Proceedings of Machine Learning Research LEAVE UNSET:393–409, 2023 Conference on Health, Inference, and Learning (CHIL) 2023

Denoising Autoencoders for Learning from Noisy Patient-Reported Data

Harry Rubin-Falcone HRF@UMICH.EDU
Computer Science and Engineering, University of Michigan, Ann Arbor, MI, USA
Joyce M. Lee JOYCLEE@MED.UMICH.EDU
Division of Pediatric Endocrinology, University of Michigan, Ann Arbor, MI, USA
Jenna Wiens WIENSJ@UMICH.EDU
Computer Science and Engineering, University of Michigan, Ann Arbor, MI, USA

Abstract

Healthcare datasets often include patient-reported values, such as mood, symptoms, and meals, which can be subject to varying levels of human error. Improving the accuracy of patient-reported data could help in several downstream tasks, such as remote patient monitoring. In this study, we propose a novel denoising autoencoder (DAE) approach to denoise patient-reported data, drawing inspiration from recent work in computer vision. Our approach is based on the observation that noisy patient-reported data are often collected alongside higher fidelity data collected from wearable sensors. We leverage these auxiliary data to improve the accuracy of the patient-reported data. Our approach combines key ideas from DAEs with co-teaching to iteratively filter and learn from clean patient-reported samples. Applied to the task of recovering carbohydrate values for blood glucose management in diabetes, our approach reduces noise (MSE) in patient-reported carbohydrates from $72g^2$ (95% CI: 54-93) to $18g^2$ (13-25), outperforming the best baseline ($33g^2$ (27-43)). Notably, our approach achieves strong performance with only access to patient-reported target values, making it applicable to many settings where ground truth data may be unavailable.

Data and Code Availability Code and simulated data are available at tinyurl.com/ctdae4pat. Simulated data were generated with a publicly available implementation of a commonly used simulator of type-1 diabetes (Man et al., 2014; Xie, 2018). Real-world data came from the 2020 and 2022 Ohio BGLP challenges, which are publicly available (with a data-use agreement) (Marling and Bunescu, 2018, 2020).

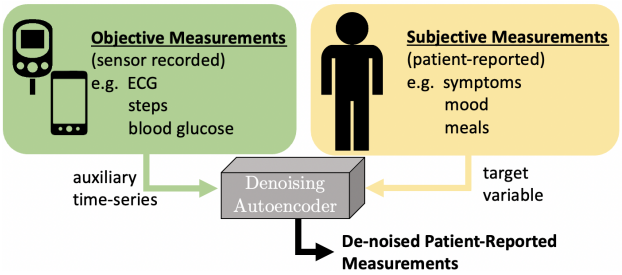


Figure 1: **Overview**. In our setting, given access to both subjective patient-reported data and higher-fidelity data from wearable sensors, we aim to denoise subjective measurements.

Institutional Review Board (IRB) This work is not regulated as human subjects research since data from the BGLP Challenge are stripped of identifiers.

1. Introduction

Motivation & Problem. With the increasing ubiquity of wearable sensor technology (e.g., fitbit), there has been an explosion in the number of studies seeking to correlate data from these technologies with patient-reported data (e.g., near-falls), with the goal of remote patient monitoring (Kious et al., 2019; Hauth et al., 2021). Patient-reported outcomes have been used in studies of cancer treatment (Nguyen et al., 2020), multiple sclerosis (D'Amico et al., 2019), diabetes (Wee et al., 2021), and mental health (McIntyre et al., 2022), and are also used to quantify patient-experience of care (Bull et al., 2019). However, patient-reported data are often noisy and dif-

ficult to validate (Churruca et al., 2021). The accuracy of these data may change day-to-day or even hour-to-hour (McKenna, 2011), making it challenging to detect meaningful changes over time (van der Willik et al., 2020). Moreover, in many cases, ground truth is difficult if not impossible to obtain. In light of these limitations, we aim to develop an approach that can denoise patient-reported data and increase their utility in downstream tasks (e.g., adverse outcome prediction). Our approach is based on the observation that while ground truth values for the target variable may be unavailable, other more reliable data streams (i.e., data collected from wearable sensors) are often collected alongside noisy patient-reported measurements. We hypothesize that these more reliable related data streams can help in recovering the noisier variables (**Figure** 1).

Healthcare Inspired Use Case. Throughout this work, we take inspiration from a specific real-world problem affecting millions in the US: blood glucose management. Individuals with diabetes monitor several variables over time, including their blood glucose, insulin administrations, and carbohydrate intake. Blood glucose, when measured by a continuous glucose monitor (CGM), has relatively little noise (Shah et al., 2018), while the amount of carbohydrates in a meal are patient-reported and subject to error (Brazeau et al., 2012; Mehta et al., 2009). Recognizing this variation in the level of noise across signals, we propose an approach that utilizes objective measurements (e.g., blood glucose) to update noisy patient-reported data. In the context of diabetes, retrospective correction of carbohydrates could help in downstream tasks: patients can learn when they are over- or under-reporting and adjust in the future, ultimately improving disease management. While we focus on challenges related to blood glucose management, our approach could apply to many other settings in which patient-reported data are collected alongside data from wearables: patient-reported near falls paired with data collected from inertial measurement units, mood scores paired with step count data collected from a fitbit, and self-reported symptoms paired with heart rate and other data (Kious et al., 2019; Hauth et al., 2021; Quer et al., 2020). As wearable technologies become increasingly prevalent (Smart, 2018; Samet, 2022), we expect this setting will only become more common.

Gaps in Existing Work. Denoising autoencoders (DAEs) (Vincent et al., 2008) have been used to accurately denoise signals, including medical images (Gondara, 2016), ECG signals (Xiong et al., 2016), and power system measurements (Lin et al., 2019). However, this approach generally requires access to both patient-reported measurements and ground truth measurements at training. In many real-world settings (including ours), only patient-reported target samples are available at training (we don't have access to paired ground truth patient-reported data). Work in computer vision has addressed this problem through extensions that either require paired noisy samples for each data point (e.g., multiple images of the same object) (Lehtinen et al., 2018) or rely on patch-based analysis (Krull et al., 2018; Laine et al., 2019; Xie et al., 2020; Batson and Royer, 2019). Similar approaches do not extend to patient-reported data, where paired samples rarely exist and patch-based techniques do not apply due to a lack of spatial feature dependencies. Others have proposed techniques that leverage knowledge of the noise distribution to recover the clean signal, but their applicability is limited in our setting, as noise for patient-reported variables is rarely weak or known (Kim and Ye, 2021; Moran et al., 2019; Xu et al., 2020). In contrast to prior work that has focused on missingness in patient-reported datastreams (including meal reports in diabetes management), we focus only on de-noising existing measurements.

Our Contributions. In light of these gaps, we adapt DAEs for patient-reported data. Our approach, 'Noise$^+$2Noise', learns to denoise a target signal (e.g., patient-reported meals) given only potentially noisy target samples (without access to ground truth) and an auxiliary clean signal (e.g., blood glucose measurements). Inspired by work in image denoising (Lehtinen et al., 2018; Xu et al., 2020), our approach augments existing DAEs with the auxiliary signal, leveraging the relationship between the auxiliary and target signals. In addition, we adapt a novel co-teaching approach from the noisy label literature (Han et al., 2018) to train two DAEs. Our approach works by iteratively selecting lower-noise target samples for training. Through a case study in blood glucose management, we demonstrate that our proposed approach can more accurately recover patient-reported data in the presence of noise compared to several baselines. Our contributions are as follows:

•We formalize an important problem in remote patient monitoring related to denoising patient-reported data.

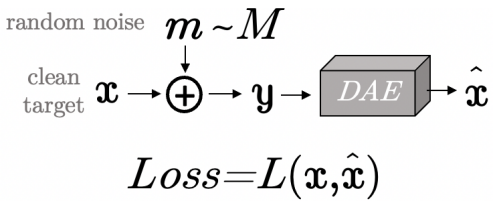


Figure 2: In vanilla DAE training, access to a clean target identified a priori to training ($\mathbf{x}$) is assumed. Noise ($\mathbf{m}$) is added to $\mathbf{x}$ to create noisy sample $\mathbf{y}$, and the DAE is tasked with reconstructing $\mathbf{x}$.

