.689	0.476	0.476	**0.461**
	336	0.658	0.658	**0.595**	0.860	0.707	**0.707**	0.735	**0.713**	0.773	**0.535**	**0.535**	0.539
	720	0.760	**0.744**	0.757	0.892	0.868	**0.861**	0.989	0.865	**0.823**	0.608	0.608	**0.602**
Ettm2	96	0.143	0.143	**0.135**	0.183	0.173	**0.172**	0.162	0.135	**0.133**	0.134	0.134	**0.126**
	192	0.212	0.212	**0.188**	0.297	0.241	**0.239**	0.213	**0.203**	0.209	0.178	0.178	**0.161**
	336	**0.335**	0.371	0.351	0.739	0.319	**0.314**	0.281	0.293	**0.259**	0.212	0.211	**0.210**
	720	0.508	0.508	**0.422**	1.878	**0.549**	0.566	0.413	0.443	**0.398**	0.271	**0.265**	0.267
ECL	96	0.287	**0.280**	0.303	0.299	**0.297**	0.303	0.262	**0.259**	0.278	**0.148**	0.159	0.171
	192	0.294	**0.291**	0.320	0.332	**0.326**	0.332	0.281	**0.274**	0.321	**0.164**	0.173	0.183
	336	0.313	**0.301**	0.324	0.359	0.365	**0.329**	**0.280**	0.287	0.297	**0.178**	0.190	0.200
	720	0.310	**0.298**	0.324	**0.316**	0.316	0.326	**0.289**	0.293	0.312	**0.208**	0.227	0.240
Exchange	96	0.802	0.774	**0.584**	**0.960**	1.025	0.981	0.691	0.593	**0.556**	**0.087**	0.117	0.099
	192	1.216	1.260	**0.988**	**1.398**	1.402	1.685	1.235	1.253	**1.174**	**0.180**	0.226	0.199
	336	1.666	1.388	**1.319**	1.909	**1.632**	1.924	1.421	**1.362**	1.649	**0.333**	0.364	0.344
	720	2.051	1.725	**1.686**	2.095	**2.063**	2.114	2.197	1.029	**1.015**	**0.865**	0.986	0.920
Illness	24	4.724	4.724	**4.721**	3.948	3.944	**3.901**	**5.203**	5.270	5.959	1.935	**1.918**	1.942
	36	4.972	**4.924**	4.965	4.069	**4.069**	4.218	4.571	**4.528**	6.038	1.865	**1.862**	1.871
	48	4.686	4.722	**4.677**	4.432	4.433	**4.335**	5.166	**5.073**	5.678	**1.942**	1.945	1.949
	60	4.892	**4.889**	4.893	4.574	**4.564**	4.575	5.572	**5.482**	5.673	1.849	**1.849**	1.860
Traffic	96	0.691	**0.678**	0.704	0.698	**0.693**	0.702	**0.639**	0.640	0.649	**0.392**	0.435	0.459
	192	0.676	**0.660**	0.694	0.690	**0.684**	0.703	0.642	**0.635**	0.651	**0.411**	0.452	0.475
	336	0.694	**0.671**	0.713	0.690	**0.686**	0.704	0.666	**0.663**	0.669	**0.424**	0.471	0.494
	720	0.713	**0.708**	0.743	0.682	**0.678**	0.704	0.698	**0.693**	0.696	**0.454**	0.508	0.536
Weather	96	0.217	0.218	**0.213**	0.422	0.334	**0.331**	0.448	0.462	**0.435**	**0.174**	0.176	0.177
	192	0.287	**0.269**	0.270	0.703	**0.425**	0.483	0.667	**0.593**	0.640	0.226	**0.224**	0.225
	336	0.341	**0.325**	0.330	1.010	**0.588**	0.995	**0.749**	0.901	0.915	0.285	0.284	**0.282**
	720	0.460	**0.417**	0.423	0.963	**0.788**	0.914	1.002	**0.924**	1.072	0.363	0.363	**0.362**

Table 5. MSE of ERM, constrained (Ours) and resilient runs (Ours+R) for each model and dataset, part 2/2, for each prediction length.

