Forecasting with Sparse but Informative Variables: A Case Study in Predicting Blood Glucose

Anonymous

Abstract

In time-series forecasting, future target values may be affected by both intrinsic and extrinsic effects. When forecasting blood glucose, for example, intrinsic effects can be inferred from the history of the target signal alone (*i.e.* blood glucose), but accurately modeling the impact of extrinsic effects requires auxiliary signals, like the amount of carbohydrates ingested. Standard forecasting techniques often assume that extrinsic and intrinsic effects vary at similar rates. However, when auxiliary signals are generated at a much lower frequency than the target variable (*e.g.*, blood glucose measurements are made every 5 minutes, while meals occur once every few hours), even well-known extrinsic effects (*e.g.*, carbohydrates increase blood glucose) may prove difficult to learn. To better utilize these *sparse but informative variables* (SIVs), we introduce a novel encoder/decoder forecasting approach that accurately learns the per-timepoint effect of the SIV, by (i) isolating it from intrinsic effects and (ii) restricting its learned effect based on domain knowledge. On a simulated dataset pertaining to the task of blood glucose forecasting, when the SIV is accurately recorded our approach outperforms baseline approaches in terms of rMSE (13.07 [95% CI: 11.77,14.16] vs. 14.14 [12.69,15.27]). In the presence of a corrupted SIV, the proposed approach can still result in lower error compared to the baseline but the advantage is reduced as noise increases. By isolating their effects and incorporating domain knowledge, our approach makes it possible to better utilize SIVs in forecasting.

Introduction

In time-series forecasting, the future values of a target signal can depend on both intrinsic and extrinsic effects. Intrinsic effects are dynamics that depend only on the current and past values of the target signal. In contrast, extrinsic effects are dynamics that arise due to auxiliary variables. In many cases, the inclusion of such auxiliary signals as input to a forecasting model, in addition to the target signal, results in more accurate forecasts (Chakraborty et al. 1992; Rockwell and Davis 2016). However, in other settings, including auxiliary variables as input to a forecasting model produces little to no improvement in forecast accuracy, even when there is a known relationship between the additional variables and the target signal. This is particularly true in forecasting physiological variables like blood glucose. Auxiliary signals like carbohydrates consumed and bolus insulin administered both have well-known effects on blood glucose, but their inclusion as inputs to forecasting models has not, in general, led to significant improvements in performance over models based on blood glucose alone (Rubin-Falcone, Fox, and Wiens 2020; Hameed and Klienberg 2020; McShinsky and Marsha 2020). We hypothesize that this is due in part to a mismatch in the relative frequency of non-zero values between the auxiliary signal and the target signal. We refer to forecasting tasks where an auxiliary signal is sparse but has a known effect on the target signal as the *sparse but informative variable (SIV)* problem. A forecasting model that successfully addresses this problem will leverage the SIV despite its sparsity, leading to overall improved predictions.

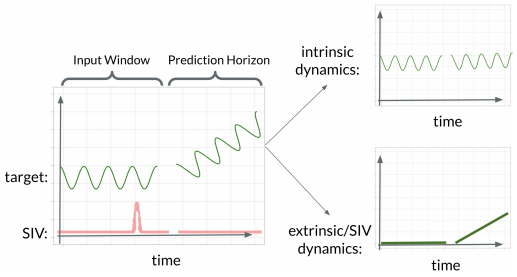

Figure 1: An overview of the SIV problem. In this toy example, the target variable exhibits oscillatory behavior when only zero SIV values are present (intrinsic dynamics), and the presence of a non-zero SIV value causes the target signal to increase linearly (extrinsic [SIV] dynamics).

Problem Definition. In this work, we introduce and address the SIV problem (**Figure 1**), which arises when an auxiliary variable that occurs infrequently is known to cause an increase or decrease in the target variable's magnitude over time, although the exact effect may be unknown. The sparsity of the SIV often results in the failure of standard multi-input forecasting approaches in leveraging the auxiliary variable, *i.e.*, models that include the variable perform similarly to univariate-input approaches. A model that has overcome the SIV problem utilizes the SIV in making its predictions, resulting in improved forecasting accuracy relative to a model that does not use the additional variable. The

SIV problem occurs when an important variable is mostly zero-valued. This is *not* the same as a sparsely *sampled* variable (SSV). In the case of under-sampling, the variable is sparse because it is not measured. In the SIV problem, we assume that the variable is measured frequently and accurately, but for most timepoints it is zero. We examine our setting as this assumption is relaxed and noise is added, but we consider SIVs to be generally noise-free, as noise and missingness pose a separate problem. While approaches for addressing missingness or SSVs have been extensively studied (Pratama et al. 2016), the SIV problem has not.

Challenges. Although developing forecasting strategies that make use of SIVs has the potential to improve predictive accuracy in several domains, it has not been directly addressed in previous work. Recent multivariate forecasting approaches attempt to learn complex inter-variable dependencies (Qin et al. 2017; Freiburghaus, Rizzotti-Kaddouri, and Albertetti 2020; Xu et al. 2020). However, these approaches do not explicitly account for the relative sparsity of some variables. Naive approaches to addressing the SIV issue include re-sampling the data so that more samples with non-zero SIV values are given to the network and carrying forward the SIV values to the end of the input window, but in practice we have found that these approaches generally fail to improve performance. As we will demonstrate, the incorporation of domain knowledge in terms of restricting model outputs could help encourage a forecasting model to make better use of the SIV. However, existing state-of-the-art deep forecasting approaches generally do not use such restrictions in order to maintain flexibility. Here, we strive for a combination of the two: a forecasting approach that maintains flexibility while incorporating domain knowledge.

Our Idea. To address the SIV problem we propose a novel forecasting approach: "The Linked Encoder/Decoder". Our model integrates two main ideas: (i) the isolation of intrinsic and extrinsic effects, and (ii) the incorporation of domain knowledge. We implement the first idea with two separate but connected decoder networks. One network learns per-timepoint SIV effects (the SIV network), and the other learns the intrinsic dynamics of the target variable (the target network). We implement the second idea by restricting the output of the SIV network based on domain knowledge. Combined, these ideas lead to overall improved usage of the SIV and in turn more accurate forecasts. Our main contributions are as follows:

- We present the sparse informative variable problem.
- We propose a novel forecasting approach designed to leverage the SIV by isolating the effect of the SIV and incorporating domain knowledge.
- We evaluate our model on type 1 diabetes (T1D) blood glucose data and show that it more effectively incorporates SIVs compared to several baselines, even in the presence of a small amount of noise.

