A APPENDIX: PER DATASET STUDY - VISUALIZATION OF FUTURE TRAJECTORIES, HYPER-PARAMETER STUDY, AND FURTHER DETAILS

A.1 BASELINES, EVALUATION, HYPERPARAMETERS ON PUBLICLY AVAILABLE DATASETS

Background on Normalizing Flows: *Normalizing Flows* (Rezende & Mohamed, 2015; Dinh et al., 2014; 2016; Papamakarios et al., 2017; Kingma & Dhariwal, 2018; Chen et al., 2019; Papamakarios et al., 2019) define a smooth, invertible transformation $f\colon \mathcal{X} \mapsto \mathcal{Z}$, of a simple base distribution $p(\mathbf{z})$ (e.g. an isotropic Gaussian) on the space $\mathcal{Z} = \mathbb{R}^D$ into a more complex distribution $p(\mathbf{x})$ on the space $\mathcal{X} = \mathbb{R}^D$ by a sequence of invertible and differentiable mappings. Its reverse operation $\mathbf{x} = f^{-1}(\mathbf{z})$ synthesizes realistic samples from the prior, is easy to evaluate and computing the Jacobian determinant takes $\mathcal{O}(D)$ time. Via the change of variables formula $p(\mathbf{x})$ can be expressed as $p(\mathbf{x}) = p(\mathbf{z}) \left|\det\left(\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}\right)\right|$.

Real-valued non-volume preserving (RealNVP) models introduce a *coupling layer*, which is the building block/bijection leaves part of its inputs unchanged $\mathbf{c}^{1:d} = \mathbf{x}^{1:d}$ and transforms the other part via functions of the untransformed variables $\mathbf{c}^{d+1:D} = \mathbf{x}^{d+1:D} \odot \exp(s(\mathbf{x}^{1:d})) + t(\mathbf{x}^{1:d})$. Here, $\odot$ is an element wise product, $s()$ is a scaling and $t()$ a translation function from $\mathbb{R}^d \mapsto \mathbb{R}^{D-d}$, given by neural networks. To model a nonlinear density map $f(\mathbf{x})$, a number of coupling layers which map $\mathcal{X} \mapsto \mathcal{C}_1 \mapsto \ldots \mapsto \mathcal{C}_{K-1} \mapsto \mathcal{Z}$ are composed together all the while alternating the dimensions which are unchanged and transformed. Via the change of variables formula the probability density function given a data point can be written as $\log p(\mathbf{x}) = \log p(\mathbf{z}) + \log|\det(\partial \mathbf{z}/\partial \mathbf{x})| = \log p_{\mathcal{Z}}(\mathbf{z}) + \sum_{i=1}^{K} \log|\det(\partial \mathbf{c}_i/\partial \mathbf{c}_{i-1})|$.

Baselines: The baselines considered for our experiments include: **1**: KVAE, combines a variational autoencoder with a linear state space model which describes the dynamics. **2**: Vec-LSTM-ind-scaling, models the dynamics via an RNN and outputs the parameters of an independent Gaussian distribution with mean-scaling. **3**: Vec-LSTM-lowrank-Copula, instead parametrizes a low-rank plus diagonal covariance via Copula process. **4**: GP-scaling, unrolls an LSTM with scaling on each individual time-series before reconstructing the joint distribution via a low-rank Gaussian. **5**: GP-Copula, unrolls an LSTM on each individual time-series and then the joint emission distribution is given by a low-rank plus diagonal covariance Gaussian copula. **6**: TCNF, uses LSTM to model the temporal conditioning and Masked Autoregressive Flow (Papamakarios et al., 2017) or Real-NVP for the distribution emission model. We also consider **7**: DeepVAR estimator, which is a multivariate variant of DeepAR (Salinas et al., 2019b).