		Autoformer			FiLM			Informer			Nonstationary Transformer		
		ERM	Ours	Ours+R	ERM	Ours	Ours+R	ERM	Ours	Ours+R	ERM	Ours	Ours+R
Etth1	96	0.024	0.032	**0.019**	0.070	0.064	**0.052**	0.183	0.122	**0.114**	0.060	0.060	**0.053**
	192	**0.139**	0.141	0.338	0.073	0.064	**0.052**	0.211	0.183	**0.128**	0.086	0.086	**0.081**
	336	0.083	0.109	**0.028**	0.075	0.063	**0.053**	0.181	**0.156**	0.200	**0.187**	0.207	0.209
	720	0.557	**0.278**	1.208	0.142	0.134	**0.126**	0.170	**0.169**	0.192	0.171	**0.160**	0.164
Etth2	96	0.041	0.029	**0.028**	0.046	0.043	**0.041**	0.138	0.110	**0.109**	0.070	**0.069**	0.072
	192	**0.015**	0.023	0.047	0.053	0.050	**0.048**	0.251	0.155	**0.147**	0.059	0.057	**0.055**
	336	0.079	**0.020**	0.073	0.054	0.051	**0.049**	0.203	0.097	**0.088**	0.045	**0.042**	0.045
	720	0.071	0.088	**0.045**	0.071	0.067	**0.066**	0.248	0.142	**0.118**	0.047	0.036	**0.035**
Ettm1	96	0.063	**0.061**	0.065	0.070	0.054	**0.051**	0.157	**0.119**	0.133	**0.120**	**0.120**	0.175
	192	0.091	0.093	**0.038**	0.088	0.077	**0.069**	0.120	**0.066**	0.072	0.109	**0.104**	0.117
	336	0.062	0.063	**0.033**	0.105	0.098	**0.089**	0.170	0.172	**0.137**	**0.102**	0.104	0.108
	720	0.062	0.061	**0.055**	0.102	0.094	**0.076**	0.128	0.127	**0.101**	**0.112**	**0.112**	0.143
Ettm2	96	0.034	0.024	**0.020**	0.028	0.025	**0.022**	0.064	0.030	**0.029**	0.036	**0.032**	0.035
	192	0.027	**0.025**	0.026	0.035	0.032	**0.030**	0.065	**0.050**	0.051	0.051	**0.049**	0.054
	336	0.041	0.038	**0.032**	0.048	0.046	**0.044**	0.080	**0.076**	0.080	0.082	**0.070**	0.080
	720	**0.052**	0.061	0.072	0.064	0.061	**0.059**	0.258	**0.222**	0.228	0.087	0.087	**0.084**
ECL	96	0.015	0.015	**0.013**	0.050	0.031	**0.029**	0.014	**0.013**	0.018	0.017	**0.014**	0.015
	192	0.022	0.021	**0.020**	0.045	0.029	**0.026**	0.041	0.037	**0.029**	0.026	**0.019**	0.020
	336	0.043	0.039	**0.020**	0.055	0.037	**0.032**	0.022	**0.020**	0.021	0.031	**0.024**	0.029
	720	0.035	**0.030**	0.156	0.052	0.053	**0.048**	0.028	**0.023**	0.040	0.040	0.037	**0.033**
Exchange	96	0.081	0.085	**0.049**	0.058	0.044	**0.043**	0.279	**0.274**	0.305	0.095	0.083	**0.072**
	192	1.349	**0.115**	0.115	0.116	0.096	**0.095**	0.308	**0.299**	0.331	**0.143**	0.153	0.148
	336	2.134	0.210	**0.208**	0.229	0.177	**0.176**	0.388	0.191	**0.178**	**0.215**	0.287	0.244
	720	1.038	0.743	**0.320**	0.665	**0.485**	0.487	0.946	0.128	**0.125**	0.938	**0.643**	0.646
Illness	24	**0.113**	0.140	0.203	0.607	0.602	**0.599**	**0.438**	0.472	0.510	**0.361**	0.517	0.515
	36	**0.197**	0.213	0.342	0.557	**0.547**	0.549	0.700	0.699	**0.658**	**0.279**	0.500	0.513
	48	0.308	**0.237**	0.376	0.618	0.615	**0.614**	**0.651**	0.725	0.932	**0.243**	0.474	0.457
	60	0.561	0.566	**0.398**	0.405	0.403	**0.403**	**1.029**	1.037	1.077	**0.485**	0.514	0.500
Traffic	96	0.023	**0.019**	0.024	**0.026**	0.121	0.117	0.030	**0.025**	0.057	0.018	0.014	**0.011**
	192	0.036	**0.026**	0.047	**0.025**	0.125	0.117	**0.038**	0.048	0.042	0.021	0.017	**0.017**
	336	0.031	0.029	**0.027**	**0.031**	0.133	0.125	0.077	**0.068**	0.085	0.021	0.026	**0.018**
	720	**0.036**	0.040	0.064	**0.054**	0.133	0.118	0.071	0.066	**0.062**	0.029	0.032	**0.023**
Weather	96	0.060	0.033	**0.031**	0.048	**0.043**	0.043	0.118	0.101	**0.099**	0.055	**0.047**	0.054
	192	0.053	0.051	**0.045**	0.058	**0.055**	0.055	0.158	**0.129**	0.160	0.085	**0.081**	0.084
	336	0.068	**0.063**	0.065	**0.070**	0.070	0.070	0.136	**0.119**	0.177	0.113	**0.104**	0.114
	720	0.079	0.074	**0.061**	**0.083**	0.089	0.088	0.364	**0.251**	0.528	0.117	**0.107**	0.126