Problem Setup

Here, we formalize our task and describe our motivating setting: blood glucose forecasting in type 1 diabetes.

Task Formalization. We focus on the task of multi-input univariate-output time-series forecasting in which we aim to predict the future values of a single target variable $x \in R$, but have access to an additional auxiliary variable $x' \in R$ that is sparse but informative. More specifically, x' is zero at a much higher frequency than the target signal and the presence of non-zero x' values has a known effect on the target signal (*e.g.*, they result in either an increase or decrease). Given data pertaining to the previous T values of the target signal, $\mathbf{x}_{-T+1:0}$, and the auxiliary signal $\mathbf{x'}_{-T+1:0}$, we aim to predict the next h timepoints of the target signal: $\mathbf{y} = \mathbf{x}_{1:h}$.

As is common in forecasting work (Chatfield 2000), we assume the target signal is generated by some underlying autoregressive process. We assume non-zero SIV values contribute in an additive autoregressive way. Our setup focuses on the setting in which extrinsic effects are driven by an SIV, but there are settings where extrinsic effects may be driven by a variable that is not sparse, or a combination of sparse and non-sparse variables. We focus on the SIV problem and assume that any additional non-sparse extrinsic effects can be modelled using standard forecasting approaches.

A Motivating Example- Predicting Blood Glucose. The SIV problem arises in blood glucose forecasting, which has been extensively studied in the past (Oviedo et al. 2016), including in deep learning settings (Rubin-Falcone, Fox, and Wiens 2020; Fox et al. 2018; Munoz-Organero 2020). Specifically, one aims to estimate blood glucose concentration (i.e., the target variable) for some prediction horizon into the future, based on a history of blood glucose and other signals. This represents a challenging forecasting task since glucose dynamics vary based on activity, time of day, hormone levels and more, resulting in significant non-stationarity throughout the day. Accurate models for blood glucose forecasting are critical to the development of algorithms for managing blood glucose in individuals with diabetes (one in ten people in the US). Individuals with type one diabetes require insulin injections to maintain healthy glucose levels throughout the day and especially around meals. Carbohydrates (meals) increase blood glucose, while insulin decreases blood glucose. However, current approaches do as well without information on carbohydrates or insulin as with (Rubin-Falcone, Fox, and Wiens 2020; Hameed and Klienberg 2020; McShinsky and Marsha 2020).

In this setting, insulin boluses and carbohydrates are considered SIVs, since they occur only a few times a day. In contrast, blood glucose is recorded every five minutes as unique, non-zero values. Carbohydrates and insulin result in an increase or decrease (respectively) of blood glucose after a delay of 30 minutes to an hour. However, the effects of these variables are not always long lasting. As a result, blood glucose forecasters can learn to ignore these variables while still generating accurate predictions for most timepoints. Still, models that utilize these variables will perform more accurately during critical changes in blood glucose.

Methods

Overview. To effectively capture the autoregressive dynamics of forecasting with an SIV, our architecture, the "Linked Encoder/ Decoder", relies on a recursive framework (**Figure 2**). It involves one encoder network and two linked decoder networks, which are used to isolate the SIV dynamics from

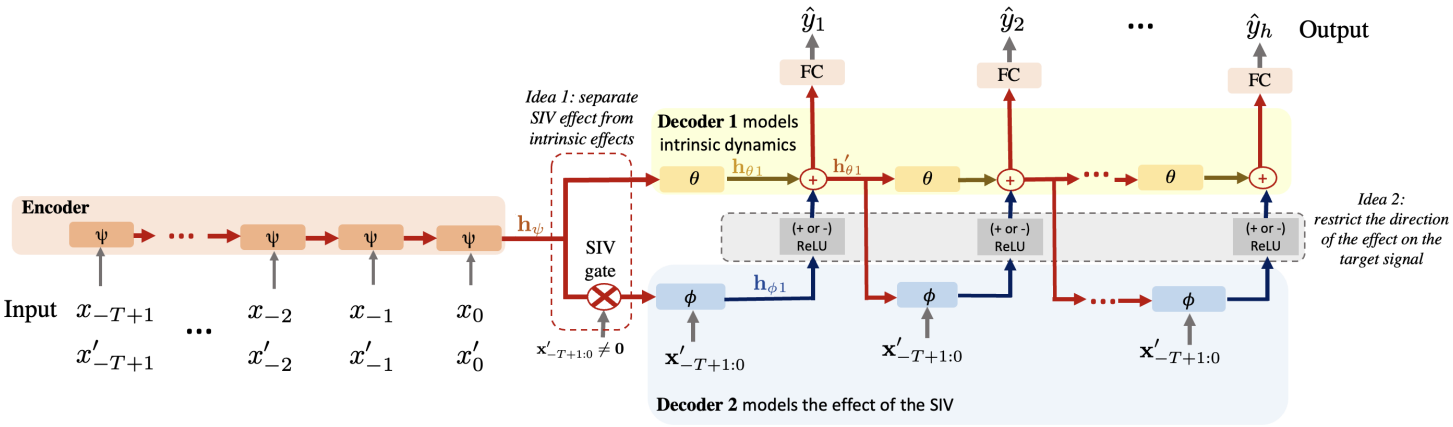

Figure 2: Our architecture: the Linked Encoder/Decoder, shown with an input length T and prediction horizon h. The θ network models intrinsic dynamics, while the ϕ network models the SIV dynamics. The ψ network models shared dynamics. Input time-series are gated, such that only inputs ($[\mathbf{x}_{-T+1:0}, \mathbf{x}'_{-T+1:0}]$) containing a non-zero SIV at any timepoint are passed through the ϕ network. The ReLU network shown in grey is used to ensure that the relationship between SIV and target is as expected.

the intrinsic dynamics. The SIV signal is input directly into the SIV decoder. The decoder systems are linked through a shared hidden state, which is processed in parallel, so that the intrinsic and extrinsic dynamics can be learned separately. Once isolated, we restrict the direction of the effect of the SIV on the target signal based on domain knowledge.

Standard Encoder/Decoder. Our approach is based on a standard encoder/decoder recurrent neural network, as depicted by the orange and yellow sections of **Figure 2**. For samples where $\mathbf{x}'_{-T+1:0} = \mathbf{0}$, this encoder/decoder is not modified. A single encoder (ψ), takes $\mathbf{x}_{-T+1:0}$ and $\mathbf{x}'_{-T+1:0}$ as input. The encoder outputs a hidden state $\mathbf{h}_\psi = \psi([\mathbf{x}_{-T+1:0}; \mathbf{x}'_{-T+1:0}])$, which is passed through a decoder LSTM (θ) that outputs hidden state $\mathbf{h}_{\theta 1} = \theta(\mathbf{h}_\psi)$. At each timepoint in the forecast horizon, the output from the previous time step $\mathbf{h}_{\theta t-1}$ is passed through θ, such that $\mathbf{h}_{\theta t} = \theta(\mathbf{h}_{\theta t-1})$. This learned representation is also passed through fully connected output network FC at each time step t in the forecast horizon to output a prediction $\hat{y}_t = FC(\mathbf{h}_{\theta t})$.