•We propose a novel approach based on DAEs that leverages an auxiliary low-noise signal to denoise a target variable without access to ground truth target data.
•We demonstrate improved denoising of carbohydrate values for blood glucose management compared to baselines in a simulated data setting.
•We propose and validate a proxy measure for evaluating carbohydrate denoising when ground truth is unavailable. Our approach outperforms baselines on a real-life dataset based on this metric.

2. Background and Related Work

Our approach takes inspiration from work in DAEs, commonly used in image denoising, and work in in noisy label learning. Below we briefly provide background on these topics.

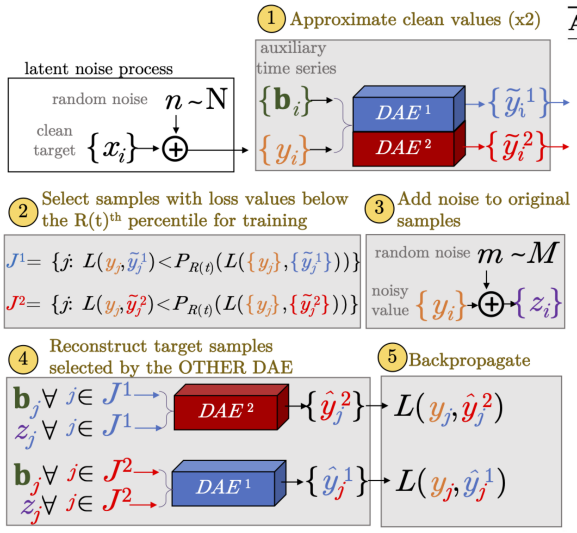
In DAE training, a model input is corrupted and a network is tasked with recovering the original input (**Figure** 2). In this way, the network cannot learn the identity, unlike in basic autoencoder training (Vincent et al., 2008). Recent work has focused on using DAEs to recover clean signals from only noisy signals. The vast majority of this work lies in image analysis and builds off of "noise2noise" (Lehtinen et al., 2018), an approach that uses multiple noisy instances of the same image to learn to denoise the image. The approach relies on the fact that if the noise is zero mean, using a secondary noisy instance (besides the input image) as a target will produce a network that learns the clean image, in expectation, when enough training data are available. When paired samples are unavailable, other approaches exploit patches sampled from the image (Laine et al., 2019; Xie et al., 2020; Batson and Royer, 2019), but these do not apply to our setting since the target data we aim to correct are univariate. Other approaches eschew relying on inter-variable relationships but rely heavily on a known noise function (Moran et al., 2019; Kim and Ye, 2021) or a low expectation and variance noise function (Xu et al., 2020). Our approach builds off Xu et al. (2020), learning to reconstruct a signal from only potentially noisy samples of that signal, but in contrast to Moran et al. (2019) or Kim and Ye (2021), we do not make strong assumptions about the noise distribution. Instead, we leverage an auxiliary signal and iteratively filter out noisy samples.

Our approach is, in part, related to work in noisy-label learning, where a common approach involves identifying and reweighting samples with clean labels during training. Samples are filtered based on gradient values (Ren et al., 2020), Jacobian ranking (Mirzasoleiman et al., 2020) or some latent state (Lee et al., 2019; Wu et al., 2020). Co-teaching (Han et al., 2018), which builds off of mentor net (Jiang et al., 2018), filters out incorrectly labeled samples by utilizing two networks in parallel. For each network, backpropagation is performed using only samples within the current mini-batch for which the loss of the *other* network is lowest. Intuitively, samples with incorrect labels are likely to have higher loss and therefore be removed. Using two networks in parallel provides robustness to outliers and initially misclassified samples, to which single-network boosting-style approaches are sensitive. To date, these approaches have been primarily explored in supervised and semi-supervised settings. In contrast, we consider an unsupervised setting in which ground truth labels are unavailable and, instead, the input signals themselves are corrupted. To the best of our knowledge, such a co-teaching approach has not been explored in the context of denoising.

3. Problem Setup

Problem Overview. Given a noisy *target* variable and a reliably measured *auxiliary* time-series, we aim to recover the true values of the target variable. We assume there exists a relationship between the auxiliary and target variables, and that some samples from the noise distribution associated with the target variable will be close to zero, although which samples is unknown in advance.

Formalization. Given dataset $\mathcal{D}$ with k samples $\mathcal{D} = \{y_i, \mathbf{b}_i\}_{i=1}^{k}$, where $y_i \in \mathcal{R}$ denotes a noisy target



Algorithm Noise$^+$2Noise: Co-taught Denoising Auto-Encoders

Input DAE^1 and DAE^2 with randomly initialized weights, Loss function L, noise distribution M, fixed τ, epoch E_k, epoch E_{max}

for $t = 1, 2, \ldots, E_{max}$ **do**

Fetch batch $\{y_i\}$ and $\{\mathbf{b}_i\}$

(1) $\{\tilde{y}_i^1\} = DAE^1(\{\mathbf{b}_i\}, \{y_i\})$ \\Approximate clean values
$\{\tilde{y}_i^2\} = DAE^2(\{\mathbf{b}_i\}, \{y_i\})$

(2) $J^1 = \{j : L(y_j, \tilde{y}_j^1) < P_{R(t)}(L(\{y_j\}, \{\tilde{y}_j^1\}))\}$ \\Select samples
$J^2 = \{j : L(y_j, \tilde{y}_j^2) < P_{R(t)}(L(\{y_j\}, \{\tilde{y}_j^2\}))\}$

(3) **Sample** $\{m_i\} \sim M$
$\{z_i\} = \{y_i + m_i\}$ \\Doubly noisy values

(4) $\{\hat{y}_j^1\} = DAE^1(\{\mathbf{b}_j\}, \{z_j\})$ for $j \in J^2$ \\Reconstruct noisy signal
$\{\hat{y}_j^2\} = DAE^2(\{\mathbf{b}_j\}, \{z_j\})$ for $j \in J^1$

(5) **BACKPROP** $L(y_j, \hat{y}_j^1)$ for $j \in J^2$ and $L(y_j, \hat{y}_j^2)$ for $j \in J^1$

Update $R(t) = 1 - min\{\frac{t}{E_k}\tau, \tau\}$

Output DAE^1 and DAE^2

Figure 3: 'Noise$^+$2Noise'. Sample selection is performed with each DAE's output when given the uncorrupted y signal, but backpropagation is performed on the model's output when given a corrupted y signal. The loss values of DAE^1 are used to select the sample for backpropagation for DAE^2 and vice versa. $P_{R(t)}$ denotes the $R(t)^{th}$ percentile, where $R(t)$ is a function of iteration t. $\mathbf{b}$ signals are aligned so that non-zero y values always occur exclusively at the first position in the input window. The value of y is passed through to the decoder at each timepoint in a separate channel from $\mathbf{b}$.

sample and $\mathbf{b}_i \in \mathcal{R}^T$ denotes an auxiliary time-series. True target values, $\{x_i\}_{i=1}^k$, are unknown. For each sample, $y_i = x_i + n_i$, where n_i denotes a random variable drawn from some unknown distribution, i.e. $n_i \sim \mathcal{N}$. We aim to learn some mapping: $f : (y, \mathbf{b}) \rightarrow x$, given $\mathcal{D}$.

Assumptions. We assume that the distribution $\mathcal{N}$ is independent of both $\mathbf{b}$ and y. We assume that $\mathbf{b}$ is related to x, such that some approximate mapping $\mathbf{b} \rightarrow x$ exists. Implicitly, we assume that the timing of the target variable, relative to $\mathbf{b}$, is fixed and that the ordering is such that a mapping is possible (i.e., if x causes a change in $\mathbf{b}$, then $\mathbf{b}$ values must follow y in time so x can be learned retrospectively). We also assume that some values of n are near zero such that within training data $\mathcal{D}$, there exists a subset S with sufficient size for training such that the mean and variance of $n_s \forall s \in S$ are negligible compared to the mean and variance of $x_s \forall s \in S$. We assume that the distribution of low noise samples is such that they cover all regions of the input; *i.e.* that S does not exclude entire regions in the range of possible y values. Finally, we assume that the relationship between $\mathbf{b}$ and x can be accurately captured with a recurrent neural network (RNN).

While we note that these assumptions allow for noise of arbitrary average magnitude, access to some low-noise (though unlabeled) samples is assumed. Similar assumptions are common in healthcare applications (Chang et al., 2020; Geng et al., 2022; Zhang et al., 2021), where access to a small number of low noise or ground truth samples (through data curation) is possible but labor intensive and costly (e.g., prospective data collection or clinician review).