Evaluation: To evaluate the quality of our forecasts, we compute CRPS and Normalized-MSE of the predictions. CRPS is a well-known metric for assessing the quality of probabilistic forecasts in the univariate case – it measures the the fit of the predictive distribution to the true one. CRPS is empirically evaluated using a samples generated from the predictive distribution, since we do not have a closed-form for the predictive distribution and CDF. Employing the empirical CDF of F, i.e. $\hat{F}(z) = \frac{1}{N_s}\sum_{i=1}^{N_s} \mathbb{I}\{Y_i \leq z\}$ with N_s samples $Y_i \sim F$ as a natural approximation of the predictive CDF, CRPS can be directly computed from simulated samples of the conditional distribution at each time point: $\mathrm{CRPS}(F, y) = \int_{\mathbb{R}} (F(z) - \mathbb{I}\{y \leq z\})^2 \,\mathrm{d}z$ where $\mathbb{I}\{y \leq z\}$ is the indicator function which is one if $y \leq z$ and zero otherwise.

Marginal CRPS is not applicable in the multivariate setting as it does not capture correlations across time-series. CRPS-Sum is an extension of CRPS to the multivariate case that can capture joint effects. CRPS-sum is the CRPS computed on the sum (across series) of the observed target values, yielding an estimate $\hat{F}_{\mathrm{sum}}(t)$ for each time point. Then CRPS-Sum is computed as $\mathrm{CRPS}_{\mathrm{sum}} = \mathbb{E}_t\left[\mathrm{CRPS}\left(\hat{F}_{\mathrm{sum}}(t), \sum_i y_{n,t}\right)\right]$. The MSE is the mean squared error over all time-series, i.e., $i = 1, \ldots N$, and over the whole prediction range, i.e., $t = T+1, \ldots, T = \tau$ with respect to the test data: $\mathrm{MSE} = \frac{1}{N\tau}\sum_{i,t}(y_{i,t} - \hat{y}_{i,t})^2$.

Training: Our algorithms are implemented in MXNet Gluon (Chen et al., 2015). To train the model, we use the Adam optimizer with the learning rate set to 0.0001. The network architecture

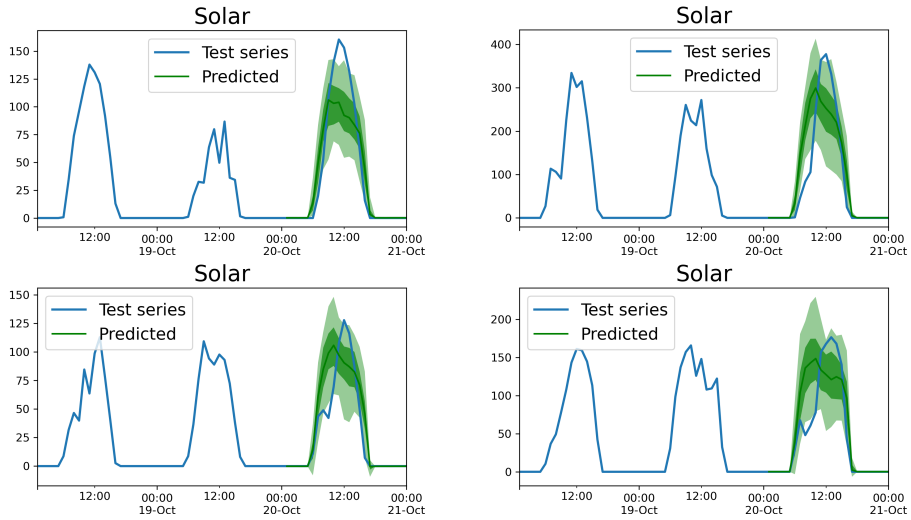

Figure 6: Example LatTe-Flows forecasts on Solar – We show the first 4 dimensions.