Linked SIV Decoder and Gating. For input samples that contain a non-zero SIV value, we augment this standard encoder/decoder with a second decoder ϕ that aims to model SIV dynamics, depicted in the blue section of **Figure 2**. By gating the output of the encoder based on the SIV values, we separate the extrinsic effects of the SIV on the target variable from the intrinsic effects of the target signal on itself. When the corresponding SIV values are non-zero (*i.e.*, $\mathbf{x}'_{-T+1:0} \neq \mathbf{0}$), the network engages the second decoder, which processes hidden state $\mathbf{h}_{\theta t}$ for $t = 1, ..., h$, in parallel with θ, as described below. Because ϕ is only engaged when an SIV is present, θ learns to forecast in the absence of an SIV, while ϕ learns the effect of the SIV.

SIV Decoder. In order to encourage the SIV decoder to utilize the SIV, the decoder also receives the entire SIV signal as input. We shift the SIV signal at each timepoint so that the encoder's position in time relative to the SIV is included in the representation implicitly (see implementation details).

We incorporate knowledge regarding how the SIV affects the target variable in processing the output of ϕ at each time step. We pass $\mathbf{h}_{\phi t}$ (the output of ϕ) through a ReLU function after it is output by ϕ, before adding it to $\mathbf{h}_{\theta t}$ and passing the shared hidden state to subsequent time steps and the output network. If the SIV is expected to lead to a decrease in the target signal, $\mathbf{h}_{\phi t}$ is multiplied by -1 after it is passed through the ReLU function. This restricts the effect of the SIV on the target variable to the expected direction.

Linked Hidden State Processing. Both decoders process a single hidden state in parallel, and their outputs are summed, after restriction has been applied to the output of the SIV decoder. At the first time step in the prediction horizon, both decoders take as input $\mathbf{h}_\psi$, but they each output a unique hidden state ($\mathbf{h}_{\theta t}$ and $\mathbf{h}_{\phi t}$). At subsequent time steps, these two hidden states are summed to create a new hidden state (*i.e.*, as illustrated in Figure 2; we define: $\mathbf{h}'_{\theta t} = \mathbf{h}_{\theta t} + \mathbf{h}_{\phi t}$), for $t > 0$. This combined hidden state ($\mathbf{h}'_{\theta t}$) is passed to the output network (FC) and both decoders for the next step. The final forecast $\hat{\mathbf{y}}$ is a sum of the outputs of the two decoders capturing both intrinsic and restricted extrinsic effects (*i.e.*, $\hat{y}_t = FC(\mathbf{h}'_{\theta t})$). Note that when $\mathbf{x}' = \mathbf{0}$, ϕ is not engaged, and $\mathbf{h}'_{\theta t} = \mathbf{h}_{\theta t}$.

Additional Variables. In **Figure 2** we present an overview of the proposed architecture for a setting with a single SIV. In the setting of multiple SIVs, one would increase the number of secondary decoders and apply restrictions according to the known effect of each SIV. Each ϕ would take as input only the relevant SIV signal, along with $\mathbf{h}'_{\theta t}$. $\mathbf{h}'_{\theta t}$ is modified by all SIV decoder systems, so that the hidden state that is passed to FC and subsequent decoder steps is a sum of the number of SIVs plus one components. In addition, non-sparse auxiliary variables, if any, are given to the ψ network, along with $\mathbf{x}$ and $\mathbf{x}'$, so that non-SIV extrinsic effects can be modeled (by θ, since these variables do not effect gating). The θ network must perform well in the absence of non-zero SIV signals, thus the ϕ network is en-

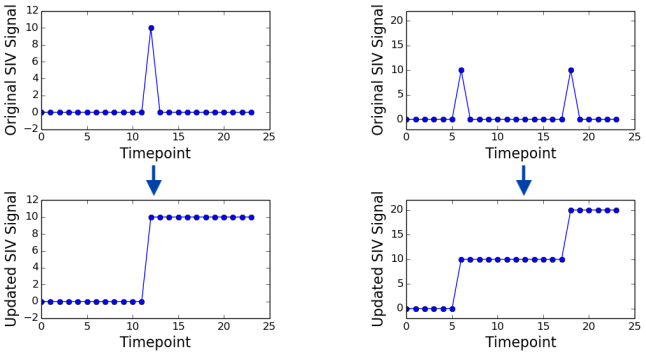

Figure 3: Our sum-total approach. We use the sum total up to the current point within an input window as input. This method allows the SIV signal to make a larger impact on the gradient while maintaining all temporal information.

couraged to learn the remaining extrinsic effects for samples with non-zero SIV values (which must be the SIV effects).

SIV Representation. One issue that makes utilizing SIVs difficult is that they usually occur at only one timepoint in the input window, having little effect on the gradient. To increase its effect, we use the sum-total SIV value up to the current timepoint as input (**Figure 3**). Up until the first non-zero SIV value of an input time-series, the signal value is zero. After any non-zero SIV, the input is the sum of observed values prior to and including that point, *in the input time-series only*. SIV values from before the input window are ignored. This approach was found to improve performance during baseline tuning and is used for all analyses, including baselines and ablations. This sum-total method improved performance for all approaches (see **Appendix B**).

Experimental Setup

We evaluate our approach on several datasets pertaining to blood glucose forecasting in T1D. We compare to several baselines, including resampling approaches. We evaluate forecast accuracy and the extent to which each model utilizes the SIVs.

Forecasting Task

We aim to forecast blood glucose values 30 minutes into the future ($h = 6$), based on a history of blood glucose and two sparse but informative variables: carbohydrate and insulin bolus values from the past 2 hours ($T = 24$). Here, $h = 6$ as it represents a common BG forecasting benchmark (Marling and Bunescu 2018) and we set T=24 based on prior work that suggests longer histories do not provide additional benefit (Oviedo et al. 2016). Bolus insulin and carbohydrate values are considered SIVs. All variables were scaled to between zero and one. Each individual's data were split into overlapping windows of length $T + h$ with a stride of 1, to be used as model input and labels.