4. Methods

Overview. Our method, 'Noise$^+$2Noise' (**N$^+$2N**), is summarized in **Figure 3**. At a high level, we filter out noisy samples during training, refining the model parameters on selected samples of y estimated to have the least noise. To identify these higher fidelity samples within a batch, we identify the subset of samples with loss values below the $R(t)^{th}$ percentile of the batch, where $R(t)$ is an increasing function of iteration t. These 'low-noise' samples are augmented with additional noise and, along with corresponding

b vectors, are input to the DAE, which outputs a reconstruction $\hat{y}$. We then backpropagate using the squared error between $\hat{y}$ and y. This corresponds to training on the samples estimated to have the least noise, while utilizing the noisy signal as input. To increase robustness, we consider an ensemble approach in which we use co-teaching to train two DAEs (DAE^1 and DAE^2). During training, the samples identified as low-loss by DAE^1 are augmented and passed to DAE^2 for backpropagation, and vice-versa. The auxiliary signal **b** is input to the model during both the sample selection and backpropagation steps, allowing our approach to function even when the variable of interest y is too noisy or low-dimensional to be denoised alone.

Denoising Autoencoder. In our setup, additional noise (m) is added to y to produce z, a 'doubly' noisy measurement of x. z and **b** are input to a network (henceforth denoted DAE) that outputs $\hat{y} = DAE(z, \mathbf{b})$, and the network is trained to reconstruct y: loss is measured between y and $\hat{y}$. As shown by Xu et al. (2020), when the expectation and variance of the noise distribution N are negligible compared to those of the signal, the model parameters that minimize the loss between $\hat{y}$ and y are very close to the optimal parameters of a model trained on data that is identified as clean prior to training. Thus, provided the signal to noise level is high enough, we can pass y to DAE and expect a reduction in noise, with an output much closer to x, at inference time.

Co-teaching DAEs. We do not expect the noise value to be below a certain threshold at all times, but we do assume that some of the samples will have noise close to zero. We identify and train using these samples via an adapted co-teaching approach (Han et al., 2018). We utilize two DAEs, and for each, we backpropagate using only the samples for which the denoised y values from the other DAE are near the original y values. If the denoised y values approach x, we are then selecting samples for which the estimated noise n is lowest.

Claim: When using co-teaching to train two DAEs (DAE^1 and DAE^2) in parallel, denoised y values $\tilde{y}^1 = DAE^1(y, \mathbf{b})$ and $\tilde{y}^2 = DAE^2(y, \mathbf{b})$ approach x. *Justification*: Based on the main result of Xu et al. (2020), if the two DAEs are trained in a standard fashion, $\tilde{y}^1$ and $\tilde{y}^2$ converge to approximately x if, in the training data, the expectation and variance of the signal are much greater than those of the noise. We have assumed that such a sub-sample exists in our dataset, and propose that co-teaching is likely to select such a sub-sample. Note that each model is trained on data where additional noise is added to y and is tasked with reconstructing y, so they will not perform well if they output an identity function. Because the noise n is independent of both y and **b**, only the true component of the signal can be learned during training, so the models will learn some function of x. The intuition motivating Han et al. (2018) is that, in a noisy-label-learning setting, small-loss instances occur either when both the model and label are correct, OR when the model has memorized an incorrect label. By gradually decreasing the sample size of the training data based on loss values at each iteration, noisy label samples are dropped before the model can memorize them. In a similar vein, in our setting, a model will output a cleaned value $\tilde{y}$ that is close to y when either it has learned the correct function x and y is near to x, or if the model has memorized part of the noise. By employing the co-teaching sample selection method, we believe that the model selects the clean samples before it can memorize the noisy ones. We note that it is not impossible for the model to learn a biased function of x, but in practice, we have found that this approach works well even when there is fairly substantial bias in the noise.

Sample Selection. We select the samples with the lowest estimated noise for backpropagation. If $\tilde{y}^1$ and $\tilde{y}^2$ approach each network's estimated value of x, then DAE^1's estimate of n, the noise between x and y, is approximately $y - \tilde{y}^1$ (and similar for DAE^2). For loss function L, we use $L(\tilde{y}^1, y)$ and $L(\tilde{y}^2, y)$ to select samples. Given a batch $\{y_i\}_{i=1}^n$, at iteration t, we identify the subset of samples with values of $L(\tilde{y}_i^1, y_i)$ below the $R(t)^{th}$ percentile as $J^1 = \{j : L(y_j, \tilde{y}_j^1) < P_{R(t)}(L(\{y_j\}, \{\tilde{y}_j^1\}))\}$, where $P_{R(t)}$ denotes the $R(t)^{th}$ percentile, and similarly define J^2 for DAE^2. As in Han et al. (2018), we begin by training on the full sample. Over the course of training, as the DAEs are expected to become more accurate, we gradually reduce the sample. This prevents the memorization of noisy samples that can occur later in training. Hyperparameter $\tau \in (0, 1)$ in the pseudocode of **Figure 3** represents the maximum proportion of samples removed and E_k represents the iteration at which we stop increasing the proportion of samples removed. A linear decrease in sample size as a function of iteration t is implemented by using the lowest-loss $R(t) = (1 - Maximum(\frac{t}{E_K}\tau, \tau)) \cdot 100\%$ of samples for backpropagation.

Training. Each DAE is trained on the samples for which the other network estimates that the noise is

lowest: samples selected by DAE^1 (y_j where $j \in J^1$) are augmented with additional noise $m_j \sim M$ to generate z_j values. z_j, along with corresponding $\mathbf{b}_j$ vectors, are input to DAE^2, which outputs a reconstruction of y_j: $\hat{y}_j^2$. We then backpropagate using the squared error between $\hat{y}_j^2$ and y_j. Similarly, we only use samples y_j where $j \in J^2$, augmented with $m_j \sim M$, to backpropagate DAE^1. By selecting samples based on $L(\tilde{y}, y)$ rather than based on $L(\hat{y}, y)$, we are able to select a sample independent of secondary noise value m. Selecting samples dependent on m would be confounding because samples might then be selected based on how low the value of m is at the current iteration, rather than the value of n, which is hidden. Back-propagation is performed on an input that does not include unaltered y values, so the model is not likely to learn the identity function. Sample selection is always performed by the other DAE, so compared to boosting or other one-network approaches, our method is less sensitive to error propagation from wrongly selected samples early in training.

Co-teaching+. We utilize co-teaching+ (Yu et al., 2019), where samples for which the models disagree are selected for backpropagation. As a result, each model learns from the samples for which the other model's estimates were better. This prevents the models from learning from the samples that they agree upon, which prevents convergence (Yu et al., 2019), maintaining unique strengths in each model. We remove the $\sigma\%$ of samples for which the models' outputs are closest (σ is a hyperparameter). This step is performed prior to the sample selection step: the $\sigma\%$ of samples for which the distance between $\hat{y}^1$ and $\hat{y}^2$ are lowest are removed, and then the remaining samples for which $L(\tilde{y}^1, y)$ is lowest are used for the backpropagation of DAE^2 and vice versa.

5. Real-world Problem Setup: Blood Glucose Management

To explore the benefit of our proposed approach, we consider a real-world problem setup based on blood glucose management that inspired the setting described in Section 3. Nearly two million people in the US have type I diabetes and require insulin to maintain healthy glucose levels. They must deliver boluses of insulin through an injection or an insulin pump prior to eating to counteract the rise in blood sugar that results from meals. Bolus amounts are calculated based on patient-reported estimates of carbohydrates. Carbohydrates and bolus insulin generally cause blood glucose values to increase or decrease after a delay of 30 minutes to an hour. In our setup, carbohydrates correspond to x values, glucose levels and insulin values correspond to $\mathbf{b}$ values.