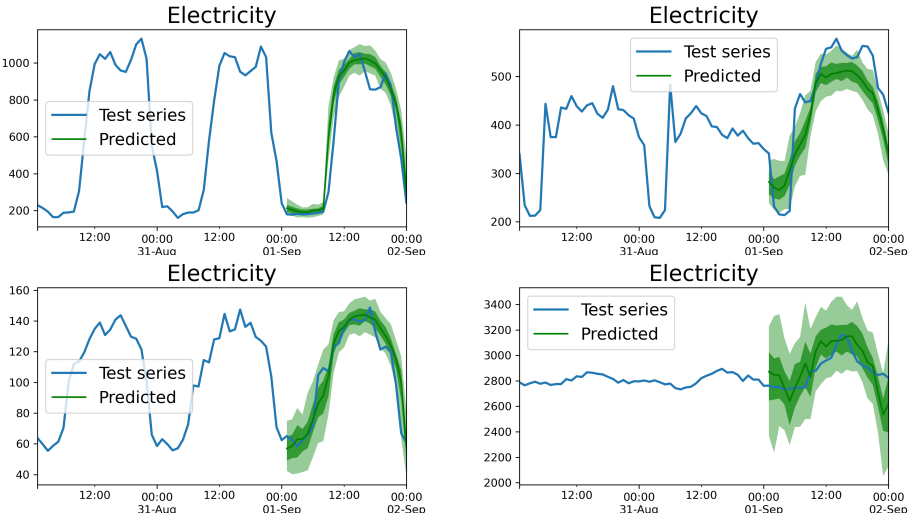

Figure 7: Example LatTe-Flows forecasts on Electricity – We show the first 4 dimensions.

of the encoder uses feed forward network (FNN) layers with ReLU activations. Details are set as follows:

The encoder architecture considered for training *LatTe-Flows* on **Solar** is $(N, 128) \to (128, 64) \to (64, 32) \to (32, D)$, with $D = 16$. The hyperparameter that balances the two losses is set to $\lambda = 0.01$ (see sample forecasts on Figure 6). For **Electricity** the encoder architecture is $(N, 256) \to (256, 128) \to (128, 64) \to (64, D)$, with $D = 32$. Here, λ is set to $\lambda = 0.001$ (see sample forecasts on Figure 7). We also provide the training loss components (and the overall loss) versus epochs on the traffic data. The scale of the orange curve displaying the loss of the probabilistic model is read from the right while other two is from the left. For **Traffic** the encoder architecture is $(N, 256) \to (256, 192) \to (192, 128) \to (128, 64)$, with $D = 64$. Here, λ is set to $\lambda = 0.0001$. For **Wiki** the encoder architecture is $(N, 256) \to (256, 128) \to (128, 64) \to (64, D) \to (64, D)$, $D = 32$. Here, λ is set to $\lambda = 0.01$. For **Taxi** is $(N, 256) \to (256, 192) \to (192, 128) \to (128, 64) \to (64, D)$, with $D = 32$. Here, λ is set to $\lambda = 0.01$.

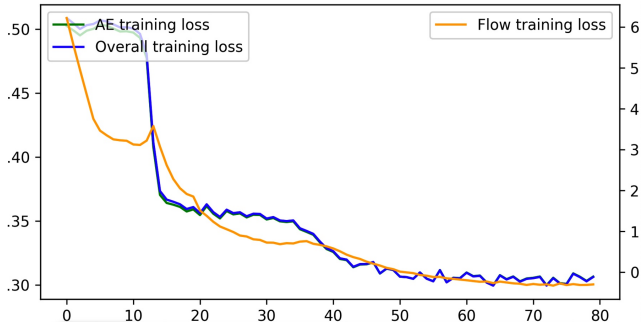

Figure 8: The training errors vs epochs for Electricity.

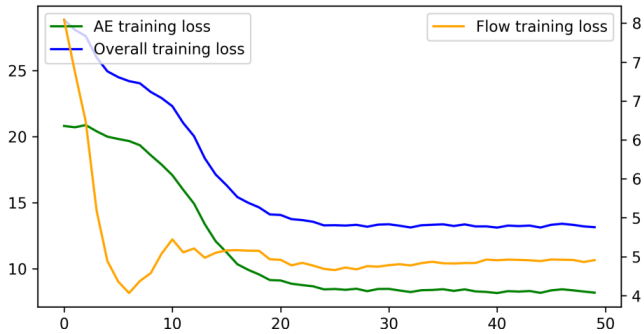

Figure 9: The training errors vs epochs for AH&MS.