Table 6. Window STD of ERM, constrained (Ours) and resilient runs (Ours+R) for each model and dataset, part 1/2, for each prediction length.

		Pyraformer			Reformer			Transformer			iTransformer		
		1ERM	2Ours	3Ours+R	1ERM	2Ours	3Ours+R	1ERM	2Ours	3Ours+R	1ERM	2Ours	3Ours+R
Etth1	96	0.074	0.074	**0.063**	0.058	**0.045**	0.046	0.146	0.120	**0.096**	0.081	0.081	**0.065**
	192	0.082	0.082	**0.072**	0.077	**0.040**	0.044	0.166	0.130	**0.087**	0.095	0.095	**0.082**
	336	0.153	**0.130**	0.146	0.062	**0.029**	0.044	0.123	0.137	**0.117**	0.104	**0.096**	0.097
	720	0.170	0.173	**0.140**	0.045	**0.044**	0.062	**0.093**	0.116	0.121	0.167	0.167	**0.163**
Etth2	96	0.082	0.082	**0.045**	0.262	0.043	**0.040**	0.050	0.036	**0.036**	0.055	0.055	**0.048**
	192	0.292	0.291	**0.275**	0.718	**0.041**	0.042	**0.053**	0.058	0.062	0.057	0.058	**0.054**
	336	0.212	0.216	**0.163**	0.332	**0.046**	0.049	0.064	0.064	**0.054**	0.051	0.054	**0.051**
	720	0.144	0.144	**0.040**	0.194	0.095	**0.053**	0.056	**0.039**	0.060	0.061	0.058	**0.057**
Ettm1	96	0.127	0.127	**0.108**	0.137	**0.082**	0.086	0.114	**0.108**	0.111	0.105	0.105	**0.093**
	192	0.104	0.104	**0.082**	0.116	**0.059**	0.062	**0.098**	0.101	0.124	0.104	0.104	**0.099**
	336	0.117	0.117	**0.087**	0.105	**0.069**	0.072	0.101	**0.099**	0.104	**0.113**	**0.113**	0.117
	720	0.122	0.113	**0.106**	0.074	**0.044**	0.050	0.152	0.124	**0.107**	0.106	0.106	**0.102**
Ettm2	96	0.038	0.038	**0.030**	0.054	0.036	**0.036**	0.039	0.031	**0.030**	0.037	0.037	**0.033**
	192	0.060	0.060	**0.046**	0.089	**0.052**	0.055	0.054	**0.049**	0.054	0.052	0.052	**0.042**
	336	**0.106**	0.134	0.109	0.226	**0.057**	0.061	0.069	0.067	**0.064**	0.065	0.065	**0.064**
	720	0.222	0.222	**0.157**	0.277	0.148	**0.140**	0.103	0.116	**0.099**	**0.073**	0.074	0.073
ECL	96	0.019	0.015	**0.015**	0.011	0.011	**0.008**	**0.007**	0.008	0.008	0.024	0.023	**0.023**
	192	0.022	0.019	**0.016**	0.020	0.020	**0.012**	0.018	0.015	**0.013**	0.028	0.025	**0.025**
	336	0.034	0.025	**0.021**	0.022	0.023	**0.011**	0.020	0.022	**0.018**	0.034	0.034	**0.033**
	720	0.015	0.016	**0.015**	0.007	0.007	**0.007**	**0.013**	0.018	0.018	**0.041**	0.045	0.043
Exchange	96	0.175	**0.105**	0.130	0.116	0.121	**0.098**	0.168	0.105	**0.088**	**0.049**	0.072	0.059
	192	0.360	**0.139**	0.278	**0.132**	0.148	0.196	0.377	0.385	**0.373**	**0.103**	0.137	0.115
	336	0.179	**0.095**	0.203	0.193	**0.154**	0.240	0.218	**0.211**	0.304	**0.194**	0.223	0.204
	720	0.788	**0.163**	0.170	0.314	**0.280**	0.344	0.611	0.169	**0.165**	**0.527**	0.605	0.561
Illness	24	1.092	1.092	**1.074**	0.459	0.460	**0.412**	**0.490**	0.508	0.679	0.807	**0.799**	0.806
	36	0.931	**0.893**	0.929	0.778	0.772	**0.445**	0.436	**0.436**	0.578	0.632	0.629	**0.625**
	48	0.862	0.882	**0.855**	0.425	0.423	**0.407**	0.403	**0.395**	0.680	**0.596**	0.602	0.599
	60	0.781	0.783	**0.772**	0.397	0.398	**0.364**	0.562	0.535	**0.457**	0.579	0.579	**0.573**
Traffic	96	0.032	0.026	**0.021**	0.021	0.021	**0.020**	0.011	**0.010**	0.010	**0.032**	0.034	0.032
	192	0.016	**0.013**	0.014	0.022	0.020	**0.019**	0.019	0.015	**0.015**	**0.029**	0.031	0.029
	336	0.015	**0.011**	0.013	0.017	0.016	**0.014**	0.025	0.021	**0.019**	**0.030**	0.035	0.033
	720	0.016	0.016	**0.016**	0.012	**0.010**	0.011	0.055	0.057	**0.038**	**0.038**	0.042	0.041
Weather	96	0.056	0.053	**0.052**	0.102	**0.086**	0.088	0.127	0.126	**0.121**	0.047	**0.045**	0.046
	192	0.096	0.085	**0.084**	0.142	**0.117**	0.157	0.243	**0.206**	0.242	0.062	0.061	**0.061**
	336	0.107	**0.098**	0.104	**0.141**	0.158	0.191	**0.225**	0.295	0.294	0.081	0.079	**0.079**
	720	0.156	**0.132**	0.134	**0.039**	0.184	0.067	0.289	**0.276**	0.321	0.097	0.094	**0.093**