Blood glucose forecasting and control have been extensively studied (Silvia Oviedo, 2016; Fox et al., 2020). Accurate models for blood glucose dynamics are critical to the development of algorithms for managing blood glucose in individuals with diabetes both in terms of patient-selected treatment options and automated solutions. However, carbohydrates consumed are patient-reported and as a result are often inaccurate Brazeau et al. (2012); Mehta et al. (2009). This in turn leads to inappropriate doses of insulin and poor blood glucose management. Besides misestimation, there are other sources of inconsistency between recorded carbohydrate values and their effects on blood glucose. Variability in meal types is generally poorly captured, which is problematic because the effect of carbohydrates on blood glucose can be moderated by how quickly a meal was consumed, or the amount of protein, fat and other nutrients ingested. Additionally, the timing of a meal may not be recorded accurately. These factors alone make utilizing carbohydrate information difficult, even when carbohydrates are accurately recorded. In an unsupervised setting, denoising approaches could learn representations of carbohydrate values that incorporate these other sources of variability. These representations could be more relevant to blood glucose management than the exact number of grams consumed. This could improve performance of forecast and control algorithms.

6. Experimental Setup

We evaluate our approach in the context of learning to correct noisy patient-reported carbohydrate measurements. We compare performance to several baselines across real and simulated datasets.

6.1. Datasets

We utilize two type I diabetes-based datasets. The simulated dataset provides access to ground truth to which we can directly compare our method's denoised outputs. The real dataset provides a more challenging setting for quantifying the efficacy of our approach, but corresponds to real-world scenar-

ios. Both datasets are publicly available and have been previously explored in the context of forecasting and control (Man et al., 2014; Xie, 2018; Marling and Bunescu, 2018, 2020). Both datasets consist of blood glucose, bolus (fast-acting) insulin, basal (slow-acting) insulin, and carbohydrate values. All variables were scaled to be between zero and one. For both datasets, time-series trajectories for each patient were split into windows of 2 hour length ($T = 24$ 5-minute time points). We ignore windows where a carbohydrate occurs in anywhere but the first position, using only windows with no carbohydrates or carbohydrates at the beginning of the window during training. This means we also ignore windows with more than one carbohydrate present. In a real-world setting these values could be updated recursively, but we simplify our setting here. The auxiliary signal is assumed to be cleaner than the highly noisy target variable, but not completely noise-free. In practice, the auxiliary signal can have noise- there is approximately 5% noise in both the simulated and real blood glucose monitor data used in experiments.

Simulated. Our primary analyses are performed on data generated using the UVA-Padova simulator (Man et al., 2014) via a publicly available implementation (Xie, 2018). For ten simulated individuals (the "adult" patients modeled in the simulator), we generated approximately 150 days worth of data each, in 30 day roll-outs of the simulator. Carbohydrate values serve as x values, while CGM values and insulin delivered as output by the simulator serve as $\mathbf{b}$ values. We use noise proportional to the true carbohydrate value, as studies on the accuracy of carb counting report errors relative to the total carbs consumed (Brazeau et al., 2012). Also based on Brazeau et al. (2012), we use a noise distribution with a negative bias, as the carbs were found to be more-often under-reported than not. We therefore set $y = (1 + \mathcal{N}(-.25, .5))x$. We then cap y below and above at 1 and 200 to keep values realistic. We consider additional noise distributions as sensitivity analyses. Bolus values were calculated based on the noisy carbohydrate values. See **Appendix A** for more details on data generation.

Real. 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, 2018, 2020). The data pertain to 12 individuals, each with approximately 10,000 5-minute samples for training and 2,500 for testing. 12% of glucose values are missing, but we do not include windows with missing glucose values. We do not include windows with more than one carbohydrate measurement in our analysis. We sum carbohydrates to the first timepoint if they are less than 15 minutes apart to maximize the amount of usable data. We include only individuals with at least 100 training carbohydrate measurements, as fewer are not sufficient for learning a model. We note that ground truth carbohydrate measurements are not available for this dataset. Only potentially noisy patient-estimated values are reported.

6.2. Baselines and Upper Bound

For all non-coteaching methods, we train two DAEs in parallel and report results on their averaged output for a fair comparison. We also note that all models receive the same auxiliary variables (blood glucose/ insulin) as input in an identical fashion. Overall time complexity is similar for all methods because there is only one back propagation per sample per-DAE.

- **CAE**: An upper performance bound. This model is an autoencoder trained with ground truth data, which we would expect to perform better than any method without access to ground truth data. In this oracle approach, x values are substituted for y values during training, but y values are used during testing.
- **NAC**: Our first baseline. A DAE that treats the noisy data as clean which has been shown to perform well in low noise settings (Xu et al., 2020).
- **NR2N**: Our second baseline is noisier2noise (Moran et al., 2019), which uses the known noise distribution to recover the clean signal. **NR2N** trains similarly to **NAC**, but at evaluation time a transform is used to recover the clean values (briefly, if the distribution of N is known and we set $M = N$, the model should learn to recover half of the noise so the value used at evaluation is $2\hat{y} - z$).
- **SUP**: Our motivating setting can be re-framed as a supervised learning problem: predict y (or x) values using $\mathbf{b}$ values as input. Depending on the noise distribution, it is possible that a model trained on noisy y values could learn to predict the correct x, using similar logic to that found in Lehtinen et al. (2018). We therefore use this supervised setting as a naive baseline. We simply input $\mathbf{b}$ to the same network used in the DAE setting and calculate loss as $(\hat{y}-y)^2$ during training, but here the model has no information regarding y or z. As in the DAE setting, at test time we evaluate $(\tilde{y} - x)^2$.
- **SUPCT**: We apply co-teaching to the supervised setting (**SUP**), to ensure that performance gains ob-

served are due to the combination of DAEs and co-teaching, and not co-teaching alone. Here, the model is tuned and trained identically to **N$^+$2N**, except the model does not receive y or z values.

6.3. Implementation & Training Details

Each DAE is implemented as a 2-layer bidirectional LSTM with 100 hidden units. The LSTM model was chosen because it has been shown to perform well in blood glucose forecasting, which is a closely related challenge (Rubin-Falcone et al., 2020; Mirshekarian et al., 2019; Rabby et al., 2021).The final hidden state is passed to a fully connected layer with a single output. The output of the model is added to the input value corresponding to y, so that the network is tasked with learning the error term rather than a complete reconstruction. Because we only aim to correct a single carbohydrate (y) value but use a time-based model, we set one dimension of the model input to be y for all timepoints. Multiple auxiliary **b** signals are input to the LSTM through separate channels: blood glucose values and bolus and basal insulin are included in this way. We also carry over bolus insulin values (which occur sparsely) to the end of the input window, to increase their impact on gradient calculations. We threshold the output of each DAE at 0, because carbohydrates (our x and y) values cannot be negative. For each co-teaching method and **NR2N**, hyperparameters were selected based on tuning to a single individual adult#001. A small number of options was considered through a simple grid search. Once selected, these hyperparameters were used across all datasets. In tuning on adult#001, a ground truth signal was used for validation. This is a limitation, as such a signal is generally not available in real-world scenarios. However, the fact that the hyperparameters were not tuned to each individual, highlights the robustness of the approach. Tuning is described in **Appendix B**. At evaluation we report the result of the average correction learned by both networks when y values are given as input (*i.e.*, where $\tilde{y}_i = DAE_i(y, \mathbf{b})$, we report $L(x, (\tilde{y}^1 + \tilde{y}^2)/2)$).

For sample selection during co-teaching, we use mean squared percentage error ($100\% \cdot ((\hat{y} - y)/y)^2$) to avoid eliminating all high-valued y samples, as they are likely to have higher noise values. As a noise function during training, we use $z = (1 + \mathcal{N}(0, .5))Bern(.5)y$, *i.e.* we add random noise to half of the samples so that the model can learn to utilize noisy z information, and zero-out the other half so that the model has to learn to distinguish zero from non-zero y values based on **b** alone. See **Appendix C** for additional training details.

6.4. Evaluation

Simulated Data Metrics. When ground truth carbohydrate values are available, we use MSE between the denoised carbohydrates and true carbohydrate values as our metric to report the remaining noise ($mean((x - \tilde{y})^2)$, where $\tilde{y} = (\tilde{y}^1 + \tilde{y}^2)/2$). The lower this value, the more noise has been removed. Although we assume that these data are not available at training time, we use them for evaluation. Since it can be difficult to interpret the meaning of a difference in MSE, we also consider a clinically motivated evaluation metric: *time in range*. Time in range is a measure of blood glucose management and varies with the accuracy of the carbohydrate measurements. The more accurate the carbohydrate estimates the more time an individual will spend 'in range.' Here, we run a simulation of the subject of interest with the default basal-bolus controller using bolus values calculated from the updated carbohydrate values, and report the proportion of time in the simulation that each individual spent with blood glucose values between 70 and 180, or the euglycemic/healthy range. This metric serves to indicate the real-world impact each approach might have. For both metrics, 95% confidence intervals are calculated for each subject using 1,000 bootstrap re-samples, and the average 2.5^{th} and 97.5^{th} percentiles across subjects are reported.