A.2 AH&MS DETAILS AND FURTHER EXPERIMENTS

The encoder architecture considered for training *LatTe-Flows* on AH&MS is $(N, 6) \rightarrow (6, 6) \rightarrow (6, 6) \rightarrow (6, D)$ with $D = 4$, with ReLU. The hyperparameter that balances the two losses is set to $\lambda = 0.01$. The two componenets of the loss function as well as the overall error during training are shown in Figure 9. Signals in AH&MS are can have missing values. We impute each signal with the mean value of the variable across past observations. The histogram per signal is provided in Figure 10. In our forecasting visualizations we provide denote the true values of sensors as well as the imputed signal.

Next, we provide additional forecasts produced by our method (see Figures 11, 12, 13, 14) over four randomly chosen individuals. We present qualitative predictions for the 20 day forecast window produced by *LatTe-Flows* for AH&MS signal observations (HRV, RHR, SC, BBT, DBP, SBP) of an

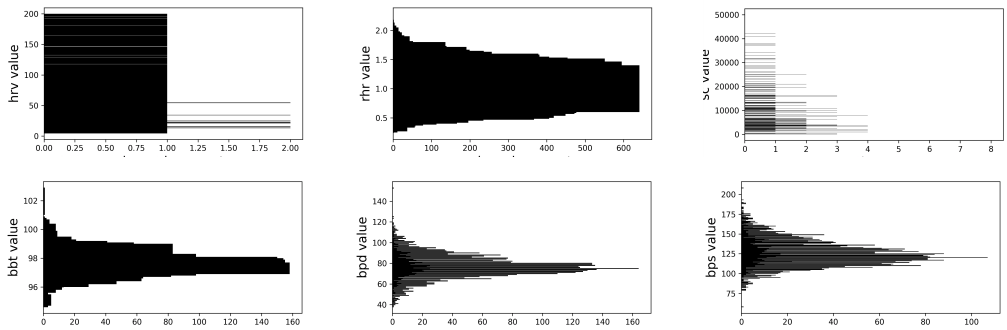

Figure 10: Histogram per signal.

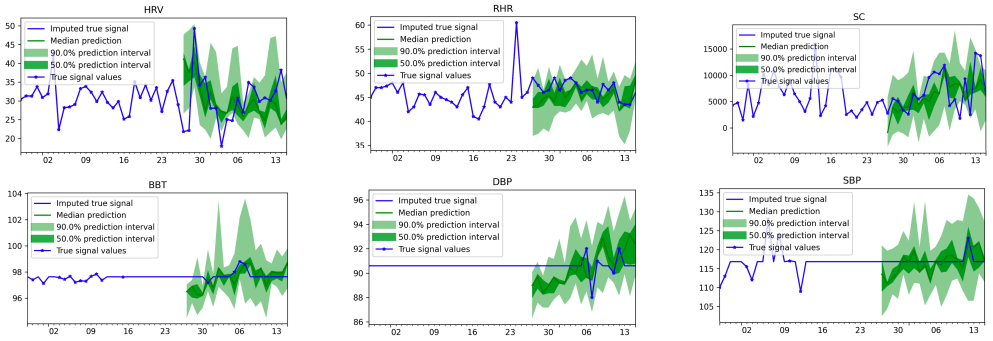

Figure 11: Predicted signals and test set ground-truth for a randomly selected subject (1st) in the study.