Table 7. Window STD of ERM, constrained (Ours) and resilient runs (Ours+R) for each model and dataset, part 2/2, for each prediction length.

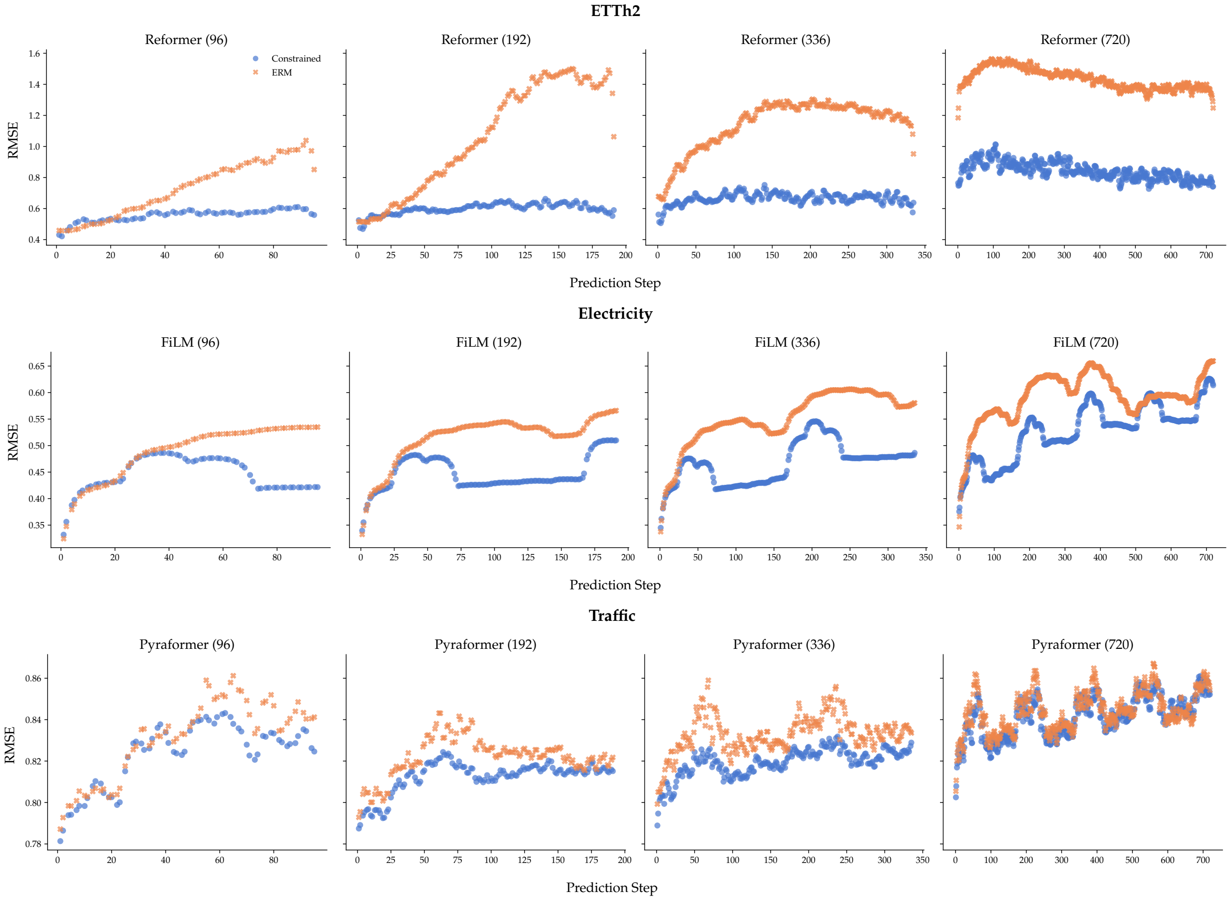

Figure 10. Test MSE for ERM and constrained runs for three different settings: (a) Model: Reformer, Dataset: ETTh2, (b) Model: FiLM, Dataset: Traffic, and (c) Model: Pyraformer, Dataset: Traffic. Each plot column shows a different predictive window length, each row shows a different dataset-model selection.

		Autoformer	Informer	Reformer	Transformer
ECL	96	0.38	0.34	0.60	0.18
	192	0.42	0.17	0.37	-0.06
	336	0.17	0.17	-0.19	-0.16
	720	0.30	0.20	0.07	0.19
Exchange	96	0.76	0.20	0.67	-0.15
	192	0.32	0.08	0.19	-0.55
	336	0.24	-0.44	0.17	0.43
	720	0.69	-0.63	0.12	-0.69
Weather	96	0.01	0.23	0.14	-0.30
	192	0.43	-0.08	-0.43	0.33
	336	0.46	-0.22	0.55	0.25
	720	0.83	0.45	0.69	0.64

Table 8. Spearman correlation for step-wise Mean Train and Test errors, for models trained with constant constraints.

the quartiles of the train and test error taken from an ERM run. Notably, for several models, including the Informer, Reformer, and Transformer, the loss landscape exhibits relative invariance across similar ϵ values. This suggests a plateau in sensitivity within this parameter range. Conversely, in some cases, a critical ϵ threshold seems to exist, after which the shape significantly changes and then stabilizes again as we relax ϵ further. Specifically, for Autoformer, Informer, and Transformer, optimal performance is achieved subsequent to a degree of ϵ, underscoring the benefit of exploring multiple constraint levels.

We have shown that our method is reasonably robust to the choice of the constraint hyperparameter. However, we note that performing a parameter search with different values of ϵ may still be valuable. When the generalization gap is high, training with constraints higher than all of the training values is often more effective. For example, Reformer's test MSE is 2.9x higher than its train MSE, and Table 9 shows its MSE can be reduced by up to 50.1% when using validation-based threshold values, while training-based ones are not as effective.