Data

We compare the performacne of our architecture to baselines on three T1D-based datasets: Simulated, Simulated-noisy, and Ohio. All three datasets are publicly available (Man et al. 2014; Xie 2018; Marling and Bunescu 2018).

Simulated. Data generated from a commonly-used T1D simulator provide a curated test setting on which to evaluate our approach. We used the UVA-Padova simulator (Man et al. 2014), via a publicly available implementation (Xie 2018). We generated ten days of data for ten individuals (the ten "adult" patients modeled in the simulator), corresponding to 28,800 timepoints. Carbohydrate and insulin values occurred every 111 timepoints on average (Carbohydrates median, [IQR]: 84 timepoints between occurrences, [60,150]. Boluses: 83, [58,148]). The meal schedule used to generate simulated data was based on the Harrison-Benedict equation (Harris and Benedict 1919) as implemented in (Fox et al. 2020), but without snacks (3 meals a day), to further highlight the SIV problem. We used the default basal-bolus controller from the existing implementation to administer insulin, but we delayed three quarters (randomly selected) of the bolus administrations to be 20 minutes to two hours after the simulated meal, randomly sampled from a uniform distribution. This delay was to control for a separate issue that can arise when multiple SIVs occur at precisely the same time. The delay disentangles their effects.

Simulated - noisy. While the simulator used above introduces noise in the blood glucose measurements, we assume that the SIV signals are recorded without noise or missingness. To measure the robustness of our approach to this assumption, we generate additional simulated datasets using the approach above in which we vary the amount of noise and missingness. After data generation, we zero out between 10% and 50% of the carbohydrate values. Separately, we also explore the effects of adding uniform random noise to the measurements, varying the maximum magnitude between 10% and 50%. These changes are made to both training and testing data.

Ohio. Finally, we examine performance on a real dataset that was made publicly available for the Knowledge Discovery in Healthcare Data Blood Glucose Level Predication Challenge (Marling and Bunescu 2018). The data pertain to 12 individuals, with blood glucose measurements every 5 minutes. Carbohydrate administrations occur every 88 timepoints on average, (median, [IQR]: 70, [56,134]), and insulin boluses every 52 timepoints on average (36, [28,63]). Carbohydrate measurements are input by the individual and as a result are subject to noise and missingness. However, unlike the 'Simulated - noisy' dataset, we cannot directly measure nor control the extent of the noise in our experiments. Further details are provided in **Appendix A**.

Baselines

Encoder/Decoder. Our primary baseline is a stand-alone encoder/ decoder system, identical to the ψ plus θ networks in our full architecture (Fox et al. 2018). Due to the smaller capacity compared to our proposed approach, we increase the minimum number of training iterations so that we match the same number of gradient updates as our approach. We also examine a model with capacity that matches our proposed approach (Full Capacity).

Full Capacity. To ensure that any performance improvements observed are not due to our model's increased capacity, we also compare to a model based on our full architecture, but with no SIV-specialization (*i.e.*, there is no gating, no direct SIV input into ϕ, and no output restriction). This model is trained for the same number of iterations as our proposed approach.

Resampling. Resampling is perhaps the simplest common-sense approach to addressing signal sparsity. In order to rule out re-sampling as a naive solution to the SIV problem, we implement our primary baseline method with two re-sampling procedures: training the model on only windows with SIV samples to initialize the weights before training on the full sample (**SIV Initialize**), and training on the full sample, then fine-tuning the model on only windows with non-zero SIV values (**SIV Fine-tune**). Similar to our primary encoder/decoder baseline, we increase the minimum number of training iterations to match our complete method's number of gradient updates.

Implementation and Training Details

Implementation. Each LSTM encoder or decoder is implemented as a 2-layer bidirectional LSTM with 100 hidden units. FC is a fully connected linear network with a single output. Our architecture uses two ϕ networks, one for carbohydrates (positive effect, reLU restriction) and one for bolus insulin (negative effect, $-1\times$ reLU restriction). In order to input each SIV signal into each ϕ network while maintaining time information, $\mathbf{x}'_{-T+1:0}$ is front-padded with h zeros and input to ϕ at the first time step of the forecast horizon. The signal is shifted back at each timepoint, such that at the i^{th} time step of the prediction horizon, the SIV signal is shifted back $i-1$ positions, so that it is front-padded with $h-(i-1)$ zeros and back-padded with $i-1$ zeros. In this way the input corresponds with the encoder's position in time. $\mathbf{x}'_{-T+1:0}$ is scaled to have the same mean as $\mathbf{h}'_{\theta t}$ for each input.

Training. We split each individual's data into training, validation and test sets used for evaluation purposes. We use a 70%/15%/15% split. We implemented and trained our models in pytorch 1.9.1 and CUDA version 10.2, using Ubuntu 16.04.7 and a GeForce RTX 2080, using an Adam optimizer (Kingma and Ba 2014) and a batch size of 500. We used a learning rate of 0.01 and a weight decay of 10^{-7}. When training, mean square error (MSE) across all timepoints in the prediction horizon was used as a loss function. We trained for at least 500 epochs, until performance did not improve for 50 epochs. Parameters found at the iteration for which the model performed best on the validation data were used at inference time. We train and test a model on each subject and report across-subject averages.

Reproducibility. Our code and simulated dataset will be made publicly available and are uploaded with this submission. The Ohio dataset can be made available through a data-use agreement with the owners (see appendix).

Evaluation

All evaluations were performed on held-out test sets. We measured forecasting performance using rMSE and mean absolute error (MAE). In order to match common practice in the blood glucose forecasting literature, we calculated error terms based on the prediction accuracy of the final timepoint in the prediction window (Marling and Bunescu 2018).

SIV Usage Metric. Let X denote a dataset with an SIV, and let $X_\emptyset$ denote the exact same dataset with all SIV values set to zero. Let f define a mapping $f : X \rightarrow \hat{\mathbf{y}}$, where $X \in X$, and $\hat{\mathbf{y}} \in R^h$ is a prediction of the next h points of the target variable. f is trained and evaluated on X, while $f_\emptyset$ is trained and evaluated identically to f, except using $X_\emptyset$. Let L denote the error of the model's prediction (here, rMSE or MAE). We define SIV usage as $L(f_\emptyset(X_\emptyset))$ - $L(f(X))$. It is inspired by the Shapley Regression Value (Lipovetsky and Conklin 2001). This metric reflects how error changes when the SIV is removed. When removing the SIV, we both train and test on data without the SIV (rather than performing a permutation test or similar), so the model can learn the maximum amount of information available from the target variable alone.