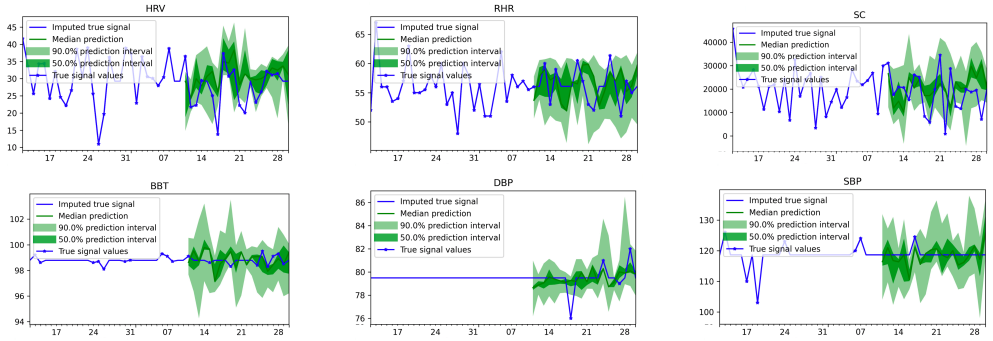

Figure 12: Predicted signals and test set ground-truth for a randomly selected subject (2nd) in the study.

individual. We estimate 10 trajectories for a latent dimension of $D = 4$ and plot the estimates. Note that the forecast is displayed in terms of a probability distribution: the shaded areas represent the 50% and 90% prediction intervals, respectively, centered around the median (dark green line).

The ground truth is overlaid in blue with stars denoting true observations and linear interpolation between the points. We also we provide forecasts (for the same testing window of length τ) produced by our method (Figure 15) compared with existing baselines (Figure 16).

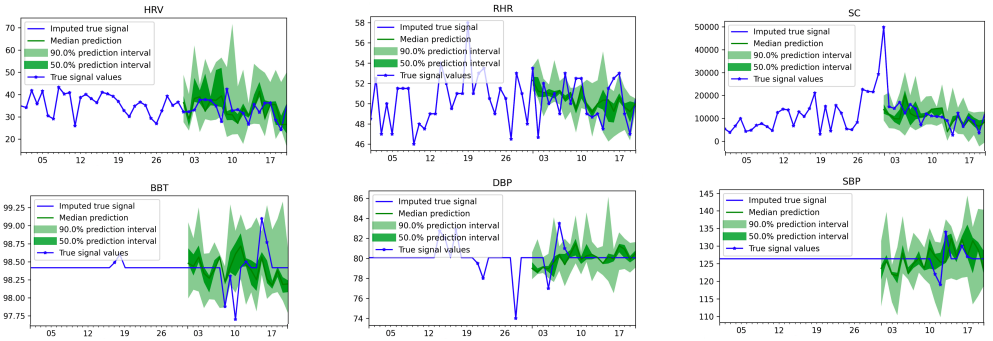

Figure 13: Predicted signals and test set ground-truth for a randomly selected subject (3rd) in the study.

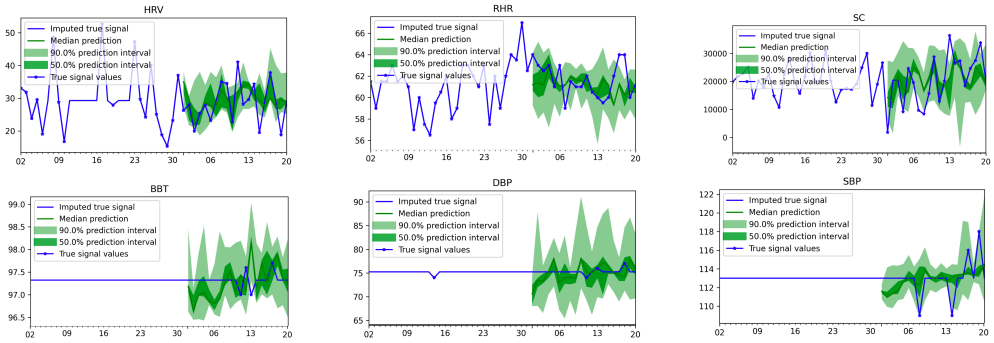

Figure 14: Predicted signals and test set ground-truth for a randomly selected subject (4th) in the study.