Sensitivity Analyses. To evaluate our model under different noise assumptions, we repeat our analysis with multiple noise generation methods ($x \rightarrow y$), without altering hyperparameters or our $y \rightarrow z$ function. We use various Gaussian and uniform distributions reported in **Appendix D**, which include zero and negative mean multiplicative and additive noise functions. We do not aim at a comprehensive evaluation of all possible noise types, but rather we aim to include various distributions that are likely similar to those that might arise in our motivating domain. To further test the ability of the approach to recover the target values, we varied the amount of noise in the auxiliary signal from 5% to 25% magnitude by adding a multiplier to the noise component generated by the simulated CGM, and evaluated each approach as this **b** signal became increasingly noisy (**Appendix E**). We also examine our model's sensitivity to including

Table 1: Our approach outperforms baselines for all evaluation metrics and both datasets, falling only 2% short of the upper bound for the clinical measurement 'Time in Range.' 95% CIs are calculated from 1,000 bootstrap re-samples. CRC r and p values are calculated from the Spearman correlation.

	SIMULATED			REAL
Model	Remaining Carb MSE(g^2)[95% CI]	Time in Range (%)[95% CI]	CRC (r)[p]	CRC (r)[p]
N/A-clean carb	0.00 [0.00, 0.00]	73.18 [72.49,73.88]	N/A	N/A
N/A-noisy carb	72.26 [54.16, 92.58]	65.43 [64.70,66.16]	N/A	N/A
CAE (Oracle)	6.96 [5.03, 9.18]	72.44 [71.76,73.11]	0.32 [< 0.001]	N/A
SUP	58.37 [45.11, 73.31]	64.59 [63.89,65.30]	0.04 [0.14]	0.19 [< 0.001]
SUPCT	100.50 [84.12,118.40]	60.22 [59.43,60.94]	< 0.001 [0.91]	0.03 [0.61]
NAC	36.40 [27.02, 46.95]	68.22 [67.53,68.93]	0.11 [< 0.001]	0.13 [0.05]
NR2N	33.44 [26.59, 43.23]	68.79 [68.09,69.51]	0.11 [< 0.001]	0.13 [0.05]
N^+2N (Ours)	**17.91 [12.61,24.98]**	**71.24 [70.54,71.90]**	**0.19** [$<$ **0.001**]	**0.22** [$<$ **0.001**]

varying amounts of noisy data in the final training sample by varying hyperparameter τ, which controls the proportion of samples within each minibatch used in backpropagation (**Appendix F**).

Real Data Analysis. Without access to ground truth carbohydrate values at test time for the real dataset (unlike the simulated dataset), we evaluate the performance of our denoising approach based on a proxy. We take advantage of the fact that poorly estimated carbohydrates result in inappropriate bolus calculations, which result in poor blood glucose management. We expect that inaccurate carbohydrates estimates (large $[x-y]$ values) result in the poor blood glucose management. For a model that has come close to estimating x correctly, we would observe a correlation between $[\tilde{y}-y]$ values and blood glucose control in the time period following a meal. We assess this with Correction-Risk-Correlation (CRC), defined as the Spearman correlation between the squared carbohydrate correction value ($(\tilde{y}-y)^2$) and the average Magni Risk (Magni et al., 2007) of blood glucose in the second hour following the carbohydrate. We use the Spearman correlation to account for nonlinearities in the risk and correction value distributions. Magni risk is a measure of how far from a safe value blood glucose is; higher risk values correspond to blood glucose values that are either dangerously high or dangerously low. We use the second hour following the carbohydrate because the effects of the carbohydrate consumption and insulin bolus have not fully taken effect in the first hour. We calculate this correlation across all carbohydrates observed in all individuals. For validation purposes, we also calculate this metric for the simulated dataset.

7. Results and Discussion

Through our experiments, we aim to answer the following questions.

- Does our approach meaningfully reduce error across a variety of simulated individuals, compared to existing approaches?
- Is our model robust to different domain-appropriate noise distributions?
- Does our model show strong performance in real data, indicating accurate denoising?

Error Reduction for Simulated Data. Our approach, $\mathbf{N^+2N}$, outperforms baselines in terms of noise reduction (MSE) (**Table** 1). $\mathbf{N^+2N}$ reduces MSE from $72g^2$ to $18g^2$, but falls short of the value achieved by our oracle approach **CAE** ($7g^2$), as expected. **CAE** does not achieve perfect MSE, probably due to insufficient training data or to a small amount of noise in the CGM signal. Our approach's reduction in noise is meaningful since it leads to significantly better time in range. Most methods offer an improvement in % time in range when used in a basal bolus controller, with baselines increasing over the noisy value from 65% to 69%, and $\mathbf{N^+2N}$ further improving performance to 71%, recovering 6% time in range out of a total of 8% lost when using noisy versus clean values. **SUPCT** performs worse than any other method including **SUP**, likely because without the noisy carbohydrate measurement as input, co-teaching cannot learn the relationship between $\mathbf{b}$ and y as easily, and therefore does not identify the less corrupted samples during training. This results

in essentially random sub-sample selection, hampering performance as less training data becomes available. **SUP** does not suffer from this problem because it always utilizes the entire dataset.

Sensitivity Analyses. **N$^+$2N** outperforms all baselines across the majority of noise distributions (**Figure** 4). For zero-mean uniform multiplicative noise, **NAC** outperforms the proposed approach. We hypothesize that **NAC** performs well in this setting because the expected value of the noise is zero and the variance is lower than in other settings (it is 33%, which is approximately 30g, compared to 75% in the multiplicative normal setting, or 40g and 60g in the additive noise settings, see **Appendix D**). Of note, this analysis was carried out without additional tuning, demonstrating the resilience of our approach to varying noise assumptions. Across biased noised distributions, our proposed approach consistently outperforms *all* baselines. This resilience is likely due to our approach's ability to select clean samples for training, without reliance on the secondary noise function used during training.

We found that our approach is fairly robust to additional noise in the auxiliary signal: when we increased the amount of noise in **b** by 5X, our approach remained competitive with baselines trained with a clean **b** signal. At all noise settings our approach strongly outperformed baselines trained with similar data. Details and figures are in **Appendix E**.

In our examination of final training sample size, we found that performance is relatively stable with a larger, noisier, sample, with all τ values between zero and 0.5 (half of the samples excluded) offering substantial improvement over the best performing baseline. Performance degrades for larger values of τ, which is likely due to the limited number of training samples. See **Appendix F** for details and a plot of model performance as τ is varied.

Experiments on Real Data. For the real dataset, **N$^+$2N** outperforms all baselines with respect to CRC. For simulated data, we see that, without exception, models with lower remaining MSE after denoising have a higher or equal CRC. This indicates that our metric serves as a reasonable proxy for remaining error when true values are unavailable. Plots showing the components used to calculate CRC (magnitude of carbohydrate correction versus Magni risk an hour after the meal) can be found in **Appendix G**. Interestingly, the **SUP** baseline performs fairly well for this task on the real dataset (r,p=0.19, 3e-3, vs. proposed approach 0.22, 9e-4). We hypothesize that this may be because carbohydrate measurements are so unreliable for this dataset that learning to predict them from scratch (without access to noisy values at test time) is sufficient for an error estimation proportional to the actual error, especially given the implicitly correct timing data.

8. Conclusions

We propose a new approach to denoising, 'Noise$^+$2Noise', that does not assume access to ground truth target samples. Our approach leverages an auxiliary time-series that is related to the target signal to help identify target samples with less noise. Our approach is the first to adapt co-teaching to de-noising, extending the applicability of this method to many potential settings. While the approach recovers target data retrospectively and cannot be used in real time for forecasting, it could be used in a number of downstream tasks. For example in the clinical context, errors in carbohydrate measurements could aid in evaluating an individual's efforts in blood glucose management and provide a potential target. In the context of carbohydrate recovery for blood glucose management, compared to existing approaches, 'Noise$^+$2Noise' leads to better signal reconstruction that is both statistically significant and clinically significant.