Individual-Level Analyses. We compare the error and SIV usage of the baseline encoder/decoder to the improvement over baseline offered by our method across individuals. We expect that our model will offer greater improvement over baseline for individuals with high baseline error and low baseline SIV usage, as those are the individuals for which SIVs are most poorly modeled in the baseline. This would indicate that our approach addresses baseline deficits in SIV-modelling.

Ablations. We perform the following ablation analyses to examine which elements contribute to our models performance. **No Gating**: All samples are passed through both decoders. **No Restriction**: The outputs of the SIV decoder systems are not passed through a ReLU function. **No Dec. SIV Input**: The Decoder receives only the hidden state as input, and not the SIV signal. **Only Dec. SIV Input**: The baseline model is used, but with the modification that the SIV is input to the decoder directly, as in our model.

Noise and Missingness. We assume that SIV signals are noise free. Here, we evaluate our model's performance as this assumption is relaxed. This is possible in our simulated dataset because we have ground truth carbohydrate values. In real data, not only may the magnitude of these values be inaccurate (they are estimated by the patient), but patients may skip recordings altogether. In order to examine how unreliable carbohydrate values impact our approach, we randomly hide between 10% and 50% of the carbohydrate values (by setting their value to zero), and also add between 10% and 50% magnitude uniform random noise to the carbohydrate measurements (both after data generation, and to both training and testing data), to evaluate the performance of our model vs the stand-alone encoder/decoder baseline as carbohydrate values become unreliable. We report average forecast error across all individuals and 5 random noise-generation seeds for this analysis. We also evaluate our approach on a real dataset which is expected to have some degree of missingness and noise for comparison.

Results and Discussion

We aim to answer the following questions.

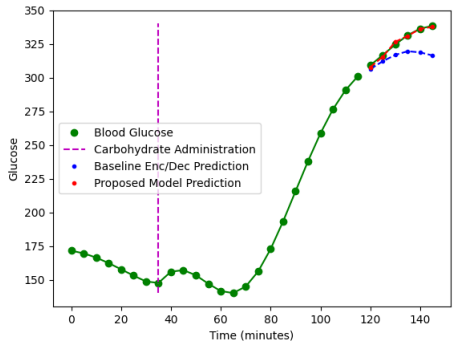

Figure 4: A sample prediction for a simulated individual (adult#004). Our model better accounts for the steep rise in the blood glucose signal following a meal.

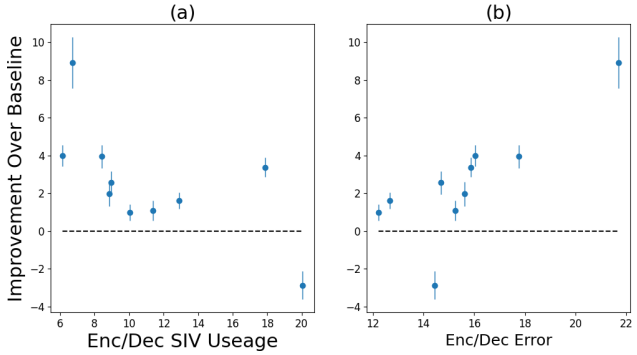

Figure 5: (a) Our architecture's improvement over the encoder/decoder baseline vs baseline SIV usage. Our method's benefit increases as baseline SIV usage decreases. (b) Improvement over baseline vs baseline error. Improvement over baseline is positively correlated with baseline error.

• Does our model offer improvement over baseline approaches in terms of forecast error and SIV Usage?

• Across individuals, when does our model offer the greatest improvement?

• What elements of our model improve performance?

• How is our model impacted when the SIV is missing at times or inaccurately recorded?

Improvement Over Baselines. Across patients, our model outperforms baselines leading to lower average rMSE/ MAE and greater relative SIV usage (**Table 1, top**). Specifically, our model appears better able to account for the effect of the SIV on the target signal (e.g., accurately predicting sharp rises that the baseline encoder/decoder cannot account for**Figure 4**). Naive SIV re-sampling approaches are generally outperformed by other baseline approaches, or perform similarly. Our approach shows a large improvement over even the strongest baseline (rMSE 13.07 vs 14.14, paired t-test across individuals [t statistic/p-value]: 3.6/0.002). In general, the increased SIV usage of our proposed approach corresponds to lower error, demonstrating its informativeness.

Table 1: Average forecasting error and SIV usage. Outcomes are reported as: Error [95% confidence interval] (SIV Usage). Our proposed approach outperforms baselines and ablations. 1,000 re-sample bootstraps were used for CIs.

Model	rMSE [95%CI] (Usage)	MAE [95%CI] (Usage)
Encoder/Decoder	15.63,[14.08,16.89] (11.13)	12.42,[11.14,13.59] (6.63)
SIV Fine-tune	27.30,[24.66,29.09] (-0.54)	22.22,[19.9,23.89] (-3.17)
SIV Initialize	15.37,[13.63,16.86] (11.39)	11.99,[10.68,13.17] (7.06)
Full Capacity	14.14,[12.69,15.27] (12.62)	11.21,[9.97,12.24] (7.85)
Proposed	13.07,[11.77,14.16] (13.69)	10.45,[9.37,11.37] (8.61)
No Gating	13.93,[12.57,15.04] (12.84)	11.11,[9.94,12.11] (7.95)
No Restriction	13.12,[11.8,14.21] (13.64)	10.43,[9.31,11.38] (8.62)
No Dec. SIV Input	14.20,[12.67,15.36] (12.56)	11.18,[9.96,12.23] (7.87)
Only Dec. SIV Input	13.97,[12.58,15.10] (12.79)	11.12,[9.96,12.15] (7.94)

Individual Level Results. We consider ten different individuals who differ in terms of simulated physiological parameters. Here, we investigate how model performance with respect to the baseline Encoder/Decoder varies across individuals. In particular, we identify settings in which our approach is more beneficial.

Our model's benefit over baseline varies inversely with the extent to which the baseline approach relies on the SIV (*i.e.*, SIV usage) across individuals (Pearson r=-0.65, p=0.041, **Figure 5 (a)**). This supports the hypothesis that our model's improved performance over the baseline is due in part to the increased usage of the SIV. For individuals for whom the baseline model was able to achieve high usage, our model was not necessary, while individuals with low usage stood to benefit. We also observe a strong correlation between baseline error and our approach's improvement (r=0.80, p= 0.0056, **Figure 5 (b)**). This suggests that our approach addresses the deficits of the baseline at the individual level, decreasing variation in the error across individuals. The higher variability in the performance of the baseline across individuals compared to the proposed approach (range: 6.3 vs 9.5) may be due in part to difficulties in SIV modeling, for which our model compensates.

