We thank the reviewers for their meaningful and valuable comments, which help to improve the quality of our work.

R1 on paper motivation: Figure 1 in our submission illustrates the limitation of using MSE as a loss for training deep forecasting models. Since MSE is similar for the three predictions in (a), (b) and (c), a gradient-based optimization is unable to produce training signals preferring predictions (b) and (c) over (a). By no means we wanted to claim that certain existing approaches are unable to perform step prediction in this simple example. This work does not focus on designing new forecasting models, but introduces the STDL loss function as an alternative to MSE. STDL is thus model-agnostic and can be used for training various forecasting models - as shown in experiments and below.
R1 on state-of-the-art methods: We thank R1 for suggesting to compare our results to online regression with adaptive parameters (forgetting recursive least squares, adaptive Kalman filters). Although these are historical approaches for Bayesian inference in State Space Models (SSMs), their direct application to multi-step forecasting is not straightforward because they require input data for adaptation at each time step. For this reason, several state-of-the art multi-step approaches combine SSMs and deep learning based on Seq2Seq architectures [1, 2, 3, 4].
To fulfill R1 requests, we perform additional experiments (shown in blue) on the Traffic dataset (Table 4 in submission). The results of the Deep AR baseline[1] (obtained with GitHub code) is still outperformed by a simple Seq2Seq model trained with STDL (results shown in submission, column 4 in black), and equivalent in temporal metrics. Training Deep AR with STDL would be an interesting future exploration. Finally, we provide results when training the recent TT-RNN (refs [48,49] in submission) with STDL, reaching the best shape and temporal performances. These new results further highlight the importance of STDL ; we will be glad to add these comparisons in the final paper if accepted.

Eval loss		LSTNet-rec (MSE)	TT-RNN (MSE)	Deep AR (MSE)	Seq2Seq (STDL)	TT-RNN (STDL)
Euclidian	MSE (x100)	1.74 ± 0.11	$\mathbf{0.840 \pm 0.106}$	0.966 ± 0.068	1.00 ± 0.260	$\mathbf{0.930 \pm 0.09}$
Shape	DTW (x100)	42.0 ± 2.2	25.9 ± 1.99	27.8 ± 1.55	23.0 ± 1.62	$\mathbf{21.4 \pm 0.79}$
	Ramp (x10)	9.00 ± 0.577	6.71 ± 0.546	7.56 ± 0.42	5.93 ± 0.235	$\mathbf{5.27 \pm 0.27}$
Time	TDI (x10)	25.7 ± 4.75	17.8 ± 1.73	$\mathbf{14.6 \pm 0.94}$	$\mathbf{14.4 \pm 1.58}$	$\mathbf{15.7 \pm 1.02}$
	Hausdorff	2.34 ± 1.41	2.19 ± 0.12	2.04 ± 0.11	2.13 ± 0.514	$\mathbf{1.88 \pm 0.153}$

R2 on more complex datasets: As requested, we provide additional experiments on 2 more complex datasets: household electricity consumption and solar energy. The former corresponds to a multivariate forecasting problem involving 10 exogenous input variables (global intensity, voltage, sub-metering, date, *etc*), requiring the extraction of complex interactions in data for spiky patterns prediction. The latter has very fine time granularity (10min vs 1h for Traffic), needing to extract accurate time features. The results shown below again illustrate the superiority of training Seq2Seq models with SDTL compared to MSE.

	Household electricity consumption			**Solar energy**		
Method	MSE (x10)	DTW	TDI	MSE (x1000)	DTW (x100)	TDI (x10)
Seq2Seq MSE	$\mathbf{18.3 \pm 2.5}$	4.54 ± 0.40	$\mathbf{2.49 \pm 0.26}$	$\mathbf{13.7 \pm 1.5}$	24.3 ± 3.4	12.9 ± 1.4
Seq2Seq STDL	$\mathbf{19.9 \pm 2.4}$	$\mathbf{3.85 \pm 0.26}$	$\mathbf{2.30 \pm 0.59}$	$\mathbf{14.4 \pm 0.57}$	$\mathbf{20.9 \pm 1.1}$	$\mathbf{5.71 \pm 0.83}$

R2 on α tuning: α is chosen on a validation set, by selecting the lowest value for which $\mathcal{L}_{shape}$ gets comparable performance than a reference DTW_γ trained model. This setup will be added in the final version if accepted.

R3 on reporting STDL as evaluation metric: these results will be added in our tables if accepted.
R3 on training time: 1 training epoch with our Seq2Seq GRU network takes about 0.5s for MSE vs 1.7s for SDTL on Synthetic (1s vs 8s on ECG5000, 3s vs 33s on Traffic). The overhead is due to the sequential computation of the STDL (dynamic programming in forward and backward passes). Note that no overhead is involved at test time.
R3 on choice of δ: We choose δ as the euclidean loss (paper l. 102-103), which is common for computing DTW, but any other distance (*e.g.* mean absolute error) could be employed.
R3 on code sharing: source code will be made available on GitHub after acceptance.
R3 on ECG: predicting the shape and time interval between heartbeats could be helpful for cardiologists, to detect abnormal heartbeats such as 'premature ventricular contraction'.
R3 on feature interpretation: understanding the effects of our shape and time loss terms on the learned features is an interesting but non trivial perspective. A possible direction to this end is to use feature visualization techniques [5].
R3 on confidence intervals: we could use MC Dropout (Gal *et. al.*, ICML'16) to compute the predictive distribution of trajectories ; or embed the STDL loss in a deep SSM architecture suited for probabilistic forecasting.

[1] D. Salinas, V. Flunkert, J. Gasthaus. "DeepAR: Probabilistic forecasting with autoregressive recurrent networks", ICML 2017
[2] S. Rangapuram, M. Seeger, J. Gasthaus, L. Stella, Y. Wang, "Deep state space models for time series forecasting", NeurIPS2018
[3] X.Jin, S.Li, Y. Zhang, X.Yan, "Multi-step deep autoregressive forecasting with latent states", ICML 2019 Time Series Workshop
[4] Y. Wang, A. Smola, D. C Maddix, J. Gasthaus, D. Foster,T. Januschowski. "Deep factors for forecasting", ICML 2019
[5] Andrej Karpathy, Justin Johnson, and Li Fei-Fei. "Visualizing and understanding recurrent networks", ICLR 2016