While promising, our approach is not without limitations. Our primary analyses are on simulated data where ground truth labels are available, but in real datasets common evaluation metrics (e.g., MSE) do not apply and we must rely on proxies. As presented, our approach is designed for retrospective carbohydrate correction; more work is necessary to investigate its applicability to closer-to-real-time correction. Four individuals in the Real dataset had too few carbohydrate measurements to reliably train a de-noising model, which means that further work on model efficiency is necessary for this model to be broadly applicable. While our approach was designed for and evaluated on denoising non-missing measurements, our method could be extended to address missing measurements as well, provided some additional regularization is utilized to ensure that the model does not impute meal announcements too excessively. Finally, while we have empirically shown that co-teaching appears to select a low-noise sample, we have not provided statistical guarantees.

Despite these limitations, we have demonstrated that it is feasible to correct a noisy variable with-

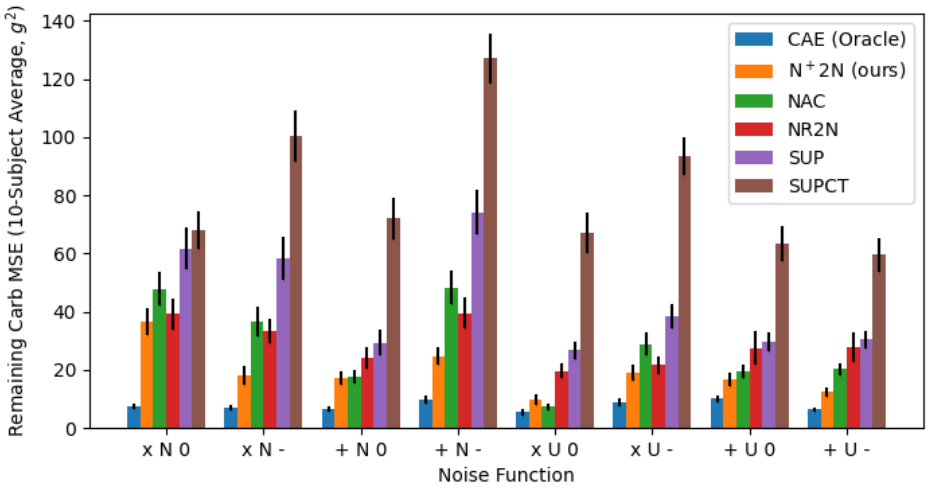


Figure 4: Performance on datasets with multiplicative ($\times$) vs. additive ($+$), Normal (N) vs. Uniform (U), and zero (0) vs. negative mean ($-$) noise functions. $\mathbf{N^+2N}$ generally outperforms baselines. Error bars represent standard error (68% confidence interval) from 1000 bootstrap samples.

out access to ground truth samples during training, expanding the utility of ideas from image analysis and noisy label learning. Applied to domains in which data are composed of both individual-reported data and data measured from reliable sensors (e.g., mHealth), our approach could aid in improving the fidelity of patient-reported data.

Acknowledgements

This work was supported by JDRF (award no. 5-COE-2019-861-S-B). The views and conclusions in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied of JDRF.

References

Joshua Batson and Loic Royer. Noise2self: Blind denoising by self-supervision. *ICML*, 2019.

Anne-Sophie Brazeau, H Mircescu, Katherine Desjardins, C Leroux, I Strychar, J.M. Ekoé, and R Rabasa-Lhoret. Carbohydrate counting accuracy and blood glucose variability in adults with type 1 diabetes. *Diabetes research and clinical practice*, 99, 2012.

Claudia Bull, Joshua Byrnes, Ruvini Hettiarachchi, and Martin Downes. A systematic review of the validity and reliability of patient-reported experience measures. *Health Services Research*, 2019.

Yi Chang, Luxin Yan, Meiya Chen, Houzhang Fang, and Sheng Zhong. Two-stage convolutional neural network for medical noise removal via image decomposition. *IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT*, 2020.

Kate Churruca, Chiara Pomare, Louise A. Ellis, Janet C. Long, Suzanna B. Henderson, Lisa E. D. Murphy, Christopher J. Leahy, and Jeffrey Braithwaite PhD. Patient-reported outcome measures (proms): A review of generic and condition-specific measures and a discussion of trends and issues. *Health Expectations*, 2021.

Emanuele D'Amico, Rocco Haase, and Tjalf Ziemssen. Review: Patient-reported outcomes in multiple sclerosis care. *Multiple Sclerosis and Related Disorders*, 2019.

Ian Fox, Joyce Lee, Rodia Pop-Busui, and Jenna Wiens. Deep reinforcement learning for closed-loop blood glucose control. *Proceedings of Machine Learning Research*, 2020.

Mufeng Geng, Xiangxi Meng, Jiangyuan Yu, Lei Zhu, Lujia Jin, Zhe Jiang, Bin Qiu, Hui Li, Hanjing Kong, Kun Yang, Hongming Shan, Hongbin Han, Zhi Yang, Qiushi Ren, and Yanye Lu. Content-noise complementary learning for medical image denoising. *IEEE Transactions on Medical Imaging*, 2022.

Lovedeep Gondara. Medical image denoising using convolutional denoising autoencoders. *IEEE 16th International Conference on Data Mining Workshops*, 2016.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *NeurIPS*, 2018.

James Arthur Harris and Francis Gano Benedict. A biometric study of basal metabolism in man. *Carnegie institution of Washington*, 1919.

Jeremiah Hauth, Safa Jabri, Fahad Kamran, Eyoel W. Feleke, Kaleab Nigusie, Lauro V. Ojeda, Shirley Handelzalts, Linda Nyquist, Neil B. Alexander, Xun Huan, Jenna Wiens, and Kathleen H. Sienko. Automated loss-of-balance event identification in older adults at risk of falls during real-world walking using wearable inertial measurement units. *Sensors*, 2021.

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. *ICML*, 2018.

Kwanyoung Kim and Jong Chul Ye. Noise2score: Tweedie's approach to self-supervised image denoising without clean images. *NeurIPS*, 2021.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference for Learning Representations*, 2014.

Brent M. Kious, Amanda Bakian, Joan Zhao, Brian Mickey, Constance Guille, Perry Renshaw, and Srijan Sen. Altitude and risk of depression and anxiety: findings from the intern health study. *International Review of Psychiatry*, 2019.

Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. Noise2void - learning denoising from single noisy images. *CVPR*, 2018.

Samuli Laine, Tero Karras, Jaakko Lehtinen, and Timo Aila. High-quality self-supervised deep image denoising. *NeurIPS*, 2019.

Kimin Lee, Sukmin Yun, Kibok Lee, Honglak Lee, Bo Li, and Jinwoo Shin. Robust inference via generative classifiers for handling noisy labels. *ICML*, 2019.

Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2noise: Learning image restoration without clean data. *ICML*, 2018.

You Lin, Juanhui Wang, and Mingjian Cui. Reconstruction of power system measurements based on enhanced denoising autoencoder. *IEEE General Meeting Power Energy Society*, 2019.

Lalo Magni, Davide M. Raimondo, Luca Bossi, Chiara Dalla Man, Giuseppe De Nicolao, Boris Kovatchev, and Claudio Cobelli. Model predictive control of type 1 diabetes: An in silico trial. *Journal of diabetes science and technology*, 1, 2007.

Chiara Dalla Man, Francesco Micheletto, Dayu Lv, Marc Breton Boris Kovatchev, and Claudio Cobelli. The uva/padova type 1 diabetes simulator. *J Diabetes Sci Technol.*, 8, 2014.

Ciny Marling and Razvan C. Bunescu. The OhioT1DM dataset for blood glucose level prediction. *International Workshop on Knowledge Discovery in Healthcare Data-KHD@IJCA*, 2018.

Ciny Marling and Razvan C. Bunescu. The OhioT1DM dataset for blood glucose level prediction: Update 2020. *International Workshop on Knowledge Discovery in Healthcare Data-KHD@IJCA*, 2020.

Roger S. McIntyre, Zahinoor Ismail, Christopher P. Watling, Catherine Weiss, Stine R. Meehan, Primrose Musingarimi, and Michael E. Thase. Patient-reported outcome measures for life engagement in mental health: a systematic review. *Journal of Patient-Reported Outcomes*, 2022.