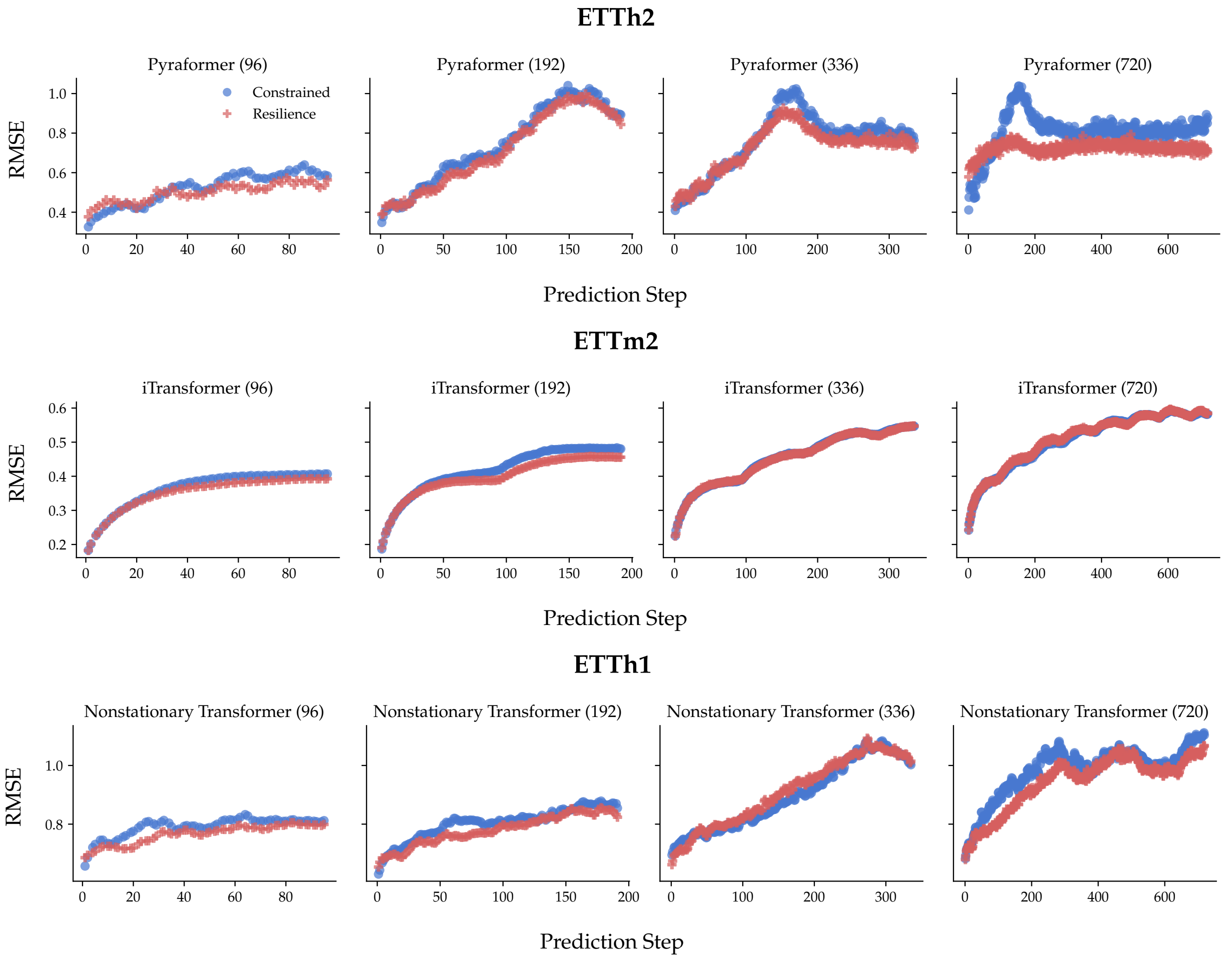

Figure 11. Test MSE for constrained and resilient runs for three different settings: (a) Model: Pyraformer, Dataset: ETTh2, (b) Model: iTransformer, Dataset: ETTm2, and (c) Model: Nonstationary Transformer, Dataset: Exchange Rate. Each plot column shows a different predictive window length, each row shows a different dataset-model selection.

D.6. Outlier in STD histogram

In Section 5.1, we showed that on average, using constraints reduces Window STD. However, we identified some outliers in Figure 4. In one instance, the constrained Window STD is 371% higher than ERM. It corresponds to Reformer on the Weather dataset with predictive window length of 720 steps, shown in Figure 9. Closer inspection reveals that the reason for this increase is due to the constrained model finding a

Model	**Train 25%**	**Train 50%**	**Train 75%**	**Val 25%**	**Val 50%**	**Val 75%**
Autoformer	48.85	39.92	45.48	17.64	46.98	2.61
FiLM	29.88	29.64	29.61	29.44	29.18	28.87
Informer	0.85	1.54	1.51	56.48	60.50	60.50
Nonstationary Transformer	-35.28	-36.45	-39.60	-2.79	4.31	16.54
Pyraformer	7.82	14.63	14.63	15.89	21.22	23.40
Reformer	1.76	1.53	1.72	-41.37	-48.63	-50.12
Transformer	-0.73	0.24	0.20	-2.46	53.18	54.38
iTransformer	-11.75	-13.02	-13.91	-15.52	-8.76	-8.74

Table 9. MSE % changes of constrained runs relative to ERM for varying values of ϵ, in the Exchange Rate dataset, with window length 720.