Ablations. Ablation analyses reveal that, in general, our approach's strong SIV usage and forecast accuracy are a result of the combined effect of each implementation detail, rather than any one component. **Table 1, bottom** shows the results of our ablation study: removing any component results in a decrease in performance accuracy and SIV usage. Notably, removing the domain-guided restriction *i.e.,* when the ReLU functions are not included, has the smallest effect on performance. This is likely because, in our Simulated dataset, the effect of the insulin boluses and carbohydrate administrations are strong enough that the model can learn them easily without supervision. We expect this component to have a greater effect in situations where the impact of the SIV is more subtle.

Noise and Missingness. In the above experiments, we assume the SIV is accurately measured. To measure the impact of noise in this measurement on the benefit of our approach, we perturb the SIV as described in the experimental setup section. We find that as missingness and noise increase, our

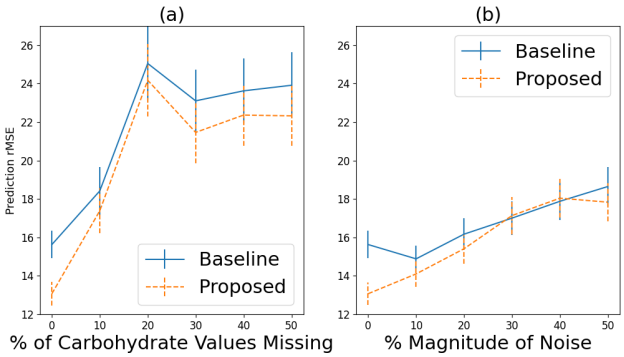

Figure 6: Average model performance across all individuals as (a) simulated carbohydrate values are hidden and (b) noise is added to carbohydrates. As noise and missingness increase, our model fails to reliably outperform the baseline. Error bars represent 100 re-sample bootstrap standard errors across all seeds and subjects.

approach's performance degrades (**Figure 6**). Our approach is more impacted by corrupted SIV values relative to the baseline, in part because of the increased dependence on the SIV (*i.e.*, greater SIV usage). As expected, completely omitting carbohydrates has a greater effect than simply corrupting the magnitude. Though performance decreases with increased noise/missingness, we are encouraged by the fact that our proposed approach remains competitive with the baseline.

We further explore the effects of a corrupted SIV signal using the 'Ohio' data, a real dataset generated by humans with type 1 diabetes. While we cannot measure the amount of noise in the Ohio dataset, we expect our approach will perform more similarly to how it does in the simulated-noise setting than the noise-free setting. Somewhat reassuringly, our approach performs no worse than existing approaches and in some cases even provides a small benefit over the baseline even in this noisy setting. Specifically, our approach consistently leads to lower forecast rMSE compared to all baselines, though performance gains are modest, rMSE=20.16 vs. the strongest baseline rMSE=20.36. Furthermore, in ablation analyses we found that the restriction component was beneficial for this dataset, supporting the hypothesis that domain knowledge insertion is beneficial for more challenging tasks (see **App. A** for full set of results).

Discussion & Conclusion

The SIV problem arises in forecasting domains when the relative sparsity of an auxiliary signal makes it challenging to learn its effect on a target signal. We introduce the problem and propose a forecasting approach that leverages SIVs. Our approach isolates SIV dynamics and restricts them based on domain knowledge, achieving higher SIV-usage and stronger forecasting performance than baselines. While our approach assumes accurately measured SIVs, reassuringly it performs no worse than baselines in the presence of missing or noisy SIV measurements. Though we focused on a specific use case for which we had a reliable simulator, we expect the SIV problem to arise frequently in healthcare. In such settings, SIVs are likely associated with time periods during which a patient is most vulnerable (*i.e.*, medication administration). Therefore, prediction models that address the SIV problem could lead to more accurate predictions during time periods that are critical for health outcomes.

We are the first to identify the SIV problem that arises when using RNNs for multi-input forecasting and the first to propose a solution. While sparsely sampled variables (SSVs) have been studied (Pratama et al. 2016), the SIV problem is distinct. Interpolation approaches for addressing missingness and noise in SSVs are not directly applicable to the SIV setting. Although the SIV problem has not yet been addressed, several techniques have been proposed to learn inter-variable relationships in forecasting tasks, which in part inspire our approach. In the context of multi-input forecasting, Pantiskas et al. and Qin et al. use attention mechanisms to identify which variables to focus on (Qin et al. 2017; Pantiskas, Verstoep, and Bal 2020). However, in contrast to our approach, these approaches do not account for signals that are mostly zero-valued, nor do they incorporate domain knowledge. Beyond attention based mechanisms, in a probabilistic setting, normalizing flows have been used to directly model the joint probabilities between variables (Rasul et al. 2020; Emmanuel de Bezenca an et al. 2020). However, SIVs are often too sparse to accurately estimate a joint probability. Several other approaches have been proposed to explicitly model inter-variable relationships (Gu, Dang, and Prioleau 2020; Freiburghaus, Rizzotti-Kaddouri, and Albertetti 2020; Pantiskas, Verstoep, and Bal 2020; Cao et al. 2020; Xu et al. 2020). However, none explicitly addresses the sparsity issue. Moreover, while some of these architectures separate the effects of variables, none use this isolation to restrict the effects based on domain knowledge as we do. There has been other work in forecasting that combines deep learning with domain knowledge to reduce the hypothesis space. However, researchers have relied on strong assumptions, *e.g.*, structuring deep architectures to match clinical intuition (Munoz-Organero 2020), combining deep approaches with physiological-model-based simulators (Miller, Foti, and Fox 2020), and estimating expert judgements on model outputs via Monte-Carlo approximations (Huanga, Qiaob, and Wang 2014). In contrast, we only restrict the sign of the SIV network's hidden state.

While there are many different ways to forecast signals, we focus on RNN-based techniques. Our primary contribution is the identification of the SIV problem in forecasting, and the failure of common RNN-based approaches to address this problem. We demonstrate how addressing the SIV problem can lead to improvements over directly comparable baselines. We do not claim SOTA in forecasting, but our findings could apply to many settings in which variants of RNNs are applied to forecasting problems with SIVs. The two main ideas behind our approach include gating and output restriction. While neither of these methodological developments are unprecedented on their own, their combined application to the SIV problem poses a novel direction for forecasting in related domains.

References

Cao, D.; Wang1, Y.; Duan, J.; Zhang, C.; Zhu, X.; Huang, C.; Tong, Y.; Xu, B.; Bai, J.; Tong, J.; and Zhang, Q. 2020. Spectral Temporal Graph Neural Network for Multivariate Time-series Forecasting. *Conference on Neural Information Processing Systems (NeurIPS)*.