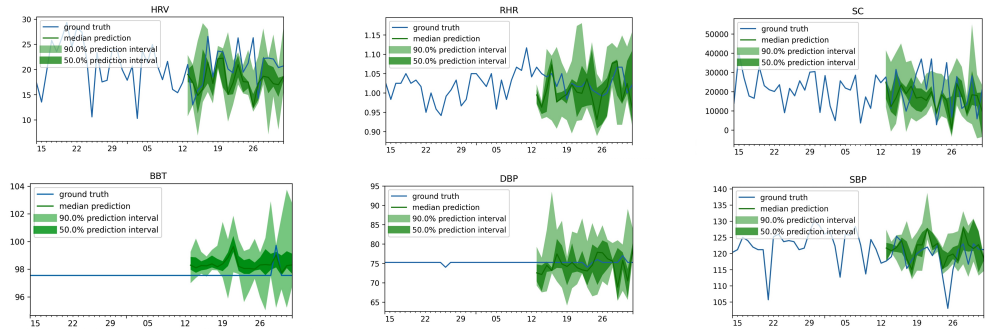

Figure 15: Qualitative predictions for the 20 day forecast window produced by *LatTe-Flows* for AH&MS signal observations (HRV, RHR, SC, BBT, DBP, SBP) of an individual. We estimate 10 trajectories for a latent dimension of $D = 4$ and plot the estimates.

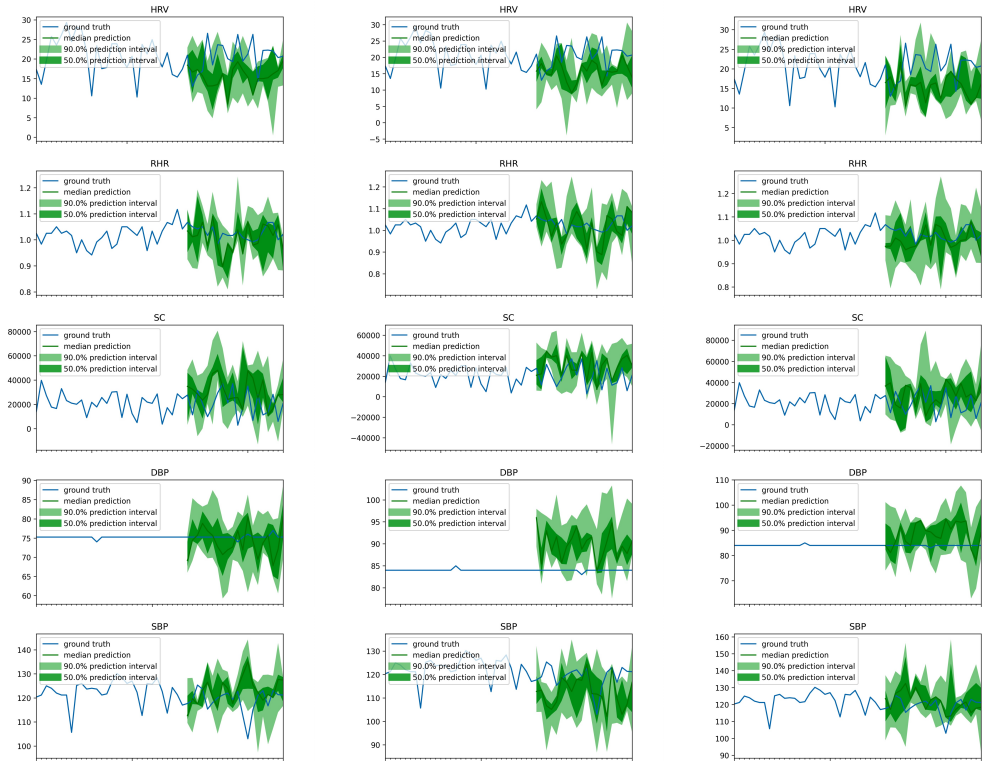

Figure 16: Signal modeling predictive performance for an individual in the test set. Baselines: Deep-VAR, GP-Copula, LSTM-MAF TCNF.