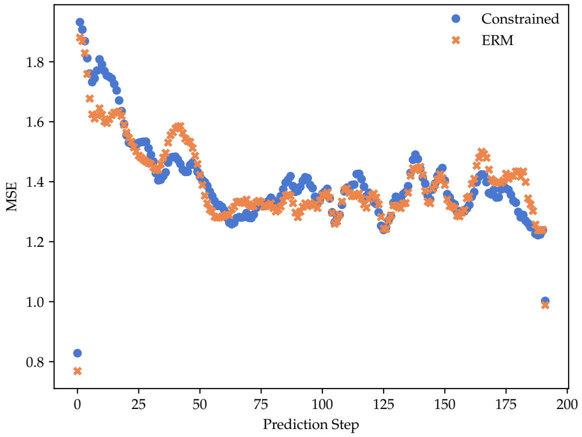

Figure 12. Test MSE of ERM and constrained models for Reformer on Exchange Rate, with window length of 192

much lower MSE (constrained 0.788 versus ERM 0.963) in the first part of the window. Interestingly, the resilient model settles for a middle point, with a lower MSE than ERM but less variance increase than the constrained model.

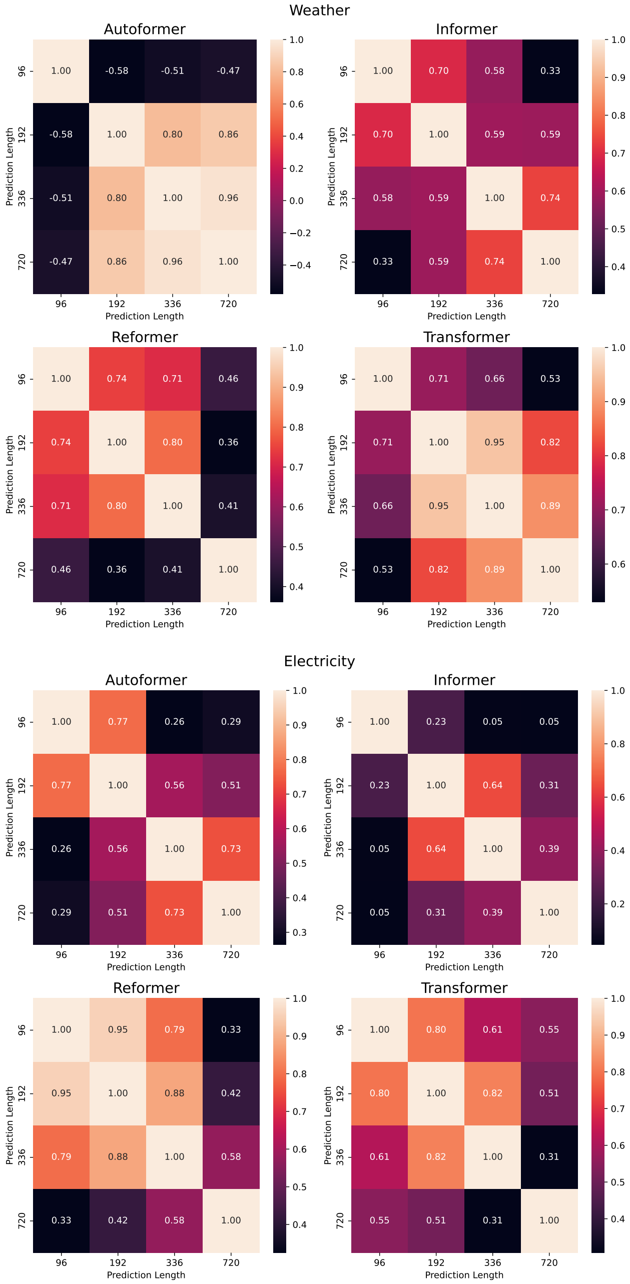

Figure 13. Pearson correlation between test step wise errors for weather and electricity datasets. Each heatmap shows pairwise correlations between all prediction lengths for a given model.