Chakraborty, K.; Mehortra, K.; Mohan, C.; and Ranka, S. 1992. Forecasting the behavior of multivariate time series using neural networks. *Neural Networks*, 5: 961–970.

Chatfield, C. 2000. In *Time-Series Forecasting*. Chapman Hall/CRC.

Emmanuel de Bezenca an, S. S. R.; Benidis, K.; Bohlke-Schneider, M.; Kurle3, R.; Hasson, L. S. H.; Gallinari, P.; and Januschowski, T. 2020. Normalizing Kalman Filters for Multivariate Time Series Analysis. *Conference on Neural Information Processing Systems (NeurIPS)*.

Fox, I.; Ang, L.; Jaiswal, M.; Pop-Busui, R.; and Wiens, J. 2018. Deep Multi-Output Forecasting. *KDD*.

Fox, I.; Lee, J.; Pop-Busui, R.; and Wiens, J. 2020. Deep Reinforcement Learning for Closed-Loop Blood Glucose Control. *Proceedings of Machine Learning Research*.

Freiburghaus, J.; Rizzotti-Kaddouri, A.; and Albertetti, F. 2020. A Deep Learning Approach for Blood Glucose Prediction of Type 1 Diabetes. *International Workshop on Knowledge Discovery in Healthcare Data-KHD@IJCA*.

Gu, K.; Dang, R.; and Prioleau, T. 2020. Neural Physiological Model: A Simple Module for Blood Glucose Prediction. *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*.

Hameed, H.; and Klienberg, S. 2020. Investigating potentials and pitfalls of knowledge distillation across datasets for blood glucose forecasting. *International Workshop on Knowledge Discovery in Healthcare Data*.

Harris, J. A.; and Benedict, F. G. 1919. A biometric study of basal metabolism in man. *Carnegie institution of Washington*.

Huanga, A.; Qiaob, H.; and Wang, S. 2014. Forecasting Container Throughputs With Domain Knowledge. *Procedia Computer Science*, 31.

Kingma, D. P.; and Ba, J. 2014. Adam: A Method for Stochastic Optimization. *International Conference for Learning Representations*.

Lipovetsky, S.; and Conklin, M. 2001. Analysis of regression in game theory approach. *Applied Stochastic Models in Business and Industry*, 17.

Man, C. D.; Micheletto, F.; Lv, D.; Kovatchev, M. B. B.; and Cobelli, C. 2014. The UVA/PADOVA Type 1 Diabetes Simulator. *J Diabetes Sci Technol.*, 8.

Marling, C.; and Bunescu, R. C. 2018. The OhioT1DM Dataset for Blood Glucose Level Prediction. *International Workshop on Knowledge Discovery in Healthcare Data-KHD@IJCA*.

Marling, C.; and Bunescu, R. C. 2020. The OhioT1DM Dataset for Blood Glucose Level Prediction: Update 2020. *International Workshop on Knowledge Discovery in Healthcare Data-KHD@IJCA*.

McShinsky, R.; and Marsha, B. 2020. Comparison of Forecasting Algorithms for Type 1 Diabetic Glucose Prediction on 30 and 60-Minute Prediction Horizons. *International Workshop on Knowledge Discovery in Healthcare Data*.

Miller, A. C.; Foti, N. J.; and Fox, E. 2020. Learning Insulin-Glucose Dynamics in the Wild. *Machine Learning for Healthcare*, 126: 1–25.

Munoz-Organero, M. 2020. Deep Physiological Model for Blood Glucose Prediction in T1DM Patients. *Sensors*.

Oviedo, S.; Vehí, J.; Calm, R.; and Armengol, J. 2016. A review of personalized blood glucose prediction strategies for T1DM patients. *Int J Numer Method Biomed Eng*.

Pantiskas, L.; Verstoep, K.; and Bal, H. 2020. Interpretable Multivariate Time Series Forecasting with Temporal Attention Convolutional Neural Networks. *IEEE Symposium Series on Computational Intelligence*.

Pratama, I.; Permanasari, A. E.; Ardiyanto, I.; and Indrayani, R. 2016. A review of missing values handling methods on time-series data. *International Conference on Information Technology Systems and Innovation (ICITSI)*.

Qin, Y.; Song, D.; Cheng, H.; Cheng, W.; Jiang, G.; and Cottrell, G. W. 2017. A Dual-Stage Attention-Based Recurrent Neural Network for Time Series Prediction. *Joint Conference on Artificial Intelligence (IJCAI)*.

Rasul, K.; Sheikh, A.-S.; Schuster, I.; Bergmann, U.; and and, R. V. 2020. Multi-variate Probabilistic Time Series Forecasting via Conditioned Normalizing Flows. *International Conference on Learning Representations (ICLR)*.

Rockwell, P. J.; and Davis, R. A. 2016. Multivariate Time Series. In Rockwell, P. J.; and Davis, R. A., eds., *Introduction to Time Series and Forecasting*, 227–257. Springer Texts.

Rubin-Falcone, H.; Fox, I.; and Wiens, J. 2020. Deep Residual Time-Series Forecasting: Application to Blood Glucose Prediction. *International Workshop on Knowledge Discovery in Healthcare Data-KHD@IJCA*.

Xie, J. 2018. Simglucose v0.2.1 [Online]. Avaible: https://github.com/jxx123/simglucose. *Accessed on: Jan-20-2020*.

Xu, D.; Cheng, W.; Zong, B.; Song, D.; Ni, J.; Yu, W.; Liu, Y.; Chen, H.; and Zhang1, X. 2020. Tensorized LSTM with Adaptive Shared Memory for Learning Trends in Multivariate Time Series. *AAAI Conference on Artificial Intelligence*.

Appendix A: Ohio Dataset Experiments

Data and Training Description

This dataset includes both the OHIOT1DM 2018 and 2020 datasets, developed for the Knowledge Discovery in Healthcare Data Blood Glucose Level Predication Challenge (Marling and Bunescu 2020). The data pertain to 12 individuals, each with approximately 10,000 5-minute samples for training and 2,500 for testing, with carbohydrate administrations occurring every 88 timepoints on average, (median, [IQR]: 70, [56,134]), and insulin boluses occurring every 52 timepoints on average (36, [28,63]). 12% of glucose values are missing, but we do not include windows with missing glucose values.

This dataset contains the same variables and is processed and analyzed identically to the simulated dataset, except as described here. For the real dataset we evaluated on the held-out test data from the challenges. The remaining data were split into 80% train and 20% validation. Models were trained for at least 25 epochs, and then until validation data performance did not improve for 10 epochs.