Stephen P McKenna. Measuring patient-reported outcomes: moving beyond misplaced common sense to hard science. *BMC Medicine*, 2011.

S.N. Mehta, N. Quinn, L.K. Volkening, and L.M. Laffel. Impact of carbohydrate counting on glycemic control in children with type 1 diabetes. *Diabetes Care*, 32, 2009.

Sadegh Mirshekarian, Hui Shen, Razvan Bunescu, and Cindy Marling. Lstms and neural attention models for blood glucose prediction: Comparative experiments on real and synthetic data. *Annu Int Conf IEEE Eng Med Biol Soc.*, 2019.

Baharan Mirzasoleiman, Kaidi Cao, and Jure Leskovec. Coresets for robust training of deep neural networks against noisy labels. *NeurIPS*, 2020.

Nick Moran, Dan Schmidt, Yu Zhong, and Patrick Coady. Noisier2noise: Learning to denoise from unpaired noisy data. *CVPR*, 2019.

Hanh Nguyen, Phyllis Butow, Haryana Dhillon, and Puma Sundaresan. A review of the barriers to using patient-reported outcomes (pros) and patient-reported outcome measures (proms) in routine cancer care. *Journal of Medical Radiation Sciences*, 2020.

Giorgio Quer, Jennifer M. Radin, Matteo Gadaleta, Katie Baca-Motes, Lauren Ariniello, Edward Ramos, Vik Kheterpal, Eric J. Topol, and Steven R. Steinhubl. Wearable sensor data and self-reported symptoms for covid-19 detection. *Nature Medicine*, 2020.

Md Fazle Rabby, Yazhou Tu, Md Imran Hossen, Insup Lee, Anthony S. Maida, and Xiali Hei. Stacked lstm based deep recurrent neural network with kalman smoothing for blood glucose prediction. *BMC Medical Informatics and Decision Making*, 2021.

Zhongzheng Ren, Raymond Yeh, and Alexander Schwing. Not all unlabeled data are equal: Learning to weight data in semi-supervised learning. *NeurIPS*, 2020.

Harry Rubin-Falcone, Ian Fox, and Jenna Wiens. Deep residual time-series forecasting: Application to blood glucose prediction. *International Workshop on Knowledge Discovery in Healthcare Data-KHD@IJCA*, 2020.

Alexandra Samet. The top medical monitoring and healthcare wearable device trends of 2022. *Insider Intelligence*, 2022.

Viral N Shah, Lori M Laffel, R Paul Wadwa, and Satish K Garg. Performance of a factory-calibrated real-time continuous glucose monitoring system utilizing an automated sensor applicator. *Diabetes Technol Ther.*, 2018.

Remei Calm Joaquim Armengol Silvia Oviedo, Josep Vehí. A review of personalized blood glucose prediction strategies for t1dm patients. *Int J Numer Method Biomed Eng*, 2016.

Lamkin P. Smart. wearables market to double by 2022: $27 billion industry forecast. *Forbes*, 2018.

Esmee M. van der Willik, Caroline B. Terwee, Willem Jan W. Bos, Marc H. Hemmelder, Kitty J. Jager, Carmine Zoccali, Friedo W. Dekker, and Yvette Meuleman. Patient-reported outcome measures (proms): making sense of individual prom scores and changes in prom scores over time. *Nephrology*, 2020.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. *ICML*, 2008.

Priscilla Jia Ling Wee, Yu Heng Kwan, Dionne Hui Fang Loh, Jie Kie Phang, Troy H Puar, Truls Østbye, Julian Thumboo, Sungwon Yoon, and Lian Leng Low. Measurement properties of patient-reported outcome measures for diabetes: Systematic review. *Journal of Medical Internet Research*, 2021.

Pengxiang Wu, Songzhu Zheng, Mayank Goswami, Dimitris Metaxas, and Cho Chen. A topological filter for learning with label noise. *NeurIPS*, 2020.

Jinyu Xie. Simglucose v0.2.1 [online]. avaible: https://github.com/jxx123/simglucose. *Accessed on: Jan-20-2020*, 2018.

Yaochen Xie, Zhengyang Wang, and Shuiwang Ji. Noise2same: Optimizing a self-supervised bound for image denoising. *NeurIPS*, 2020.

Peng Xiong, Hongrui Wang, Ming Liu, Suiping Zhou, Zengguang Hou, and Xiuling Liu. Ecg signal enhancement based on improved denoising auto-encoder. *Engineering Applications of Artificial Intelligence*, 52, 2016.

Jun Xu, Yuan Huang, Ming-Ming Cheng, Li Liu, Fan Zhu, Zhou Xu, and Ling Shao. Noisy-as-clean: Learning self-supervised denoising from the corrupted image. *TIP*, 2020.

Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor W. Tsang, and Masashi Sugiyama. How does disagreement help generalization against label corruption? *ICML*, 2019.

Haoming Zhang, Mingqi Zhao, Chen Wei, Dante Mantini, Zherui Li, and Quanying Liu. Eegdenoisenet: a benchmark dataset for deep learning solutions of eeg denoising. *Journal of Neural Engineering*, 2021.

Appendix A. Simulated Dataset Details

During data generation, days where a patient either had more than 25 timepoints of glucose at the minimum value of 40, or more than 35 timepoints over 450 were thrown out for being non-realistic. 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). In our simulation, for all datasets generated, we used the default basal-bolus controller from the existing implementation of the simulator to administer insulin, but we delayed five sixths (randomly selected) of the bolus administrations up to 3.5 hours, with delay time randomly sampled from a uniform distribution. The delay allows for disentanglement between carbohydrate and bolus effects. 20% of carbohydrates are not reported, to make the dataset more realistic, as missingness is common.

Appendix B. Tuning Details

For each model, tuning was performed on simulated adult#001 using validation performance. No additional tuning was performed for other individuals or noise functions. For Noisier2Noise, we selected α, the parameter that controls the relative noise distributions, from [0.1,0.3,0.5,0.7,0.9,1.0,1.25,1.5,1.75,2], ultimately selecting $\alpha = 1$. Because we do not assume access to the exact noise we would not expect this method to perform spectacularly, but note that it often outperforms other baselines.

For co-teaching methods, we performed a simple grid search over the values of E_k=[250,500] (where 500 is the minimum number of training iterations), τ=[0.333,0.5,0.667], and σ = [0.1,0.3,0.5,0.7]. For $\mathbf{N^+2N}$, we selected $E_k = 250$, $\tau = 0.333$, and $\sigma = 0.1$. For the supervised setting co-teaching (**SUPCT**), we set selected $T_k = 500$, $\tau = 0.5$ and $\sigma = 0.3$.

Hyperparameter options were selected from a limited but comprehensive spectrum of values that cover a reasonable search space (given that all hyperparameters are limited to a fixed interval) without consideration for task. Only a small number of options were considered to avoid computational burden, as a simple grid search was used. A ground truth signal was used for evaluation during tuning, which is a limitation, as such a signal is generally not available in real-world scenarios. However, we note that we did not re-tune for each individual (tuning to simulated adult#001), nor did we retune for the real-world dataset, which is substantially different from the simulated dataset. Proxy measures such as CRC may also be used for tuning.

Appendix C. Additional Training Details

We split each dataset into training, validation and test sets used for evaluation purposes. For the simulated dataset, we use 80 days for training, 20 for validation, and 50 for testing. For the real dataset, we split the training data into 80% train and 20% validation. The held-out test data were used for evaluation only. We implement and train our models in Pytorch 1.9.1 with CUDA version 10.2, using Ubuntu 16.04.7, a GeForce RTX 2080, an Adam optimizer (Kingma and Ba, 2014), and a batch size of 500. We use a learning rate of 0.01 and a weight decay of 10^{-7}. We train for at least 500 iterations, and then until validation performance does not improve for 50 iterations, selecting the model for which validation performance was best. For both datasets, we train and test a model on each individual and report across-individual averages. Such individual-specific models/evaluations are common in blood glucose control and forecasting (Silvia Oviedo, 2016), since dynamics vary greatly across individuals and individual-specific training data are typically available.

We perform co-teaching on samples containing non-zero y values only. However, when training all models (including baselines) we also pass zero-valued y samples (and their corresponding $\mathbf{b}$ values) through both DAEs and take loss equal to $\hat{y}^2$ for these samples. We do this because there are many more samples with zero-valued carbohydrates than there are with positive values, and this allows the models to learn from this larger collection. We report results on only positive-valued y values, because denoising is only applied to such values.