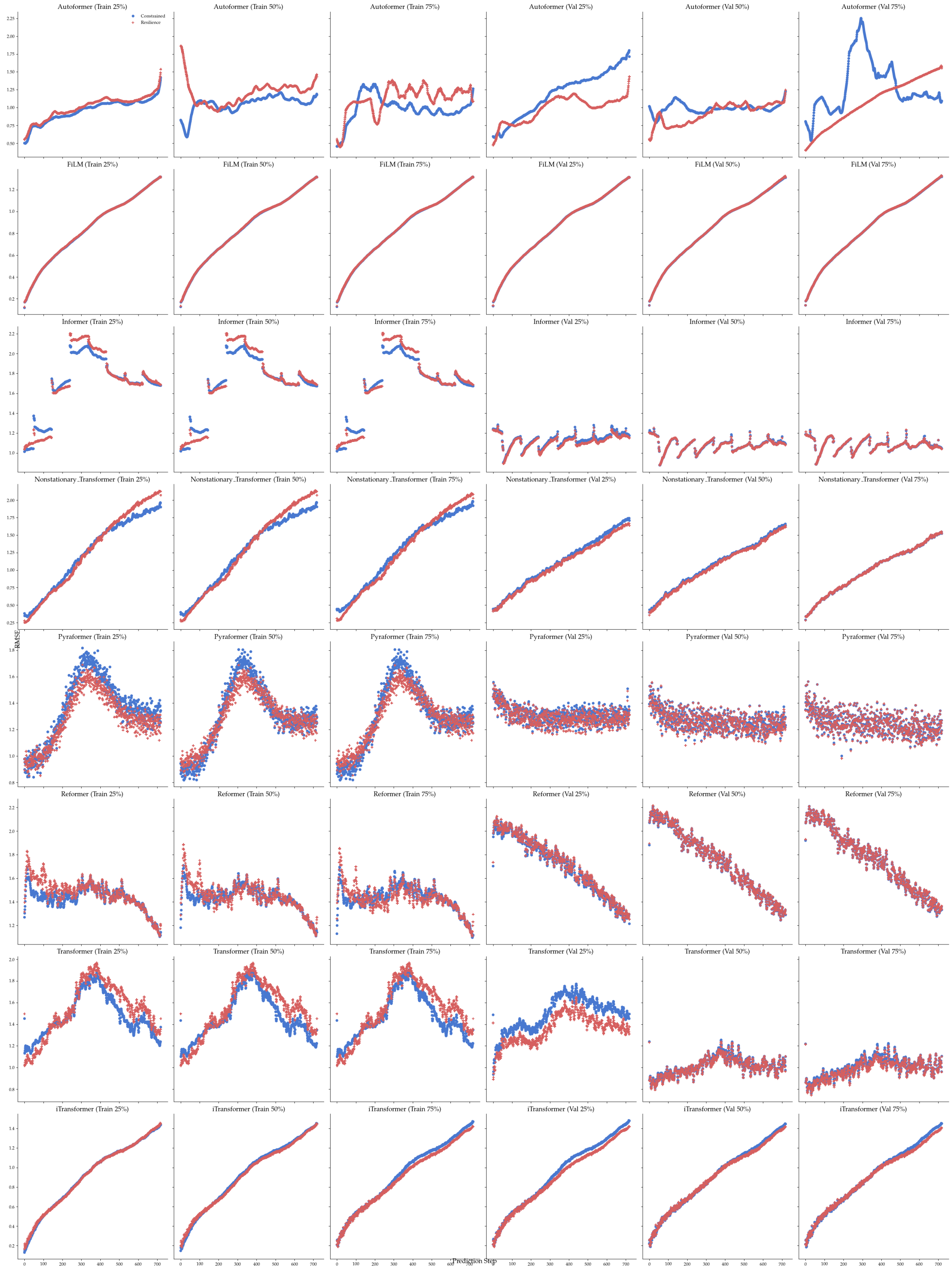

Figure 14. Ablation plots for the ϵ parameter on the Exchange Rate dataset, with predictive window length of 720. Rows are different models, and columns are increasing values of ϵ. Each plot shows test MSE of Constrained and Resilient runs.