The Ohio Dataset (Ohio T1D Blood Glucose Level Prediction Challenge, 2018 and 2020), can be made available through a data-use agreement with the owners: http://smarthealth.cs.ohio.edu/OhioT1DM-dataset.html.

Primary Results

On the Ohio dataset, performance gains are more moderate (rMSE 20.16 vs 20.36, **Table 2**), when compared to the simulated dataset. Multiple approaches exhibit negative SIV usage for the Ohio dataset, indicating that including the SIVs does more harm then good. We hypothesize that this is due to noise in the carbohydrate signal.

Table 2: Forecasting Error and SIV usage for the real dataset. Outcomes are reported as: Error [95% confidence interval] (SIV Usage). Our proposed approach outperforms baseline, although to a lesser degree than the simulated dataset. Confidence intervals were calculated from bootstraps with 1,000 re-samples.

Model	rMSE [95%CI] (Usage)	MAE [95%CI] (Usage)
Encoder/Decoder	20.36,[19.46,21.30] (0.08)	14.67,[14.11,15.24] (0.24)
SIV Fine-tune	21.74,[20.87,22.64] (-1.30)	16.25,[15.68,16.85] (-1.35)
SIV Initialize	20.98,[20.00,21.97] (-0.54)	14.99,[14.42,15.59] (-0.09)
Full Capacity	20.98,[20.04,21.92] (-0.54)	15.09,[14.5,15.69] (-0.18)
Proposed	**20.16,[19.28,21.06] (0.28)**	**14.64,[14.09,15.20] (0.27)**

Individual Level Results and Ablations

With respect to the Ohio data, while the overall trend was the same as the simulated data, across individuals, the correlation between baseline error and our approach's improvement over baseline was not significant (r=0.22, p=0.49, **Figure 7 (b)**). We hypothesize that this again might be due to the presence of noise in the carbohydrate signal, which prohibits our model from accurately modeling the SIV signal (explored in Section 5.5). Alternatively, the intrinsic dynamics in the Ohio dataset may simply be more complex and thus result in more variability across individuals. The association between baseline encoder/decoder SIV usage and improvement over baseline does hold for real data (**Figure 7** (a)), Pearson r=-0.59, p=0.042,

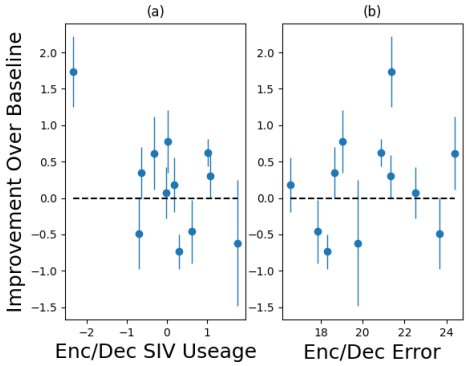

Figure 7: (a) Our architecture's improvement over the encoder/decoder baseline vs baseline SIV usage for the Ohio dataset. Our method's benefit increases as baseline SIV usage decreases. (b) Improvement over baseline vs baseline prediction error for Ohio data, for each individual. Improvement over baseline is not correlated with baseline error.

Table 3: rMSE and MAE, with SIV usage, for each ablation. Outcomes are reported as: Error [95% confidence interval] (SIV Usage). Confidence intervals were calculated from bootstraps with 1,000 re-samples.

Model	rMSE [95%CI] (Usage)	MAE [95%CI] (Usage)
	Ohio	
No Gating	20.34,[19.48,21.27] (0.11)	14.77,[14.22,15.33] (0.13)
No Restriction	20.38,[19.48,21.31] (0.06)	14.73,[14.16,15.32] (0.18)
No Dec. SIV Input	20.71,[19.81,21.64] (-0.27)	14.97,[14.41,15.55] (-0.07)
Only Dec. SIV Input	20.52,[19.62,21.47] (-0.08)	14.70,[14.14,15.29] (0.20)
Proposed	20.16,[19.28,21.06] (0.28)	14.64,[14.09,15.20] (0.27)

The restriction element is important for the Ohio dataset (**Table 3**, rMSE increases to 20.38 from 20.16 when restriction is removed). This is likely because this dataset presents a more difficult challenge, compared to the simulated dataset, due to noise in the SIV signal and more complex target variable dynamics. For the Ohio dataset, we see a decrease in performance for each ablation. Our architecture works by isolating the effect that the SIV signal has on the target variable and enforcing consistency with domain knowledge. Although the domain knowledge is very general (we only restrict the signal direction), it improves performance, offering a benefit over isolation alone for the Ohio dataset. More restrictive model guidelines, such as directly restricting the architecture to use a detailed physiological model, could be beneficial, but during model development, we found that "less is more," in that a small amount of restriction with significant flexibility was most effective. However, some sort of domain-knowledge-based-guidance is helpful to overcome the challenges posed by the SIV problem, since without it, it is difficult to learn anything useful from the small number of non-zero samples.

Appendix B: Impact of carry-forward approach

Utilizing the Carry-forward approach improves performance on both datasets for both the baseline encoder/decoder and our proposed approach (**Table 4**).

Table 4: Forecasting Error and SIV usage for both datasets, examining our primary baseline and proposed approach with and without our carry-forward approach. Outcomes are reported as: Error [95% confidence interval] (SIV Usage). Both methods benefit from utilizing the carry-forward approach on both datasets. Confidence intervals were calculated from bootstraps with 1,000 re-samples.

Model	rMSE [95%CI] (Usage)	MAE [95%CI] (Usage)
	Simulated- Carry Forward	
Encoder/Decoder	15.63,[14.08,16.89] (11.13)	12.42,[11.14,13.59] (6.63)
Proposed	**13.07,[11.77,14.16] (13.69)**	**10.45,[9.37,11.37] (8.61)**
	Simulated- NO Carry Forward	
Encoder/Decoder	16.46,[14.64,17.84] (10.30)	12.97,[11.53,14.13] (6.09)
Proposed	16.08,[14.46,17.37] (10.68)	12.80,[11.43,13.91] (6.25)
	Ohio- Carry Forward	
Encoder/Decoder	20.36,[19.46,21.30] (0.08)	14.67,[14.11,15.24] (0.24)
Proposed	**20.16,[19.28,21.06] (0.28)**	**14.64,[14.09,15.20] (0.27)**
	Ohio- NO Carry Forward	
Encoder/Decoder	20.64,[19.74,21.56] (-0.20)	14.98,[14.43,15.56] (-0.08)
Proposed	20.41,[19.53,21.35] (0.03)	14.85,[14.28,15.41] (0.05)