Appendix D. Alternate Noise Functions

We consider noise functions that might arise in carbohydrate counting. None are highly dissimilar from our main analysis noise function: we aim here at feasibility, rather than a comprehensive survey on a broad selection of loss functions, which our method would

likely be unable to address without further tuning or modification. Here, $\mathcal{U}(a, b)$ denotes a uniform distribution with values between a and **b**. Carbohydrate values range between 0 and 200. After adding noise, y values are capped above and below by 1 and 200. Alternate noise functions include:

1. Zero-mean multiplicative Gaussian: $y = (1 + \mathcal{N}(0, .75))x$
2. Negative-mean multiplicative Gaussian (primary noise function): $y = (1 + \mathcal{N}(-.25, .5))x$
3. Zero-mean additive Gaussian: $y = x + \mathcal{N}(0, 40)$
4. Negative-mean additive Gaussian: $y = x + \mathcal{N}(-30, 50)$
5. Zero-mean multiplicative Uniform: $y = \mathcal{U}(.5, 1.5)x$
6. Negative-mean multiplicative Uniform: $y = \mathcal{U}(0, 1.6)x$
7. Zero-mean additive Uniform: $y = x + \mathcal{U}(-60, 60)$
8. Negative-mean additive Uniform: $y = x + \mathcal{U}(-60, 40)$

Appendix E. Sensitivity to noise in the b signal.

Although the auxiliary **b** signal is expected to be relatively low noise compared to y, some noise is possible. In our motivating domain, CGM data contain a non-negligible amount of noise. In the simulator, this noise is modeled as additive time-varying Gaussian (Man et al., 2014). To evaluate our approach's sensitivity to noise in the auxiliary signal, we added an increasing multiplier to the noise term in the CGM for each simulated individual. The multiplier ranged from 1X to 6X, with 1X being the standard CGM. At 6X noise the magnitude of the signal is more than 25% noise on average, and the original glucose signal is barely detectable (**Figure 5**).

We found that at each noise setting, our approach outperformed all baselines (**Figure 6**). Also encouragingly, even with 20% noise, our approach performs similarly to the best baseline trained on clean data. Taken together, this indicates that our approach is robust to noise in the relatively clean auxiliary signal, up to levels more than four times what is typically observed.

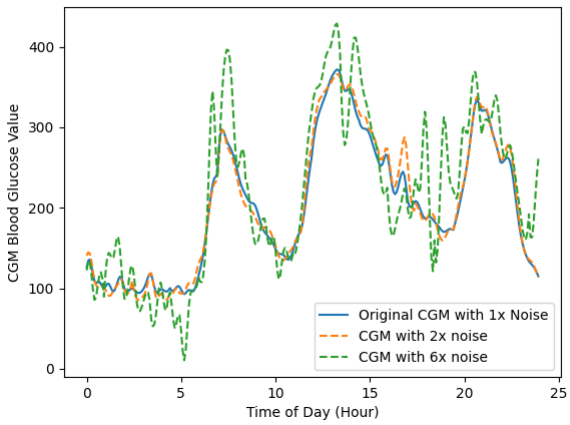


Figure 5: One day's worth of blood glucose data for simulated subject adult#002 with 1 x and 5 x additional simulator noise added. At 5 x, the signal is almost unrecognizable.

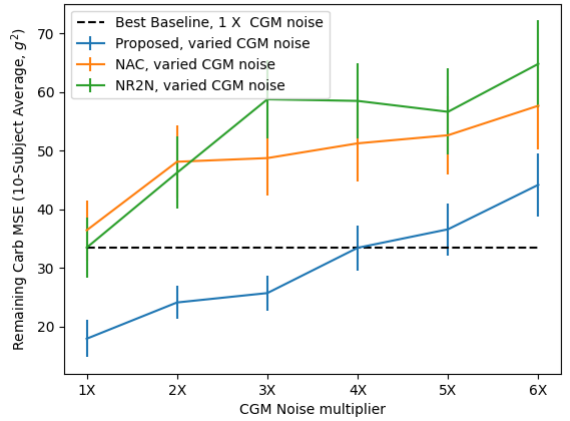


Figure 6: Our approach vs. strongest baselines for varying levels of noise in the CGM signal, average across all 10 simulated individuals with 1,000 sample bootstrap SEs as error bars. Our approach performs well even when noise is fairly large.

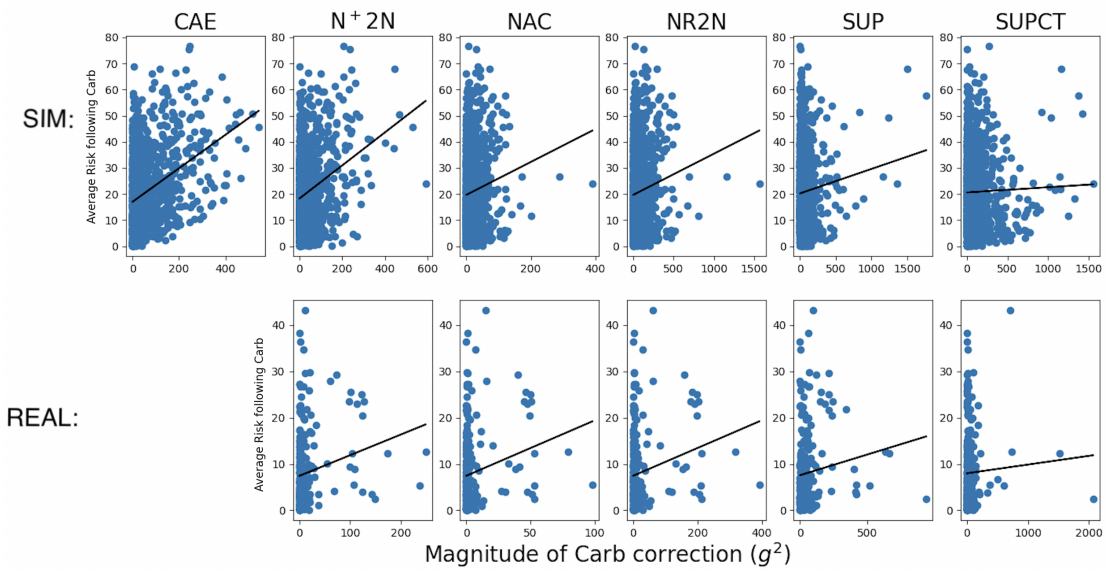


Figure 7: Risk following the carbohydrate vs. magnitude of carbohydrate correction learned for all models and both datasets. Besides the clean autoencoder, **N$^+$2N** performs best.

Appendix F. Sensitivity to hyperparameter τ

In order to examine the impact of including various amounts of data in our final training sample, we varied hyperparameter τ, which controls the proportion of samples within each minibatch that the networks are backpropagated on. We note that low values of τ, corresponding to a larger, noisier, sample, result in relatively stable performance, with all values between zero and 0.5 (half of the samples excluded) offering substantial improvement over the best performing baseline (**Figure 8**). Performance degrades for larger values of τ, which likely indicates that the approach is robust until the final training sample becomes too small to be effective.

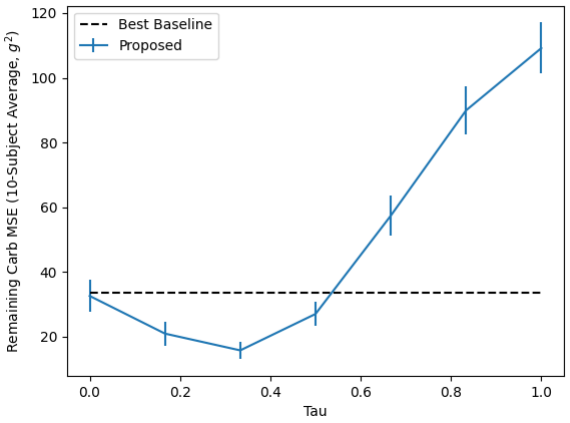


Figure 8: Model performance as a function of hyperparameter τ, with all else constant, across all 10 simulated individuals. Our model outperforms baseline as long as fewer than 50% of samples are excluded.

Appendix G. CRC Plots

With **N$^+$2N**, we see a higher correlation between the magnitude of carbohydrate correction and risk following the meal compared to baselines for both real and simulated data (**Figure 7**).